PHUSE EU Connect 2025 – Paper AS06

Multiple Imputations - overview of missing data problem, solution methods and SAS implementation

Dmytro Kalinin, CSL Vifor, Glattbrugg, Switzerland

ABSTRACT

In any study, there is often data that was intended to be collected but was not, due to various reasons such as patients not completing the study, missed visits, equipment failures, and other unforeseen circumstances. Missing data not only diminishes the power of statistical tests but can also lead to erroneous conclusions. There are numerous imputation methods available, such as Last Observation Carried Forward (LOCF), worst-case scenario, group averages, etc. However, the most reliable and universally applicable method is multiple imputations (MI). In this presentation, we will explain how to utilize multiple imputations in your analysis, the appropriate methods to select, and how to implement these techniques in SAS. Moreover, we will share our experiences in creating ADaM datasets for MI results. This article does not aim to present scientific research or novel discoveries. Rather, it serves as a consolidated overview of key information that may be valuable to statistical programmers working with SAS.

INTRODUCTION

Multiple imputation (MI) is a widely recognized methodology for analyzing datasets with missing values, specifically, where data intended to be collected is absent. In recent years, the issue of missing data in clinical trials has gained significant attention from both statisticians and regulatory authorities. This shift has prompted a reevaluation of the methods traditionally used to address missingness. Historically, relatively simple techniques - particularly single imputation methods-were commonly employed. For continuous variables, approaches such as Last Observation Carried Forward (LOCF) and Baseline Observation Carried Forward (BOCF) were routinely applied. However, recent research and regulatory guidance, including publications from the European Medicines Agency (EMA, 2010) and the FDA-commissioned panel from the National Research Council (NRC, 2010), have highlighted several limitations of these methods.

One major concern is that single imputation techniques fail to account for the uncertainty inherent in missing data. By treating imputed values as if they were observed, these methods can lead to underestimation of standard errors in statistical estimates. Additionally, contrary to earlier assumptions, single imputation may introduce bias - often favoring experimental treatment - depending on the nature and pattern of missingness in the study.

The basic idea of MI - let's build a model to predict missing data. In clinical research, statistical models are constructed using data collected from study participants to predict how a treatment is likely to perform in future patients. One practical approach involves building a model based on subjects with complete data and using it to estimate outcomes for those with missing values.

To illustrate the concept, let us consider a simplified example of statistical modeling. Let's say we measure the height of patients, and for several of them the height was not measured. We want to build a simple model with one factor "gender" to predict the height. The model will look like this:

$$H = \begin{cases} H_m \text{ if male} \\ H_f \text{ if female} \end{cases} + \varepsilon$$

where H_m is the average height of men (among those whose height was measured) and H_f is the average height of women (among those whose height was measured). That is, we will impute to all patients with missing height the average height of patients of the same sex whose height was measured.

While imputing missing values - such as patient height - using predictive models may appear to be a practical solution, it introduces a critical limitation. The imputed values are not observed measurements but statistical

estimates, each associated with a degree of uncertainty, typically expressed through a standard error and confidence interval. If these imputed values are subsequently used in downstream analyses, such as an ANOVA model where height is treated as a factor, the inherent uncertainty of the imputation is not accounted for. As a result, the analysis may underestimate the standard errors, leading to an overstatement of the precision and reliability of the results. This can ultimately compromise the validity of the statistical conclusions drawn from the study.

A solution to this problem was proposed by Donald Rubin, an emeritus professor at Harvard University in 1987. The prediction generated by the model under consideration is a random variable, and its residual error ε can be estimated. Furthermore, it is possible to compute the standard errors (SE) for the predicted mean heights of male H_m and female H_f patients, these estimates follow a normal distribution. These statistical properties are essential for understanding the uncertainty associated with the imputed values and should be considered when interpreting the results of subsequent analyses. To appropriately account for the uncertainty inherent in imputed values, a multiple imputation approach can be employed. This involves generating several plausible values for each missing observation using a random number generator, thereby creating multiple complete datasets. Each dataset is then analyzed independently using the intended statistical method, such as ANOVA, ANCOVA etc. The results from these separate analyses are subsequently combined to produce final estimates. This combination is performed using a set of established formulas known as the "Rubin Rules" which were developed by Rubin to correctly incorporate both within-imputation and between-imputation variability.

In the subsequent sections, we will examine various 'missing mechanisms', supported by illustrative examples. We will also demonstrate the implementation of the proposed multiple imputation (MI) methodology in SAS, utilizing the MI and MIANALYZE procedures. Finally, we will outline the process for creating ADaM datasets suitable for analysis using MI, in accordance with industry standards and regulatory expectations.

MISSING DATA MECHANISMS

Missing data mechanisms describe the underlying reasons why data may be absent in a database, categorized into Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR). Understanding these mechanisms is crucial for choosing appropriate data analysis and imputation methods to avoid biased results.

- Missing Completely at Random (MCAR) occurs when the missingness is entirely random and unrelated to any data, i.e. the probability that a particular patient will have a missing value does not depend on the patient.

A classic real-world example of Missing Completely at Random (MCAR) is when data is lost due to equipment failure or human error, such as a weighing scale running out of batteries or an entire batch of lab samples being improperly processed. In such scenarios, the missingness of the data is unrelated to any other observed or unobserved variables in the study, affecting all observations equally by chance rather than by a systematic pattern. This makes the remaining complete cases a random subset of the full dataset.

Example in a Study. Imagine a bone density study measuring activity levels with accelerometers. Accelerometers are attached to subjects to record their activity for 12 hours. The monitors frequently and randomly fail, leading to incomplete data for some subjects. This failure is considered MCAR because it happens randomly and is not due to any systematic difference between subjects. All subjects had an equal chance of experiencing a monitor failure. While the loss of data reduces the usable sample size and statistical power, it doesn't introduce bias into the results because the missing data are not related to any specific subject characteristics or study variables.

MCAR is the most desirable form of missingness, as it does not introduce bias and can often be handled with simple methods like listwise deletion (removing cases with missing data). While idealized, MCAR is often a strong and unrealistic assumption in real-world studies. It is more common for data to be Missing at Random or Missing Not at Random, where the missingness is related to other observed or unobserved variables, respectively. However, cases like equipment malfunction or accidental data loss remain the best examples of MCAR in practice.

- Missing at Random (MAR) happens when the probability of data being missing depends on other observed variables, i.e. the probability that a particular patient will have a missing value may depend on the observed factors

(treatment group, baseline characteristic etc). When MAR is assumed, the missing data mechanism can be considered "ignorable" for statistical analysis, meaning that appropriate statistical methods can be used to handle the missing data and obtain valid results.

A real-world example of MAR data is found in a health survey where depression severity scores are missing for a higher proportion of male participants than female participants. In this scenario, the likelihood of a score being missing is not dependent on the participant's actual depression severity (e.g., very severe cases are not more or less likely to have missing data) but is instead related to another observed variable—in this case, gender. Because the missingness is linked to an observable characteristic (gender), analytical methods that account for this relationship, such as multiple imputations, can be used to obtain unbiased results.

The missing data - depression severity scores are missing for some participants; observed variable - gender is a variable that is fully observed for all participants. The MAR relationship: the study observes that a higher percentage of males are missing these scores than females. The missingness is not because males have a specific level of depression severity (which would be MNAR), but rather because of an external factor linked to gender, such as males being less likely to complete surveys about their feelings. Since the missingness is related to the observed variable (gender), analysts can include gender in their statistical models to account for this pattern when imputing the missing depression scores.

- Missing Not at Random (MNAR) is the most complex case, where the missing data depends on the missing values themselves or unobserved variables, i.e. the probability that a particular patient will have a missing value may depend on unobservable factors, such as the missing value itself.

A common real-world example of MNAR data is in studies where participants with severe depression are more likely to drop out of a depression registry or study, making their self-reported severity the reason for their absence. The missing data (the survey responses from severe cases) are related to the unobserved values of the depression severity variable itself. You cannot assume the missing values are random; in fact, they are missing because the participants are experiencing the very phenomenon the study is measuring.

Similarly, participants in a study on weight loss might not report their weight because they are overweight, and this missingness is directly related to their actual, unobserved weight. The missing weight data are not at random; they are missing because of the actual weight value. People who are overweight have a higher probability of not having their weight recorded.

These scenarios are challenging because the reason for missingness is the variable itself or an unmeasured factor. MNAR is the most challenging type of missing data to address because the reason for missingness is not directly observed in the complete data. Researchers often need to make assumptions about the missing data mechanism or use sophisticated statistical techniques to try and account for the bias, such as sensitivity analyses.

An additional challenge in the analysis of missing data is the inherent difficulty in distinguishing between the Missing at Random (MAR) and Missing Not at Random (MNAR) mechanisms based solely on the observed data. The fundamental difference between these two mechanisms is precisely in the data that is not available. As such, the classification of missingness cannot be empirically validated using the existing dataset, which introduces uncertainty into the choice of appropriate imputation and analysis strategies.

This paper will primarily focus on the case MAR, which we aim to highlight and explore in detail.

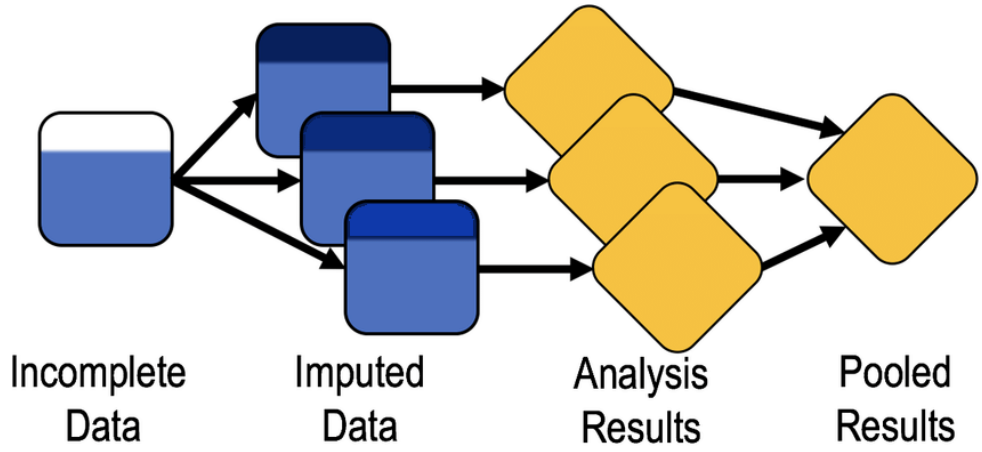
THREE STEPS OF MI

Analysis with multiple imputations is generally carried out in three steps:

1. Imputation: Multiple complete datasets are generated by replacing missing values with plausible values.

2. Analysis: Each of the multiple datasets is analyzed using a standard statistical method.

3. Pooling: The results (e.g., parameter estimates and their standard errors) from each analysis are combined into a single overall result using Rubin's rules, to account for the uncertainty introduced by the imputation process.



We will now proceed to examine each step in detail, highlight the key considerations, and illustrate them with examples implemented in SAS.

IMPUTATION STEP

The imputation step in MI involves creating multiple versions of a dataset by replacing missing values with plausible, random values drawn from a statistical model. This process generates several "complete" datasets, each with slightly different imputed values due to the random sampling, thereby capturing the uncertainty introduced by the missing data.

- Create Multiple Datasets: the original dataset with missing values is copied multiple times (we will discuss the required number of imputations later in this section).

- Replace Missing Values: in each copied dataset, the missing values are replaced with imputed values that are plausible and reflect the underlying data distribution. Random variation is introduced to ensure that each imputed dataset is slightly different, which accounts for the uncertainty in estimating the missing values.

- Goal: to create multiple “complete” versions of the original data that each contain different, but reasonable, values for the missing observations.

- The process is divided into two distinct phases: the P-step (Prediction) and the I-step (Imputation):

- In the P-step, a predictive model is constructed.
- In the I-step, random predictions are selected for imputation purposes.

Imputation step is implemented in SAS using **PROC MI**, it offers several methods for imputation of both continuous and categorical variables. Although this procedure involves a complex syntax, it is essential to understand the purpose and appropriate usage of the various PROC MI options. To ensure the correct implementation of multiple imputation (MI), the following key questions must be addressed:

- **Which variables are intended for imputation?** Identify the variables with missing data that require imputation. While study endpoints are typically the primary targets for imputation, it is advisable to impute the underlying variables when the endpoint is derived rather than directly measured. For instance, in psoriasis studies, the endpoint “PASI75” - defined as a 75% reduction in the PASI score, which quantifies disease severity—is calculated from the continuous PASI variable. In such cases, it is more appropriate to impute the PASI score itself as a continuous variable and subsequently derive PASI75, rather than imputing PASI75 directly as a binary outcome. Or, for example, when working with a quality-of-life questionnaire that includes multiple items and a composite total score, it is advisable to first impute any missing responses before recalculating the total score based on the completed dataset.

- **Which factors should be included in the imputation model?** General Recommendation - inclusion of a broader set of relevant factors generally enhances the quality and robustness of the imputation model. Key considerations for variable selection in imputation:
  - Variables targeted for analysis - include factors that will be part of the final analysis model to ensure consistency between imputation and analysis phases.
  - Variables that exhibit statistical correlation with the target of imputation should be incorporated into the imputation model, as they can enhance the accuracy of the imputed values. Examples include other relevant endpoints or repeated measurements of the same endpoint across different visits or timepoints.
  - Variables associated with missingness - consider factors that may influence the likelihood of missing data, such as completion status, reason for discontinuation, study site or location.
  - Multiple variables can be imputed simultaneously within the same procedure. Additionally, covariates included in the imputation model may themselves contain missing values and are subject to imputation. In essence, PROC MI treats all specified variables—both those targeted for imputation and those serving as predictors—as a unified set, and performs imputation for each variable based on the observed values of the others.
- **How many imputations should be performed?** Decide on the number of imputed datasets to generate, balancing statistical efficiency and computational feasibility. The sufficiency of the number of imputations can be evaluated using the Relative Efficiency (RE) parameter, which is calculated by PROC MI. This metric reflects the efficiency of the imputation relative to an infinite number of imputations. As a general guideline, an RE value greater than 0.95 is considered acceptable. Recommended Actions Based on RE:
  - If RE < 0.95, consider increasing the number of imputations.
  - Performing 50 imputations is typically sufficient in most scenarios.
  - For large datasets where computation time is a concern, reducing the number of imputations to 20–25 may be acceptable.
  - During program development or debugging, it is practical to use a smaller number of imputations (e.g., 5) and increase it once the code is finalized.
  - However, if RE remains below 0.95 even with 50 imputations, this may indicate a substantial proportion of missing data, which could compromise the validity of the overall analysis.

- **What imputation method and model type should be selected?** When choosing a method, we are guided by the type of variables and the missing pattern.

Missing data patterns describe how missing values are organized within a dataset.

- **Monotone Missing Data Patterns.** A dataset has a monotone missing pattern if the variables ("imputed" variable and all factors) can be arranged in an order such that if a variable is missing for an individual, all subsequent variables in that order are also missing.
  The pattern among missing values is evident and follows a predictable structure. It is often encountered in studies with permanent participant dropout or logically constrained data. Handling monotone data is generally simpler due to the identifiable structure.
  For instance, in a longitudinal study, if a participant drops out of the study after missing a measurement at time point 1, then all their measurements from time point 2 onwards will also be missing.

USUBJID	SEX	AGE	BASE	AVALV1	AVALV2
001	M	47	120	122	124
002	F	32	110	116	.
003	M	25	118	.	.
004	F	27	.	.	.

- **Arbitrary (Non-Monotone) Missing Data** has no such structured relationship, missing values can occur in isolated "islands" or clumps, not necessarily following any logical sequence. The position of missing

data is unpredictable and appears to be unrelated to other missing values. This pattern is more complex to handle, as the lack of a clear structure requires more sophisticated imputation methods.

USUBJID	SEX	AGE	BASE	AVALV1	AVALV2
001	M	47	120	122	124
002	F	32	110	.	116
003	M	25	.	.	120
004	F	.	121	125	120

Multiple imputation typically involves four principal methods, the selection of which depends on the type of variables and factors, as well as the underlying patterns of missing data.

- **MCMC (Monte-Carlo Markov Chain).** It is assumed that all variables - both those being imputed and the associated factors - follow a multivariate normal distribution. Consequently, all variables must, at a minimum, be continuous. If any variable is discrete, this method is deemed inappropriate. Missing pattern: arbitrary.
- **Monotone.** The variables subject to imputation, as well as the associated factors, may be continuous, binary, or categorical in nature. Missing pattern: monotone. It operates in a non-iterative manner, making it the most time-efficient approach. It is particularly suitable when the monotone pattern of missingness can be confidently established - for instance, when all factors represent baseline characteristics.
- **FCS (Fully Conditioned Specifications).** The variables subject to imputation, as well as the associated factors, may be continuous, binary, or categorical. This method is considered the most flexible approach. Missing pattern: arbitrary. It is recommended when a monotone pattern of missingness cannot be assured.
- **Hybrid MCMC/Monotone method.** This method is considered outdated and was commonly used prior to the development of FCS. Nonetheless, it remains in use by some practitioners, making it beneficial to be familiar with its structure. The hybrid approach consists of two sequential steps: initially, MCMC techniques are applied to impute intermediate values and establish a monotone missing data pattern; subsequently, the monotone method is employed to complete the imputation process.

Method	Missing pattern	Variable types	Recommendation
MCMC	Arbitrary	Only continuous	No
Monotone	Monotone	Any type	If monotone pattern
FCS	Arbitrary	Any type	If arbitrary pattern
MCMC/Monotone	Arbitrary	Any type	Use on demand

Before we proceed with the MI procedure in SAS, we would like to briefly discuss the model types supported by the most used methods. Specifically, the Monotone and Fully Conditional Specification methods accommodate a variety of model types, including the following:

- **Reg**: Linear regression.
- **Regpmm** (regression predictive mean matching): After regression, the imputed value is selected from among several observed values close to the regression predicted value.
- **Logistic**: Logistic regression.
- **Discrim**: Discriminant function.
- Formally, the Monotone method also has a **Propensity** type, but it is not recommended.

As a brief refresher, let us revisit the classification of data types:

- **Continuous** - a variable that continuously ranges over a certain domain (e.g., height, weight, blood sugar level, blood pressure, etc.) Sometimes when a variable is discrete, e.g. integer, but its range is much larger than the discreteness step, we still consider it continuous (e.g., age in years or months,

facial lesion count, etc.), but if the range is small e.g. 1, 2 or 3, we won't think of this variable as continuous.

- **Ordinal** – a variable that takes a few discrete values that are logically ordered. For instance: AE severity: mild, moderate, severe; drug doses: low dose, middle dose, high dose; global assessment of change: much worse, a little worse, the same, a little better, much better; frequency of a symptom: never, a little of the time, some of the time, most of the time, all the time.
- **Nominal** - a variable that takes a few discrete values that are not ordered in any meaningful way. For example: Sex, Race, Location of facial lesions (forehead, cheek).
- **Binary** - a special case of both nominal and ordinal which takes only just two values: yes/no, success/failure, response/non-response, etc.

The selection of the appropriate model type is guided by the nature of the variable being imputed:

- **Continuous**: Reg or Regpmm. Recommendation: variable with a large range and small discretization step (e.g., laboratory tests, vital signs) - Reg; variable with a limited range or large discretization step (e.g., questionnaire score, always an integer and in the range 0-10) - Regpmm.
- **Binary** or **Ordinal**: Logistic.
- **Nominal**: Discrim. This model type works better with continuous factors but cannot use interaction terms.

PROC MI IN SAS

Having provided the theoretical foundation for the imputation step, we now turn our attention to the implementation of the PROC MI procedure in SAS. This section will explore its syntax across various use cases and present practical examples, along with recommendations and insights drawn from practical experience. So, the basic syntax of the procedure looks like this:

```
proc mi ...;
   var var1 var2 var3 var4 ...;
   class var1 var2 ...;
   mcmc ...;
   monotone ...;
   fcs ...;
run;
```

Typically, only one operator method - either MCMC, Monotone, or FCS - is specified within a given imputation procedure. While it is permissible to invoke the same method multiple times for different variables, combining multiple distinct methods within a single procedure is not supported.

- **Main options in proc MI:**
  - DATA= specifies the input dataset.
  - OUT= designates the output dataset.
  - NIMPUTE= defines the number of imputations to be performed.
  - If the input dataset (DATA) contains N observations, the output dataset (OUT) will contain N × NIMPUTE observations. Additionally, a new variable named _IMPUTATION_ will be added to identify each imputation iteration. **A simple way to find out the missing data pattern - run proc MI with NIMPUTE=0.**
  - SEED= Sets an arbitrary numeric value to initialize the random number generator. This is essential for quality control (QC), ensuring reproducibility of results across repeated runs.
  - MIDISPLAYPATTERN= Displays missing data patterns table.
  - MINIMUM=, MAXIMUM=, ROUND=. Specifies the minimum or maximum value for the imputed variable and how it should be rounded. After the equal sign, use as many numbers as there are variables in the var statement. The period indicates that the variable should not be converted. Options

are meaningful for MCMC, Monotone/FCS Reg methods. In other cases, the imputed value is always one of the observed values.

For example: SEX, AGE, and BASE variables remain unchanged. AVALV1 and AVALV2 are rounded to the nearest integer and are limited to a range of 1 to 100.

```
proc mi minimum=. . . 1 1 maximum=. . . 100 100 round=. . . 1 1;
  var sex age base avalv1 avalv2;
```

- **Statements in proc MI:**
  - VAR - specifies all relevant variables, including those intended for imputation as well as all factors used in the modeling process.
  - CLASS - The CLASS statement lists those variables that are classificatory (categorical) - as in all SAS procedures.
  - MCMC <options> - specifies the details of the MCMC method for imputation. Some of the options: CHAIN= single or multiple, we can leave at by default; IMPUTE=Monotone - only until a monotone pattern is achieved, used in the hybrid MCMC/Monotone method.
  - MONOTONE <model type> (imputed variable = factors). For example:
    a) MONOTONE Reg (avalv2 = sex age base avalv1).
    b) MONOTONE Logistic (avalv1 = sex age trtp base trtp*base).

    All variables are imputed in turn. Factors will be all variables from the VAR specified before this one, so it is usually not necessary to specify factors after the "=" sign. It might be necessary if we want to use interaction factor (trtp*base in our example b).

  - FCS <model type> (imputed variable = factors). FCS has the same syntax as MONOTONE. For example:

    a) FCS Regpmm (avalv2 = sex age base avalv1).
    b) FCS Discrim (avalv2 = sex age base avalv1 / CLASSEFFECTS=include). The "CLASSEFFECTS" option is needed if there are classification factors among the factors.

To conclude this section, we will present several examples illustrating the application of the PROC MI procedure in SAS.

1. Method Monotone with Regpmm model:

```
proc mi data=indata out=outdata nimpute=50 seed=45780;
  var weight base chg4 chg8 chg12;
  monotone regpmm(chg4 chg8 chg12);
run;
```

2. Method FCS with Reg model for two variables:

```
proc mi data=indata out=outdata nimpute=50 seed=122001;
  var trtp skinclass base chg12 chg24;
  class trtp skinclass;
  fcs reg(chg12=trtp skinclass base trtp*base chg24);
  fcs reg(chg24=trtp skinclass base trtp*base chg12);
run;
```

3. Hybrid MCMC/Monotone method with IMPUTE=Monotone option.

   First proc MI used to prepare the monotone pattern. Then we can use the output dataset "mono" in proc MI to finalize hybrid approach.

```
proc mi data=indata out=mono nimpute=50 round=1 seed=45779;
  by trtpn;
  var weight base chg4 chg8 chg12;
  mcmc impute=monotone;
run;

proc mi data=mono out=outdata nimpute=1 <options>;
  by _imputation_;
  var weight base chg4 chg8 chg12;
  monotone reg (chg4 chg8 chg12);
run;
```

Where **trtp/trtpn** – treatment group; **weight, skinclass** – some baseline characteristics; **base** – baseline value for our endpoint, **chgN** – change from baseline value at visit N for our endpoint; **trtp*base** - an interaction factor is a term used in a model to account for how the effect of one variable changes across the levels of another variable.

ANALYSIS STEP

The analysis step in multiple imputation involves conducting the same complete-data statistical analysis on each of the separately imputed datasets. Each of the imputed datasets is analyzed separately using any method that would have been chosen had the data been complete. This step can be implemented using any analytical procedure in SAS, e.g., PROC MIXED, PROC GLM, PROC LOGITIC, PROC NPAR1WAY, PROC FREQ, etc. We apply procedures with the BY _IMPUTATION_ operator and save the results in datasets using the Output Delivery System (ODS) in SAS.

A critical component of this analysis step - addressed in detail through examples in the subsequent "Pooling" chapter - is the specification of a properly structured output dataset. This dataset must contain the estimated statistics and their corresponding standard errors from each repetition of the analysis. It serves as the essential input for the PROC MIANALYZE procedure, which is utilized during the "Pooling" step.

POOLING STEP

The pooling step in multiple imputation combines parameter estimates and their standard errors from analyses of multiple imputed datasets into a single, combined result using Rubin's rules. This final step produces a single set of results with appropriate statistical properties by incorporating the variability within each imputation and the additional variation between the imputations. The goal is to arrive at valid conclusions from the data by incorporating the uncertainty of the missing data into the final statistical inference.

Pooling step is implemented in SAS using **PROC MIANALYZE**. The syntax used for PROC MIANALYZE is contingent upon the analytical approach employed during the preceding analysis step. SAS offers direct support for certain procedures, while others require a more generalized implementation strategy. The most used examples and general approach will be discussed in this section.

1. Processing **PROC MIXED** results.

   PROC MIXED implements a mixed model, but also ANOVA and ANCOVA. We typically process results from ODS datasets LSMeans and Diffs. **PROC GLM** and **PROC GENMOD** results are processed in the same way.

```
proc mianalyze parms=LSMeans;
  class trtpn;
  modeleffects trtpn;
run;
```

Within the PARMS statement, specify the dataset containing the LSMeans or Diffs results.

In the MODELEFFECTS statement, include the same effects as specified in the PROC MIXED procedure for LSMeans - this may involve an interaction term such as trtpn*avisitn.

The CLASS statement should list all classification variables used in the model.

When processing Diffs, the BY trtpn _trtpn statement may be required if multiple comparisons are present.

The results in the ODS dataset ParameterEstimates:

Parameter	trtpn	Estimate	Std Error	95% Confidence Limits		Pr > |t|
trtpn	1	0.771134	0.148230	0.480577	1.061690	<.0001
trtpn	2	1.914720	0.150902	1.618771	2.210669	<.0001

Parameter	trtpn	Estimate	Std Error	95% Confidence Limits		Pr > |t|
trtpn	1	-1.143586	0.211537	-1.55833	-0.72884	<.0001

How PROC MIANALYZE understands the PARMS dataset?

Imputation	Effect	trtpn	Estimate	StdErr	DF	tValue	Probt
1	trtpn	1	0.7718	0.1435	89	5.38	<.0001
1	trtpn	2	1.9442	0.1430	89	13.60	<.0001
2	trtpn	1	0.7177	0.1488	89	4.82	<.0001
2	trtpn	2	1.9358	0.1483	89	13.05	<.0001
3	trtpn	1	0.7629	0.1484	89	5.14	<.0001
3	trtpn	2	1.8431	0.1479	89	12.46	<.0001
4	trtpn	1	0.7695	0.1459	89	5.27	<.0001
4	trtpn	2	1.9354	0.1454	89	13.31	<.0001

- _ IMPUTATION_ - imputation number.
- EFFECT - name from MODELEFFECTS.
- TRTPN – value for CLASS variables.
- Estimate – estimate.
- StdErr - standard error.
- Uniqueness - _ IMPUTATION_, EFFECT, TRTPN.

2. Processing **PROC LOGISTIC** results.

```
proc logistic data=indata;
  by _imputation_;
  class trtpn site / param=ref;
  model avalc = trtpn site base;
run;
```

Saving the ODS ParameterEstimates dataset.

```
proc mianalyze parms(classvar=classval)=ParameterEstimates;
  class trtpn site;
  modeleffects trtpn site;
run;
```

Differ from PROC MIXED results processing by the CLASSVAR=CLASSVAL option. As a result, we obtain estimates of the model parameters, but we need to obtain the odds ratios.

Parameter	trtpn	site	Estimate	Std Error	95% Confidence Limits		Pr > |t|
trtpn	1		-0.341673	0.420978	-1.16706	0.483716	0.4171
site		1	0.127304	0.622060	-1.09233	1.346943	0.8379
site		2	-0.693500	0.718109	-2.10162	0.714616	0.3343
site		3	-0.214927	0.646589	-1.48343	1.053577	0.7396
site		4	-0.126253	0.717925	-1.53424	1.281733	0.8604

We need to exponentiate (using the **exp** function) the parameter estimate and confidence limits.

Don't forget to specify the PARAM=REF option in the CLASS operator of PROC LOGISTIC, otherwise it won't work! It specifies the parameterization method for the classification variable.

A similar approach works for the Hazard Ratio in **PROC PHREG**.

3. Processing **general case** results.

If a procedure is not explicitly supported by PROC MIANALYZE, the results must be processed using a general approach. The input dataset should include a row for each imputation, containing paired variables that represent the estimate and its associated standard error.

```
proc mianalyze data=indata;
  modeleffects est1 est2 …;
  stderr StdErr1 StdErr2 …;
run;
```

The variables are assumed to follow a normal distribution. The PARMS option is applicable when using output datasets from procedures that are directly supported by PROC MIANALYZE, whereas the DATA option should be used in the general case where such direct support is not available

To illustrate the general approach for processing results, consider the following examples involving the Wilcoxon test and binomial proportion.

4. Processing **PROC NPAR1WAY** results for Wilcoxon test.

```
proc npar1way data=indata wilcoxon;
  by _imputation_;
  class trtpn;
  var aval;
run;
```

The WilcoxonTest dataset, generated via ODS, contains the variable Z, which represents a test statistic assumed to follow a standard normal distribution.

To prepare this dataset for analysis using PROC MIANALYZE, a DATA step is performed to append a new variable, StdErr, with a constant value of 1 across all records. This modified dataset is then used as input for PROC MIANALYZE to compute the corresponding p-value.

```
proc mianalyze data=WilcoxonTest;
  modeleffects Z;
  stderr StdErr;
run;
```

5. Processing **PROC FREQ** results for binomial proportion.

Consider a scenario where the objective is to analyze the proportion of patients who achieved a specific response. Although proportions (or percentages) inherently follow a binomial distribution, when the sample size is sufficiently large, the distribution of the estimates approximates normality. Under such conditions, it is appropriate to apply PROC MIANALYZE for further analysis.

```
proc freq data=indata;
  by trtpn _imputation_;
  tables avalc / binomial;
run;
```

The Binomial dataset generated via ODS contains the proportion of interest along with its Asymptotic Standard Error (ASE). To prepare this dataset for analysis using PROC MIANALYZE, it must be transformed - either through a PROC TRANSPOSE step or a DATA step - so that each row corresponds to a unique combination of TRTPN and _IMPUTATION_, with the proportion (_BIN_ variable) and ASE (E_BIN variable) presented as paired variables on a single line.

```
proc mianalyze data=Binomial_transformed;
  by trtpn;
  modeleffects _BIN_;
  stderr E_BIN;
run;
```

To process the difference in proportions, we calculate the SE of the difference = sqrt ($SE1^2 + SE2^2$).

6. Processing **other cases**.

The following types of analysis are difficult to process because the statistics are not normally distributed: Chi-square test, CMH test, Fisher's test, Kaplan-Meier estimates, Log-rank test, Correlation coefficients and many others.

General approach: begin by applying a transformation that normalizes the distribution of the target statistic. Specific formulas for these transformations vary depending on the type of analysis and are documented in specialized statistical literature. Once the data has been transformed, proceed with the analysis using PROC MIANALYZE. Finally, apply the inverse transformation to return the results to their original scale for interpretation.

MULTIPLE IMPUTATION AND ADAM DATASETS

A key consideration when implementing MI is whether to store the results generated by the PROC MI procedure in an ADaM dataset or to perform all three MI steps - imputation, analysis, and pooling - within the table generation program.

The general recommendation is to save the results to am ADaM dataset if the MI-based analysis is the primary focus or if multiple tables are generated using MI. Otherwise, it is acceptable to conduct all MI steps directly within the tables program.

Generating ADaM datasets for MI requires incorporating the imputed values into a CDISC-compliant structure. Key considerations include using variables to identify imputed values (like **DTYPE**), maintaining traceability, and transforming data between horizontal (for imputation) and vertical (for analysis) structures as needed.

It is advisable to create two separate datasets (e.g. ADEF and ADEFMI): one without multiple imputation (MI) and another with MI applied. Given that the MI dataset is typically large, maintaining a non-imputed version allows for quicker access to the original data when needed - for example, for generating listings. Extracting the initial data from the MI dataset can be time-consuming.

The recommended workflow for generating a dataset incorporating multiple imputation (MI):

- **Restructure the Dataset Horizontally**. If the imputation model utilizes data from multiple visits or endpoints, the source dataset must first be transposed into a horizontal format to accommodate the model requirements.
- **Execute the Imputation Procedure**. Apply the PROC MI procedure to perform the imputation. It is essential to specify a fixed SEED value to ensure reproducibility and facilitate the quality control (QC) process.
- **Revert to Vertical Structure**. Once the imputation is complete, transpose the dataset back to its original vertical structure for consistency with standard ADaM formats.
- **Identify Imputed Values**. As PROC MI does not inherently distinguish between observed and imputed values, merge the imputed dataset with the original source data using key variables. This allows for the identification of imputed records.
- **Assigning Imputation Data Type**. For each imputed value, assign a DTYPE such as "MI" to clearly indicate its origin.
- **Rename the Imputation Index Variable**. Replace the default _IMPUTATION_ variable name with a valid and meaningful label, such as IMPNO, to align with dataset naming conventions.

ACKNOWLEDGMENTS

I would like to thank CSL Vifor and my manager Vasuki Kumpeson for encouraging and supporting this work as well as for conference participation.

I would like to express my sincere gratitude to Isabelle Morin from CSL Vifor for her invaluable support in providing real project examples and highly relevant reference materials.

I would also like to express my profound gratitude to Leonid Zeitlin, my first supervisor at the Ukrainian company CS Ltd, under whose guidance I worked for nine years. Leonid played a pivotal role in shaping my professional foundation, and his mentorship was instrumental in my development. The inspiration to write this article stemmed from his seminar on multiple imputations, held in Kharkiv in 2022.

REFERENCES

"Multiple Imputations Seminar". Leonid Zeitlin, Kharkiv, 2022.

"Multiple Imputation of Missing Data Using SAS". Patricia Berglund, Steven G. Heeringa.

"Combining Analysis Results from Multiply Imputed Categorical Data". Bohdana Ratitch, Ilya Lipkovich, Michael O'Kelly: https://www.lexjansen.com/pharmasug/2013/SP/PharmaSUG-2013-SP03.pdf

Combining Survival Analysis Results after Multiple Imputation of Censored Event Times. Jonathan L. Moscovici, Bohdana Ratitch: https://www.pharmasug.org/proceedings/2017/SP/PharmaSUG-2017-SP05.pdf

SAS documentation: PROC MI and PROC MIANALYZE.

CONTACT INFORMATION

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

Dmytro Kalinin
CSL Vifor
Flughofstrasse 61, Postfach 8152 Glattbrugg, ZH, Switzerland
E-mail: dmytro.kalinin@viforpharma.com