Paper AS07

Contributions to CAMIS PHUSE project in 2025: Tobit regression, reference-based multiple imputation and tipping point analysis

Yannick Vandendijck, Johnson & Johnson, Beerse, Belgium
Cristina Sotto, Johnson & Johnson, Beerse, Belgium
Sarah Brosens, KU Leuven, Leuven, Belgium
Christina E Fillmore, GSK, London, UK
Lyn Taylor, Parexel, Sheffield, UK

INTRODUCTION

Statisticians and statistical programmers using multiple statistical software (SAS/STAT®, R, Python) will have found differences in analysis results that warrant further exploration and justification. Possible discrepancies across statistical software implementations of equivalent analysis can cause unease when submitting results to regulatory agencies. This has become increasingly important since the pharmaceutical industry is increasingly turning to open-source software like R, drawn to its flexibility, innovation, added value and cost-effectiveness. The PHUSE DVOST CAMIS (**C**omparing **A**nalysis **M**ethod **I**mplementations in **S**oftware) project aims to investigate and document differences and similarities between different statistical software by providing comparisons and comprehensive explanations. CAMIS contributes to the transition and confidence in reliability of open-source software.

Here, we will cover some important 2025 contributions to CAMIS such as tobit regression (SAS Proc LIFEREG versus 3 R-packages), reference-based multiple imputation & tipping point analysis (SAS Five Macros vs rbmi R-package). Key results and examples on differences and similarities between SAS and R will be discussed.

CAMIS

Traditionally, highly regulated industries (such as the pharmaceutical industry), have limited themselves to the use of commercially available software. When taking such an approach, the responsibility for the validation and testing of the product was often delegated to the software development company themselves, to ensure the software performs in line with its documentation, producing accurate, reliable and reproducible results. However, one downside of this approach is that new methods and functionality can be slow to be adopted, limiting new methodology implementation and tools that can bring in efficiency.[1]

The rate at which community led tools and methods are being developed in open-source software is rapid. The availability of advanced analytic capabilities has led to increased desire for statistical staff in regulated industries to have access and approval to use open-source software[2] (Rimler et al., 2022). The use of open-source software is possible in pharmaceutical industry. For example, in the Statistical Software Clarifying Statement[3], the US Food and Drug Administration (FDA, 2015) states that “FDA does not require use of any specific software for statistical analyses” and that “the computer software used for data management and statistical analysis should be reliable”.

However, several discrepancies have been discovered in statistical analysis results between different programming languages, even in fully qualified statistical computing environments. Subtle differences exist between the fundamental approaches implemented by each language, yielding differences in results which are each correct in their own right. The fact that these differences exist causes unease on the behalf of sponsor companies when submitting to a regulatory agency, as it is uncertain if the agency will view these differences as problematic. Observing differences across languages can reduce the analyst’s confidence in reliability and, by understanding the source of any discrepancies, one can reinstate confidence in reliability.

The goal of CAMIS is to demystify conflicting results between software and to help ease the transitions to new languages by providing comparison and comprehensive explanations. Currently, the most interest in CAMIS is in comparing SAS and R for a wide range of statistical analysis methods.

Please visit the CAMIS website for more information. Through the creation of the CAMIS White Paper, the group provided guidance on the types of questions statistical staff should ask to aid with replication of analysis methods in different software and to identify the fundamental sources of discrepant results between software.

CAMIS is a cross-industry PHUSE DVOST Working Group, run in collaboration with members from PHUSE, PSI, ASA and IASCT.

CAMIS REPOSITORY

The CAMIS repository is available on the CAMIS website, and encompasses many different statistical methods such as Summary Statistics and Correlation; General Linear Models; Generalized Linear Models; Non-parametric Analysis;

Categorical Data Analysis; Repeated Measures; Multiple Imputation; Survival Models; Sample Size/Power Calculations. The CAMIS repository is operating through an open-source repository on GitHub (https://github.com/PSIAIMS/CAMIS). To date, the open-source repository has over 60 contributors.

In the next sections, we summarize three additions to the CAMIS repository that were added in 2025: [1] Tobit regression, [2] Reference-based multiple imputation, and [3] Tipping point analysis. For more details, please visit the CAMIS website.

TOBIT REGRESSION

Tobit regression[5,6], also called a censored regression model, can be used to estimate linear relationships between independent variables and a dependent variable (endpoint) that is either left- or right-censored at a specific known value. The standard Tobit model assumes a normally distributed endpoint. On CAMIS, the standard Tobit model in the case where the endpoint has a lower limit was explored. For example, in virology, sample results could be below the lower limit of detection (e.g., 100 copies/mL) and in such a case we only know that the sample result is <100 copies/mL, but we don't know the exact value.

Let y^* be the true underlying latent variable, and y the observed variable. We discuss here censoring on the left:

$$y = \begin{cases} y^*, y^* > \tau \\ \tau, \ y^* \leq \tau \end{cases}$$

We consider tobit regression with a censored normal distribution. The model equation is

$$y_i^* = X_i\beta + \epsilon_i, \text{ with } \epsilon_i \sim N(0, \sigma^2).$$

But we only observe $y = max(y^*, \tau)$. It is important to note that β estimates the effect of x on the latent variable y^*, and not on the observed value y. The implementations of Tobit regression in both R and SAS are presented on CAMIS. A comparison of the implementations in both languages is shown.

R IMPLEMENTATION

The censReg, survival and VGAM packages were explored. The data used is censored on the left at a value of τ = 8.0.

ID	ARM	Y	CENS
001	A	8.0	1
002	A	8.0	1
003	A	8.0	1
004	A	8.0	1
005	A	8.9	0
006	A	9.5	0
007	A	9.9	0

Figure 1: First seven records of dataset used for tobit regression.

The censReg() function from the censReg package performs tobit regression for left and right censored. The model is estimated by maximum likelihood assuming a Gaussian (normal) distribution of the error term. The R code is:

```
censReg::censReg(Y ~ ARM, left = 8.0, data = dat_used)
```

Using the survreg() function from the survival package a tobit regression could be performed as well. The model is estimated by maximum likelihood assuming a Gaussian (normal) distribution of the error term. The R code is:

```
survival::survreg(Surv(Y, 1-CENS, type="left")~ ARM, data=dat_used, dist = "gaussian")
```

The VGAM package provides functions for fitting vector generalized linear and additive models (VGLMs and VGAMs). This package centers on the iteratively reweighted least squares (IRLS) algorithm. The vglm() function offers the possibility to fit tobit regression models. The R code is:

```
VGAM::vglm(Y ~ ARM, tobit(Lower = 8.0), data=dat_used)
```

SAS IMPLEMENTATION

The LIFEREG procedure is used for tobit regression. For tobit regression, the following model syntax needs to be used: `MODEL (lower, upper) = effects / options;` here, if the lower value is missing, then the upper value is used as a left-censored value. Therefore, a new variable should be created in the dataset that is set to missing for censored observations. The SAS code for tobit regression is:

```
data dat_used;
  set dat_used;
  if Y <= 8.0 then lower=.; else lower=Y;
run;
proc lifereg data=dat_used order=data;
  class ARM;
  model (lower, Y) = ARM / d=normal;
  lsmeans ARM /cl alpha=0.05;
  estimate 'Contrast B-A' ARM 1 -1 / alpha=0.05;
run;
```

R VS SAS COMPARISON

The following table shows the tobit regression analysis results for R and SAS for the considered dataset.

Statistic	censReg:: censReg()	survival:: survreg()	VGAM:: vglm()	SAS LIFEREG
Treatment effect	1.8225	1.8225	1.8226	1.8225
Standard error	0.8061	0.8061	0.7942	0.8061
p-value	0.0238	0.0238	0.0217	0.0238
95% CI (Wald based)	0.2427 to 3.4024	0.2427 to 3.4024	0.2661 to 3.3791	0.2427 to 3.4024
σ	1.7316	1.7316	1.7317	1.7316

The censReg() and survreg() functions in R and LIFEREG procedure in SAS provide completely matching results in terms of treatment effect estimates, standard error of the estimate, 95% confidence intervals, p-values and estimate of σ. The vglm() function provides slighty different numerical results, which is caused by a different estimation technique (the VGAM package uses vector generalized linear and additive models which are estimated using an iteratively reweighted least squares (IRLS) algorithm).

Few additional notes on the analysis:

- Typically, the Tobit model assumes normal distributed data. Within SAS LIFEREG procedure and R survival package multiple other different distributional assumptions are possible.
- In R, survreg() function allows for easy incorporation with the emmeans package.
- Using SAS LIFEREG procedure, one-sided p-values can be easily obtained by adding `UPPER` or `LOWER` in the estimate statement. In none of the considered R functions this is readily available, and the one-sided p-value needs to be calculated manually.

REFERENCE-BASED MULTIPLE IMPUTATION (RBMI)

Reference-based multiple imputation (MI) methods have become popular for handling missing data, as well as for conducting sensitivity analyses, in randomized clinical trials. In the context of a repeatedly measured continuous endpoint assuming a multivariate normal model, Carpenter et al. (2013)[7] proposed a framework to extend the usual MAR-based MI approach by postulating assumptions about the joint distribution of pre- and post-deviation data. Under this framework, one makes qualitative assumptions about how individuals' missing outcomes relate to those observed in relevant groups in the trial, based on plausible clinical scenarios. Statistical analysis then proceeds using the method of multiple imputation[8,9].

In general, multiple imputation of a repeatedly measured continuous outcome can be done via 2 computational routes[10]:

1. Stepwise: split problem into separate imputations of data at each visit
    - requires monotone missingness, such as missingness due to withdrawal
    - conditional on the imputed values at previous visit
    - Bayesian linear regression problem is much simpler with monotone missing, as one can sample directly using conjugate priors
2. One-step approach (joint modelling): Fit a Bayesian full multivariate normal repeated measures model using MCMC and then draw a sample.

Here, we illustrate reference-based multiple imputation of a continuous outcome measured repeatedly via the so-called one-step approach.

R IMPLEMENTATION

The rbmi package[11] will be used for the one-step approach of the reference-based multiple imputation using R. The package implements standard and reference-based multiple imputation methods for continuous longitudinal endpoints. The following standard and reference-based multiple imputation approaches will be illustrated

- MAR (Missing At Random)
- CIR (Copy Increment from Reference)
- J2R (Jump to Reference)
- CR (Copy Reference)

To illustrate reference-based multiple imputation a publicly available example dataset from an antidepressant clinical trial of an active drug versus placebo is used. Overall, data of 172 patients is available with 88 patients receiving placebo and 84 receiving active drug. The relevant endpoint is the Hamilton 17-item depression rating scale (HAMD17) which was assessed at baseline and at weeks 1, 2, 4, and 6 (visits 4 to 7 in the dataset). Study drug discontinuation occurred in 24% (20/84) of subjects on active drug and 26% (23/88) of subjects from placebo. All data after study drug discontinuation are missing.

PATIENT	GENDER	THERAPY	RELDAYS	VISIT	BASVAL	HAMDTL17	CHANGE
1503	F	DRUG	7	4	32	21	-11
1503	F	DRUG	14	5	32	20	-12
1503	F	DRUG	28	6	32	19	-13
1503	F	DRUG	42	7	32	17	-15
1507	F	PLACEBO	7	4	14	11	-3
1507	F	PLACEBO	15	5	14	14	0
1507	F	PLACEBO	29	6	14	9	-5
1507	F	PLACEBO	42	7	14	5	-9
1509	F	DRUG	7	4	21	20	-1
1509	F	DRUG	14	5	21	18	-3

Figure 2: Example records of the dataset used for reference-based multiple imputation.

The missingness pattern is shown below. The incomplete data is primarily monotone in nature. 128 patients have complete data for all visits. 20, 10 and 13 patients have 1, 2 or 3 monotone missing data patterns, respectively. Further, there is one subject with a single intermittent missing observation at visit 5.

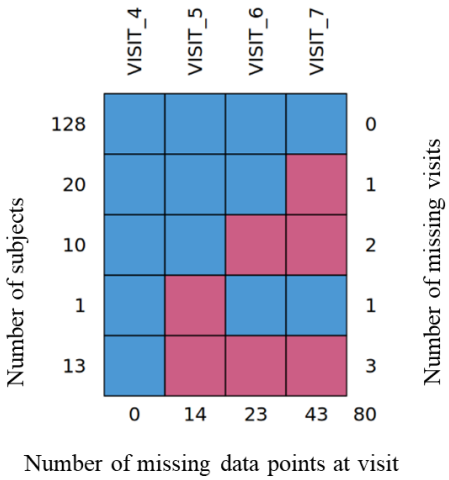

Figure 3: Missing data pattern of the dataset used for reference-based multiple imputation.

We provide a summary of the implementation of reference-based multiple imputation using the rbmi package in R. For the full code, visit the CAMIS website and/or consult the rbmi quickstart vignette.

1. rbmi expects the input dataset to be complete; that is, there must be one row per subject for each visit (note: in clinical trials ADAMs typically do not have this required complete data structure). Missing outcome values should be coded as NA, while missing covariate values are not allowed. If the dataset is incomplete, then

the expand_locf() function can be used to add the missing rows, using LOCF imputation to carry forward the observed baseline covariate values to visits with missing outcomes.

2. Next, a dataset must be created specifying which data points should be imputed with the specified imputation strategy (i.e. MAR, CIR, J2R, CR). In our example, the dataset *dat_ice* is created which specifies the first visit affected by an intercurrent event (ICE) and the imputation strategy for handling the missing outcome data after the ICE (i.e. MAR, CIR, J2R, CR). In our example, the subject's first visit affected by the ICE "study drug discontinuation" corresponds to the first terminal missing observation.

3. Next, the user needs to define the imputation model and imputation method using the set_vars() and method_bayes() function.

4. Using the draws() function the imputation model is fitted, and Bayesian posterior parameter draws are performed for this model.

5. Using the posterior parameter draws from the draws object, imputed datasets are generated via the impute() function. This function only has two key inputs: the imputation model output from draws() and the references group to be used for reference-based imputation methods. For example, for MAR this will be as follows since no reference group is used in MAR:
   ```
   impute(draws = drawObj, references = NULL)
   ```
   For MNAR approaches CIR, J2R and CR this will be as follows:
   ```
   impute(draws = drawObj, references = c("PLACEBO"="PLACEBO", "DRUG"="PLACEBO"))
   ```

6. The next step is to run the analysis model on each imputed dataset. This is done by calling the analyse() function. In our example, the ancova() function in the rbmi package is used, and fits a separate ANCOVA model for the outcomes at each visit.

7. Finally, the pool() function is used to summarise the analysis results across multiple imputed datasets to provide an overall statistic with a standard error, confidence intervals and a p-value for the hypothesis test of the null hypothesis that the treatment effect is equal to 0. Since we used method_bayes(), pooling and inference are based on Rubin's rules[8,9].

SAS IMPLEMENTATION

For reference-based multiple imputation in SAS, the five macros[10] are used (available at LSHTM DIA Missing Data under Reference-based MI via Multivariate Normal RM (the "five macros and MIWithD")). These macros fit a Bayesian normal repeated measures model and then impute post withdrawal data under a series of possible post-withdrawal profiles including J2R, CIR and CR. It then analyses the data using a univariate ANOVA at each visit and summarizes across imputations using Rubin's rules[8,9]. For all details, consult the user guide available in the download of the five macros.

Applying the five macros for reference-based multiple imputation entails the sequential run of the following macros. For example, for MNAR CR approach:

```
%part1A(jobname = HAMD,
        Data = dat,
        Subject = PATIENT,
        RESPONSE = CHANGE,
        Time = VISIT,
        Treat = THERAPY,
        Covbytime = BASVAL,
        Catcov = GENDER);
%part1B(jobname = HAMD, Ndraws = 5000, thin = 10, seed = 12345);
%part2A(jobname = HAMD_CR, inname = HAMD, method = CR, ref = PLACEB0);
%part2B(jobname = HAMD_CR, seed = 12345);
%part3(Jobname = HAMD_CR, anref = PLACEBO, Label = CR);
```

These macros do the following:

- Part1A declares the parameter estimation model and checks consistency with the dataset. It builds a master dataset which holds details of the current job (run of the macros in sequence). It also builds indexes for the classification variables, which may be either numeric or character.

- Part1B fits the parameter estimation model using the MCMC procedure and draws a pseudo-independent sample from the joint posterior distribution for the linear predictor parameters and the covariance parameters.

- Part2A calculates the predicted mean under MAR, and under MNAR for each subject based on their withdrawal pattern once for each draw of the linear predictor parameter estimates. The choice of MNAR is controlled by the method used.

- Part2B imputes the intermediate missing values using MAR and the trailing missing values using MNAR, by deriving the conditional distribution for the missing values conditional on the observed values and covariates, using the appropriate sampled covariance parameter estimates.
- Part3 carries out a univariate ANOVA analysis at selected time points usually based on the same covariates as the parameter estimation model. It then combines the least-squares means and their differences using the MIANALYZE procedure to provide results. It is in this macro which handles as well delta adjustment methods.

R VS SAS COMPARISON

The following table shows the results for all imputation strategies (MAR, MNAR CR, MNAR J2R, MNAR CIR) for M=5000 multiple imputation draws for the implementations in R and SAS. The table presents the contrast statistics at Visit 7 between DRUG and PLACEBO [DRUG - PLACEBO]. For completeness, we also present the results of a complete case analysis.

Method	Software	Point Estimate	Standard Error	95% CI	p-value
Complete case	R	-2.829	1.117	-5.035 to -0.622	0.0123
	SAS	-2.829	1.117	-5.035 to -0.623	0.0123
MAR	R	-2.830	1.123	-5.040 to -0.610	0.0128
	SAS	-2.825	1.123	-5.045 to -0.605	0.0130
MNAR CR	R	-2.394	1.112	-4.592 to -0.197	0.0329
	SAS	-2.400	1.115	-4.604 to -0.196	0.0330
MNAR J2R	R	-2.148	1.135	-4.390 to 0.095	0.0603
	SAS	-2.144	1.136	-4.389 to 0.101	0.0611
MNAR CIR	R	-2.479	1.112	-4.676 to -0.282	0.0273
	SAS	-2.481	1.115	-4.684 to -0.278	0.0276

MAR=Missing at Random; MNAR=Missing not at Random, CIR=Copy Increments in Reference; CR=Copy Reference; JR=Jump to Reference

Figure 4 below shows a further comparison of the contrast point estimates (and associated 95% confidence interval) from R and SAS for the explored dataset per number of multiple imputations M (500, 2000 and 5000). For the contrast estimate the range of the difference between R and SAS results are [-0.29 to 2.21]%, [0.0 to 0.75]% and [-0.25 to 0.19]% for M=500, 2000, 5000, respectively.

It can be observed in both the table and figure that the R rbmi package results and the SAS five macros results match. Results could be slightly different given the randomness of multiple imputation draws. It is important to note that these matching results are tested under the following assumptions:

- Equal unstructured covariance matrix across treatment groups
- Same covariates formula for the imputation and analysis model
- Similar number of MCMC tuning parameters (burn-in, thinning) was used in the MCMC
- The one intermittent missingness was imputed under MAR assumption

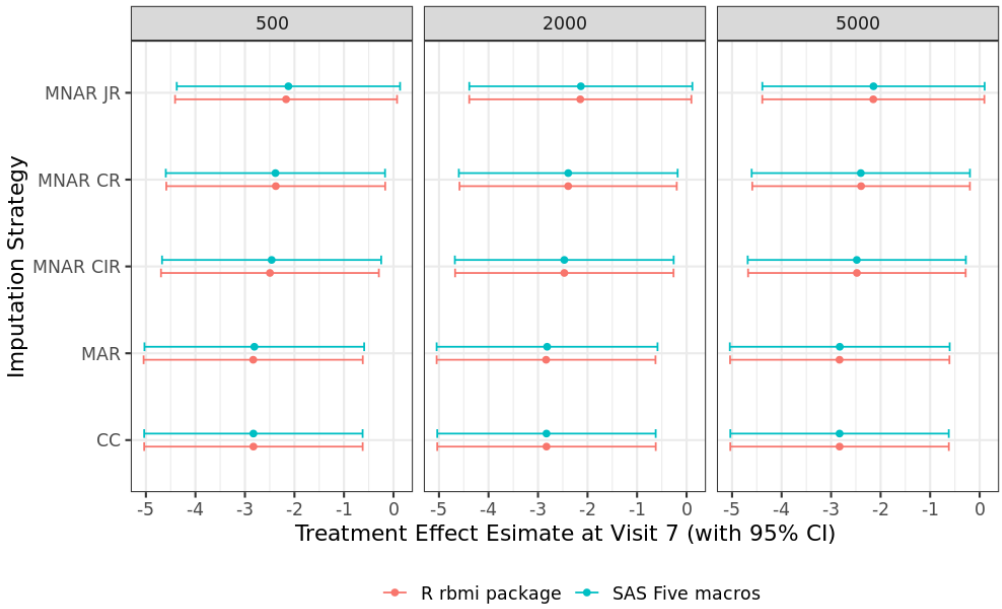

Figure 4: comparison of the treatment effect estimates (and associated 95% confidence interval) from R and SAS for the explored dataset per number of multiple imputations M (500, 2000 and 5000). CC=Complete Case; MAR=Missing at Random; MNAR=Missing not at Random, CIR=Copy Increments in Reference; CR=Copy Reference; JR=Jump to Reference.

TIPPING POINT ANALYSIS

When analyses for endpoints are performed under MAR or MNAR assumptions for missing data, it is important to perform sensitivity analyses to assess the impact of deviations from these assumptions. Tipping point analysis (or delta adjustment method) is an example of a sensitivity analysis that can be used to assess the robustness of a clinical trial when its result is based on imputed missing data (Cro et al. 2020).

Generally, tipping point analysis explores the influence of missingness on the overall conclusion of the treatment difference by shifting imputed missing values in the treatment group towards the reference group until the result becomes non-significant. The tipping point is the minimum shift needed to make the result non-significant. If the minimum shift needed to make the result non-significant is implausible, then greater confidence in the primary results can be inferred.

Tipping point analysis generally happens by adjusting imputing values by so-called delta values. The observed tipping point is the minimum delta needed to make the result non-significant. Mostly a range of delta values is explored and only imputed values from the active treatment group are adjusted by the delta value. However, delta adjustments in the control group are possible as well. We explored tipping point analysis for reference-based multiple imputation.

R IMPLEMENTATION

In the rbmi package[11], delta adjustments are implemented via the delta argument of the analyse() function. The adjustment happens right after data imputation under MAR or MNAR (using reference-based imputation approaches), but before implementing the analysis model. Sensitivity analyses can therefore be performed without having to refit the imputation model, which is computationally efficient. This approach is considered a *marginal* delta adjustment approach, because the delta is simply added to the mean of the conditional multivariate normal distribution (conditional on the observed values and the covariates) for the imputation model (Roger 2017).

We provide a summary of the implementation of delta adjustment analysis on the antidepressant data using the rbmi package in R. For the full code, visit the CAMIS website and/or consult the rbmi advanced functionality vignette.

1. We assume the user used the rbmi package to create an imputation object called imputeObj.

2. Create a data frame dat_delta using delta_template() that includes a column called delta that specifies the delta values to be added to which records (see Figures 6 and 7).

3. Use the analyse() function with the delta argument to implement the delta adjustment. Finally, use the pool() function to summarise the analysis results across multiple imputed datasets.

4. Repeat steps 2 and 3 over a range of delta values. The tipping point is the minimum shift needed to make the result non-significant.

Note 1: To decrease computation time, you may want to increase the number of parallel processes with make_rbmi_cluster().

Note 2: We advise writing functions to perform this task over a range of delta values. See CAMIS website.

Note 3: You can derive an *exact* tipping point by linearly interpolating between the last "significant" delta and the first "non-significant" delta using the approx() function.

A nice visualization of a tipping point analysis with sequential delta for the intervention group only is shown in Figure 5. The dashed horizontal line indicates a p-value = 0.05 on the left plot and no treatment effect on the right plot. We see that the p-value and treatment effect reach a tipping point from 3 onward in the range of deltas considered.

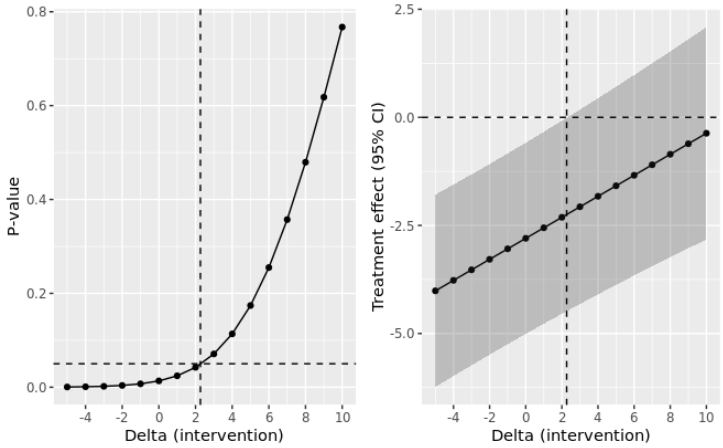

Figure 5: Tipping point analysis for MAR assumption with sequential delta_intervention (R)

DELTA ADJUSTMENTS IN R

As mentioned, the delta argument of the analyse() function allows users to adjust the imputed datasets prior to the analysis stage. This delta argument requires a data frame created by delta_template(), which includes a column called delta that specifies the delta values to be added. By default, delta_template() will set delta to 0 for all observations (see Figure 6).

PATIENT	VISIT	THERAPY	is_mar	is_missing	is_post_ice	strategy	delta
1513	4	DRUG	TRUE	FALSE	FALSE	NA	0
1513	5	DRUG	TRUE	TRUE	TRUE	MAR	0
1513	6	DRUG	TRUE	TRUE	TRUE	MAR	0
1513	7	DRUG	TRUE	TRUE	TRUE	MAR	0
1514	4	PLACEBO	TRUE	FALSE	FALSE	NA	0
1514	5	PLACEBO	TRUE	TRUE	TRUE	MAR	0
1514	6	PLACEBO	TRUE	TRUE	TRUE	MAR	0
1514	7	PLACEBO	TRUE	TRUE	TRUE	MAR	0

Figure 6: Default delta template (delta = 0)

Simple delta adjustments

The is_missing and is_post_ice columns of the delta data frame lend themselves perfectly to adjust the delta values, as the Boolean variables TRUE and FALSE are regarded as 1 and 0 by R. If you want to set delta to 5 for all missing values, you can do so by multiplying the is_missing column by 5. If you multiply this column only, you apply the delta adjustment to *all* imputed missing values, including intermittent missing values. If you consider the is_post_ice column too, you can restrict the delta adjustment to missing values that occur after study drug discontinuation due to an intercurrent event (ICE). Thus, by multiplying both the is_missing and is_post_ice columns by your chosen delta, the delta value will only be added when both columns are TRUE.

Besides choosing which missing data to apply the delta adjustment to, you may also want to apply different delta adjustments to imputed data from the different groups. As an example, we set delta = 0 for the control group, and delta = 5 for the intervention group (Figure 7).

PATIENT	VISIT	THERAPY	is_mar	is_missing	is_post_ice	strategy	delta	delta_ctl	delta_int
1513	4	DRUG	TRUE	FALSE	FALSE	NA	0	0	0
1513	5	DRUG	TRUE	TRUE	TRUE	MAR	5	0	5
1513	6	DRUG	TRUE	TRUE	TRUE	MAR	5	0	5
1513	7	DRUG	TRUE	TRUE	TRUE	MAR	5	0	5
1514	4	PLACEBO	TRUE	FALSE	FALSE	NA	0	0	0
1514	5	PLACEBO	TRUE	TRUE	TRUE	MAR	0	0	0
1514	6	PLACEBO	TRUE	TRUE	TRUE	MAR	0	0	0
1514	7	PLACEBO	TRUE	TRUE	TRUE	MAR	0	0	0

Figure 7: Simple delta template for first two patients (delta_intervention = 5, delta_control = 0)

Flexible delta adjustments

So far, we have only considered simple delta adjustments that add the same value to all imputed missing data. However, you may want to implement more flexible delta adjustments for post-ICE missing data, where the magnitude of the delta varies depending on the distance of the visit from the ICE visit. To facilitate the creation of such flexible delta adjustments, delta_template() has two additional arguments delta and dlag. The usage of these arguments is best illustrated in the advanced functionality vignette of the rbmi package.

Looking at the example data, suppose we apply a delta adjustment of 2 for each week following an ICE in the intervention group only. For example, if the ICE took place immediately after visit 4, then the cumulative delta applied to a missing value from visit 5 would be 2, from visit 6 would be 4, and from visit 7 would be 6. To program this, we first set the delta and dlag arguments to c(2, 2, 2, 2) and c(1, 1, 1, 1). Then, we use some metadata variables provided by delta_template() to manually reset the delta values for the control group back to 0.

PATIENT	VISIT	THERAPY	is_mar	is_missing	is_post_ice	strategy	delta
1513	4	DRUG	TRUE	FALSE	FALSE	NA	0
1513	5	DRUG	TRUE	TRUE	TRUE	MAR	2
1513	6	DRUG	TRUE	TRUE	TRUE	MAR	4
1513	7	DRUG	TRUE	TRUE	TRUE	MAR	6
1514	4	PLACEBO	TRUE	FALSE	FALSE	NA	0
1514	5	PLACEBO	TRUE	TRUE	TRUE	MAR	0
1514	6	PLACEBO	TRUE	TRUE	TRUE	MAR	0
1514	7	PLACEBO	TRUE	TRUE	TRUE	MAR	0

Figure 8: Flexible delta template for first two patients (delta_intervention = 2, 4, 6, 8; delta_control = 0)

Note on delta and dlag arguments:

You may also add a simple, fixed delta using the delta and dlag arguments. To do this, delta should be specified as a vector of length equal to the amount of visits, e.g. c(5, 5, 5, 5), while dlag should be c(1, 0, 0, 0). In this case, this ensures a delta of 5 is added to each imputed missing value following an ICE.

Adding a fixed delta in this way seems similar to what we explained in the *simple delta adjustments* section above, but there is a crucial difference: the delta and dlag arguments of delta_template() only apply delta adjustments to post-ICE missing values, and **not** to intermittent missing values. One should be aware of this discrepancy between delta adjustment approaches when using the rbmi package.

SAS IMPLEMENTATION

To conduct a tipping point analysis using the five macros[10], Part 1 and Part 2 are generally the same as in the scenario without any delta adjustment. Then, in Part 3, delta values can be added to the intervention group and/or control group by changing the Delta argument, setting Dlag = 1 0 0 0 and defining Dgroups based on what groups you want to apply the delta adjustments to. For the full code, visit the CAMIS website and/or consult the five macros user guide (Roger 2017).

DELTA ADJUSTMENTS IN SAS

Simple delta adjustments

To add a simple (fixed) delta, Delta should be the same value repeated for each visit, e.g. 5 5 5 5, while DLag should be 1 0 0 0. To apply this fixed delta to both groups we leave DGroups unspecified. If we only want to apply the delta value to the intervention group, one needs to specify DGroups = DRUG.

Flexible delta adjustments

Consider the same example as shown in R: a flexible delta adjustment of 2 for each week following an ICE in the intervention group only: This can be specified by setting Delta = 2 2 2 2, Dlag = 1 1 1 1 and DGroups = DRUG.

Importantly, in the five macros, delta adjustments are **not** applied to intermittent missing observations, but only to missing observations after withdrawal. From the documentation, it seems like this cannot be altered. In contrast, the choice of which missing data to apply delta adjustments to can be more freely managed in the rbmi package in R. This may lead to discrepancies between tipping point results in SAS and R.

R VS SAS COMPARISON

The following table shows the results for the different imputation strategies (M=5000) for tipping point analysis with delta adjustments in R and SAS. The table again presents the contrast statistics at Visit 7 between DRUG and PLACEBO. In R, delta adjustments were applied to *all* imputed missing data, while in SAS, delta adjustments were applied to all imputed missing data *except* intermittent missing data. There was one intermittent missing value in the entire dataset. In both languages, delta adjustments were applied in the intervention group only.

Method	Software	Delta at TP	Estimate at TP	95% CI	P-value
MAR	R	3	-2.100	-4.354 to 0.154	0.0675
	SAS	3	-2.096	-4.353 to 0.161	0.0684
MNAR CR	R	1	-2.162	-4.377 to 0.054	0.0558
	SAS	1	-2.157	-4.377 to 0.062	0.0566
MNAR J2R	R	-1	-2.383	-4.608 to -0.157	0.0361
	SAS	-1	-2.387	-4.617 to -0.157	0.0361
MNAR CIR	R	2	-1.994	-4.227 to 0.239	0.0796
	SAS	2	-1.995	-4.229 to 0.240	0.0798

MAR=Missing at Random; MNAR=Missing not at Random, CIR=Copy Increments in Reference; CR=Copy Reference; JR=Jump to Reference

Figures 9 and 10 compare the treatment effect estimates at Visit 7 and corresponding p-values (y-axis) for each increasing delta in the intervention group (x-axis), while the delta adjustment in the control group is fixed to zero. The black crosses indicate *exact* tipping points as determined by linear interpolation. There seems to be a very good correspondence between R and SAS at M = 500, and near perfect correspondence between R and SAS at M = 5000.

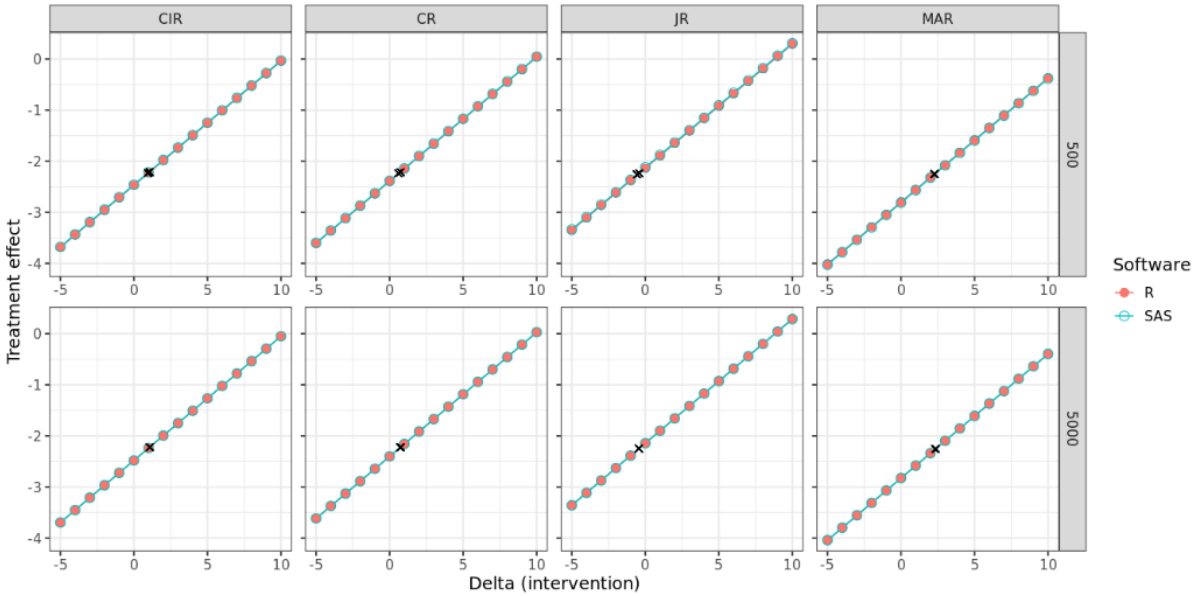

Figure 9: Comparison of tipping point analysis of treatment effect across imputation strategies and number of imputation draws M

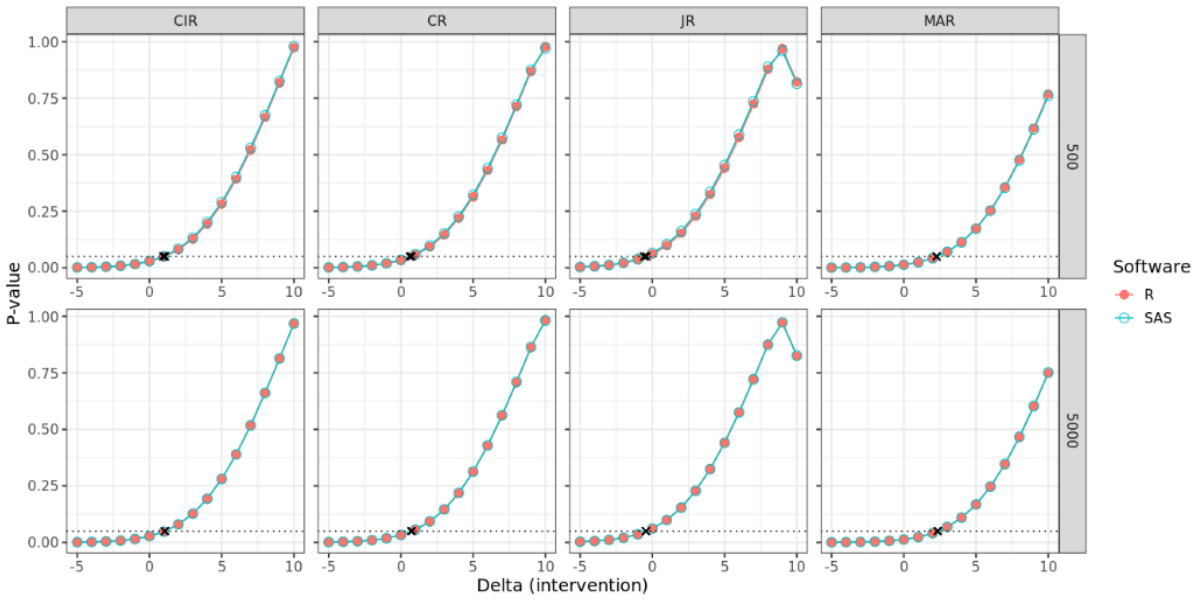

Figure 10: Comparison of tipping point analysis of p-value across imputation strategies and number of imputation draws M

CONCLUSION

For the three statistical analyses presented in this paper matching results are found between SAS and R. In case, a slight difference was found this was due to algorithmic variation and Monte Carlo simulation error.

In the transition from proprietary to open-source technology in the industry, CAMIS can serve as a guidebook to navigate this process. Knowing the reasons for differences (different methods, options, algorithms, etc.) and understanding how to mimic analysis results across software is critical to the modern statistician and subsequent regulatory submissions.

CALL TO ACTION!

You have the opportunity to contribute and help the CAMIS community as well. Interested to join? Please contact us:

Christina Fillmore: christina.e.fillmore@gsk.com
Lyn Taylor: lyn.taylor@parexel.com
Yannick Vandendijck: yvanden2@its.jnj.com

REFERENCES

1. CAMIS (2025), https://psiaims.github.io/CAMIS/
2. Rimler MS, Rickert J, Jen M, Stackhouse M (2022). Understanding differences in statistical methodology implementations across programming languages. Biopharmaceutical Report, 29 (3).
3. FDA (May 6, 2015). Statistical Software Clarifying Statement.
4. Jen M, Varney B, Lee K, Arancibia B, Qi M, Taylor L, Fillmore CE, Rickert J, Stackhouse M, Rimler M (2023). Key Considerations When Understanding Differences in Statistical Methodology Implementations Across Programming Languages – An Introduction to the CAMIS Project. Doc ID: WP-077.
5. Tobin J (1958). Estimation of Relationships for Limited Dependent Variables. Econometrica. 26 (1): 24-36. doi:10.2307/1907382.
6. Breen, R (1996). Regression models. SAGE Publications, Inc., https://doi.org/10.4135/9781412985611.
7. Carpenter JR, Roger JH & Kenward MG (2013). Analysis of Longitudinal Trials with Protocol Deviation: A Framework for Relevant, Accessible Assumptions, and Inference via MI. Journal of Biopharmaceutical Statistics 23: 1352-1371.
8. Rubin DB (1976). Inference and Missing Data. Biometrika 63: 581–592.
9. Rubin DB (1987). Multiple Imputation for Nonresponse in Surveys. New York: John Wiley & Sons.
10. Roger J (2022, Dec 8). Other statistical software for continuous longitudinal endpoints: SAS macros for multiple imputation. Addressing intercurrent events: Treatment policy and hypothetical strategies. Joint EFSPI and BBS virtual event.
11. Gower-Page C, Noci A & Wolbers M (2022). rbmi: A R package for standard and reference-based multiple imputation methods. Journal of Open Source Software 7(74): 4251.
12. Cro S, Morris TP, Kenward MG, Carpenter JR (2020). Sensitivity analysis for clinical trials with missing continuous outcome data using controlled multiple imputation: A practical guide. Statistics in Medicine. 2020;39(21):2815-2842.
13. Roger J (2017). Fitting reference-based models for missing data to longitudinal repeated-measures Normal data. User guide five macros.

ACKNOWLEDGMENTS

We thank all CAMIS contributors!

CONTACT INFORMATION

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

Author Name: Yannick Vandendijck

Company: Johnson & Johnson

Email: yvanden2@its.jnj.com

Brand and product names are trademarks of their respective companies.

Content reproduced with the permission of PHUSE CAMIS - A DVOST Working Group.