

Pattern-Mixture Models as Linear Combinations of Least Squares Means from MMRM with Delta Method Variance Estimation

Bohdana Ratitch, Quintiles, Montreal, Quebec, Canada

Michael O'Kelly, Quintiles, Dublin, Ireland

ABSTRACT

For studies with missing data, it is being increasingly recommended to provide sensitivity analyses that assume data are Missing Not at Random (MNAR). In certain therapeutic areas, MNAR assumptions about the missingness mechanism are even recommended for the primary analysis. MNAR assumptions can be modeled within several statistical frameworks, one of which is known as pattern-mixture models (PMMs). Certain PMM-based analyses for continuous outcomes can be formulated in such a way that the estimate of the difference between experimental and control treatments is expressed as a linear combination of Least Squares Means (LSMs) for different effects of a longitudinal model with correlated errors, weighted by the appropriate proportions of study drop-outs and completers. This approach requires special considerations for the estimation of the variance because the proportions of drop-outs and completers used in the linear combination of LSMs are themselves multinomial random variables and their variances need to be incorporated into the overall estimate. This can be done using a delta approximation method for variance estimation. In this paper, we present details of implementing such analyses (including delta variance estimation method) using exclusively SAS/STAT® core functionality, such as PROC MIXED, data steps, and PROC FCMP. To illustrate this approach, we are using an example of MNAR assumptions that take into account the reasons for discontinuation from the study.

1. INTRODUCTION

Longitudinal models with correlated errors, otherwise widely known as mixed models with repeated measures (MMRMs), have been increasingly used for the analysis of clinical trials with missing data. A longitudinal model is often used even though the primary objective is to estimate a treatment effect and test a null hypothesis of no treatment effect at a single specific time-point (typically at the end of double-blind period). The advantage of using an MMRM analysis in this context (compared to ANCOVA at the primary time-point) is that longitudinal models include all randomized subjects regardless of whether they completed the study (provided data for the primary time-point) or not. Model estimation and inference is done without performing any imputation of the missing data for subjects who discontinued early, yet partial data available for these subjects is fully utilized and contributes to the estimation of effects at the end of the double blind period and to the variance-covariance structure of the longitudinal model.

A standard MMRM analysis relies on Missing at Random (MAR) assumptions regarding the mechanism behind missingness of the data. With MAR missingness depends only on the available data and is independent of the outcomes that are not observed. Under the MAR assumption, as long as a statistical model includes all observed variables that explain missingness, inference from that model will not be biased due to the missing data. In less technical terms, MAR assumptions imply the following: that the subjects with missing data (typically those that discontinue from the trial prematurely) follow the same trajectory (distribution) for the outcome variable after their discontinuation as the subjects with available data (study completers) with similar characteristics captured by observed data. If, on the other hand, missingness depends on the unobserved outcome and cannot be fully explained by available data, the mechanism is considered to be MNAR. In other words, with MNAR it is assumed that the post-discontinuation outcomes of subjects with missing data are systematically different from those with observed data even after accounting for all relevant observed factors in the statistical model.

While MAR assumptions may be reasonable for many studies, one can never rule out that the missingness mechanism for a study at hand is not MNAR (it is impossible to prove that missingness does not depend on unobserved data due to the fact that these data are missing). Because of this, regulatory and industry experts strongly recommend [1,2] that sensitivity analyses be performed for departures from MAR assumptions. In practice this requires analyzing the data under various MNAR scenarios that are clinically meaningful and plausible for a study at hand. In some cases, MAR assumptions can be considered as just not appropriate for the study, and MNAR assumptions have to be used for the primary analysis. We provide one example of MNAR assumptions in Section 2.

Importantly, we cannot be sure of the dependencies, if any, between observed data values and missing data values, so a typical solution for this problem is to make some explicit assumptions regarding these dependencies or about the distribution of the unobserved outcomes, and impose these assumptions on the statistical model.

Pattern-mixture models [3] provide one framework within which MNAR assumptions can be modeled and we provide a brief introduction to this framework in Section 2. There are different ways in which PMMs can be implemented [12]. One implementation strategy relies on Multiple Imputation (MI) (e.g., see reference [11] for a description of a general approach and reference [4] for an example of a particular PMM using MI functionality in SAS). Other approaches use maximum likelihood techniques as the basis for the estimation of an underlying statistical model, onto which additional (MNAR) assumptions are imposed [5, 7, 13]. In Section 2 of this paper, we describe one way of implementing PMMs in a maximum likelihood framework in SAS for the analysis of continuous outcomes. Unlike MI-based implementations, this approach does not involve any explicit imputation, but similar to standard MMRM analysis, partial data from subjects that discontinued early are used for estimation of the parameters in the longitudinal model. With this strategy, the estimate of the experimental-control treatment difference is formulated as a linear combination of LSMs obtained from a standard MMRM, where each LSM in the linear combination can be weighted by proportions of subjects in different completer/drop-out groups. This estimate can be obtained using SAS procedure PROC MIXED. However, PROC MIXED does not compute a standard error for this estimate in a way that would account for the fact that weights used in the linear combination of LSMs are not constants, but rather multinomial random variables (proportions of subjects in different subgroups). Since these proportions are estimated from data, standard errors associated with their estimation need to be accounted for in the overall standard error of the treatment difference estimate. This can be accomplished using a so called delta approximation method [6]. We provide a general algorithm for its calculation in Section 3 with a detailed example using SAS code in Section 5.

A maximum likelihood-based approach may be appealing to statisticians that prefer the “single answer” provided by an MMRM-type analysis to the random sampling (simulation) nature of an MI-based analysis, and who may have other concerns about MI, such as treatment of non-monotone missing data. It should be noted, however, that even when the underlying MNAR assumptions are the same (in terms of their clinical interpretation), MMRM-based implementation of PMMs and implementations based on explicit imputation (both single and multiple) can result in slightly different estimates of the mean difference between treatments as well as quite different standard errors of these estimates. We will further highlight this fact in Section 5.

The main objective of this paper is to demonstrate how PMM-type analyses based on linear combinations of LSMs from an MMRM can be implemented in SAS, and how standard errors for the corresponding estimates of the experimental-control treatment differences can be computed with the delta method using standard SAS functionality. The approach is illustrated using, as an example, a particular set of MNAR assumptions based on reasons for discontinuation from the study which results in a particular instance of a PMM and a corresponding linear combination of LSMs. However, general principles and implementation strategy described in Sections 2 and 3 can be applied to impose other MNAR assumptions as long as the final estimate of the difference between treatments can be formulated as a linear combination of LSMs obtained from a longitudinal model. This article does not suggest that the set of MNAR assumptions and the way of implementing them as illustrated in this paper have some general advantages over other methods, including those results of which are provided in Section 5. The choice of the assumptions and the most appropriate method to implement them depends on various characteristics of the real-world studies and would have to be considered on a case-by-case basis. We highlight some of the pros and cons of the MMRM-based and MI-based methods for PMM implementation in the conclusion section.

2. PATTERN-MIXTURE MODELS AS LINEAR COMBINATIONS OF LEAST SQUARES MEANS

PMMs are based on a joint modeling of (clinical) outcome and missingness. Assume that the entire outcome data matrix Y is divided into two components: Y_{obs} , representing observed outcomes and Y_{mis} , representing missing (unobserved) outcomes. Assume that X represents a set of (observed) covariates and R represents a matrix of indicators of missingness. PMMs use the following factorization of the joint probability of data and missingness:

$$p(Y_{obs}, Y_{mis}, R|X) = p(R|X) p(Y_{obs}, Y_{mis} | R, X) = p(R|X) p(Y_{obs} | R, X) p(Y_{mis} | Y_{obs}, R, X) \quad (2.1)$$

The indicators of missingness R can also be regarded as indicators of belonging to a particular pattern of missingness or, in other words, a group of subjects that share certain characteristics related to missingness, for example, time of discontinuation from the clinical study, reason for discontinuation, or a treatment arm to which subjects were randomized. The probability distribution $p(R|X)$ can be viewed as a probability distribution of various missingness patterns. The factor $p(Y_{obs} | R, X)$ represents a model for available data within each pattern R , and $p(Y_{mis} | Y_{obs}, R, X)$ represents a model for missing data conditioned on observed data within each pattern. The pattern-specific estimates are not typically of interest, so the average estimates across the missing data patterns are obtained to yield an overall result. Averaging is accomplished by the weighting of $p(Y_{obs} | R, X) p(Y_{mis} | Y_{obs}, R, X)$ with probabilities $p(R|X)$.

Pattern-mixture models are, by definition, under-identified because the probability $p(Y_{mis} | Y_{obs}, R, X)$ cannot be estimated without assumptions. In order to estimate PMMs, one needs to explicitly impose those assumptions, which are sometimes referred to as “identifying restrictions”, regarding the inestimable parameters. There are several methodologies for imposing these identifying restrictions.

In this paper, we consider an approach where an MMRM model is first estimated from observed data in a standard MAR-type modeling framework with treatment arm, time, treatment-by-time interaction, and possibly some covariates included as effects. From this model, one can estimate various LSMs for outcome in different treatment arms and at different time-points. Subsequently, in order to obtain LSMs under departures from the MAR assumptions used to estimate the above model, one can impose certain identifying restrictions in such a way that LSMs for drop-out groups (patterns) are assumed be equal to specific LSMs from the MAR-based model reflecting desired clinical assumptions. An overall estimate of the treatment effect (overall LSM over completers and drop-outs) is then defined as a linear combination of LSMs for different patterns weighted by the probability of subjects belonging to each respective pattern.

We will illustrate this approach using an analysis strategy the main features of which has been recommended in the past for trials with symptomatic treatment of chronic conditions (e.g., chronic pain). In such indications, regulatory agencies recommended the adoption of an approach where subjects who discontinue study treatment due to inability to tolerate it are assumed to revert to their baseline severity of symptoms and derive no further benefit from treatment after discontinuation. When single imputation methods were used in conjunction with ANCOVA, for this group of discontinued subjects, the outcome at the final time-point was imputed using the baseline observation carried forward (BOCF) method, while other subjects were imputed using last observation carried forward (LOCF). Using BOCF for those who discontinued due to intolerability resulted in a more conservative estimate of treatment effect, compared to using LOCF universally, because sometimes these subjects (often more numerous in the experimental treatment arm) did exhibit a considerable symptom relief prior to discontinuation, which would have been carried forward by the LOCF method to the final time-point, thus attributing a positive long-term treatment effect, contrary to regulatory recommendations and clinical objectives. However, a general concern with single imputation methods (both BOCF and LOCF) is that they can potentially underestimate the variance of the treatment effect estimate by not accounting for uncertainty about missing (imputed) data.

In this paper, we demonstrate how similar clinical assumptions can be modeled with MMRM-based PMM analysis as opposed to a hybrid single imputation with BOCF and LOCF. Assume that there are $K+1$ visits in a study, with 0 being a baseline visit and K being a final time-point used for the primary endpoint. For simplicity, we assume that there are two treatment arms – experimental and control. For the purposes of our analysis, we will consider two types of primary reasons leading to subjects discontinuing the study treatment: adverse events and lack of efficacy, which we will designate together as lack of therapeutic benefit (LTB), vs. other reasons. The primary reasons of discontinuation are assumed to be collected as part of the study data and are identified by the study investigator. A standard MMRM model will include time (with baseline included as a separate time-point), treatment arm and treatment-by-time interaction as fixed effects and baseline efficacy score as covariate. It can be implemented in SAS using PROC MIXED with REPEATED statement (time as repeated factor within subject).

Assume that $\mu_{t,k}$ represents an LSM estimated from this MMRM for a specific treatment arm and time-point, where subscript t denotes the treatment arm ($t=0$ for control and $t=1$ for experimental), and subscript k denotes the time-point ($k=0,1,\dots,K$). We will also define the following LSMs for each treatment arm t at the last time-point K , which is our primary analysis time-point.

- $\mu_{t,K}^C = \mu_{t,K}$: LSM that will be attributed to study completers, as denoted by the superscript C. For completers, the LSM is assumed to be equal to the LSM $\mu_{t,K}$ estimated from the MMRM model for the final time-point in each arm.
- $\mu_{t,K}^{LTB} = \mu_{t,0}$: LSM that will be attributed to drop-outs due to lack of therapeutic benefit, as denoted by the superscript LTB. For this group of drop-outs, the LSMs for the last post-baseline time-point K are assumed to be equal to the LSMs at baseline, $\mu_{t,0}$, thus not attributing any positive treatment effect as would, otherwise, be estimated by a post-baseline estimate of $\mu_{t,K}$ from the MMRM model.
- $\mu_{t,K}^{OTH} = \frac{1}{2}(\mu_{t,1} + \mu_{t,2}), k \geq 1$: LSM that will be attributed to drop-outs due to other reasons, as denoted by the superscript OTH. For this group of drop-outs, the LSMs for the last post-baseline time-point K are assumed to be equal to an average of LSMs at the first two post-baseline visits in the respective treatment arms. This way, the LSMs at the final time-point are effectively replaced by the LSMs from the earlier time-points to which these discontinued subjects contributed their data most. (Note, this assumption may be made more general and expressed as an average of all post-baseline visits except the final one).

Based on the LSMs defined above, an overall LSM, $\mu_{t,K}^A$, for a treatment arm t at the final time-point K (over completers and drop-outs as denoted by superscript A) is defined as follows.

$$\begin{aligned} \mu_{t,K}^A &= \pi_{t,K}^C \times \mu_{t,K}^C + \pi_{t,K}^{LTB} \times \mu_{t,K}^{LTB} + \pi_{t,K}^{OTH} \times \mu_{t,K}^{OTH} = \\ &= \pi_{t,K}^C \times \mu_{t,K} + \pi_{t,K}^{LTB} \times \mu_{t,0} + \pi_{t,K}^{OTH} \times \frac{1}{2}(\mu_{t,1} + \mu_{t,2}) \end{aligned} \quad (2.2)$$

where $\pi_{t,K}^C$, $\pi_{t,K}^{LTB}$, and $\pi_{t,K}^{OTH}$ represent proportions of study completers and of drop-outs due to LTB and other reasons respectively, which can be estimated from observed frequencies of completers and drop-outs. As can be noted from

(2.2), this overall LSM for each treatment arm can be estimated using a linear combination of LSMs, $\mu_{t,k}$, obtained from a standard MAR-based MMRM; yet the overall LSMs $\mu_{t,k}^A$ reflect departures from the original MAR assumptions.

Once the overall LSMs are computed as defined in (2.2) for each treatment arm, a difference between LSMs of the two treatments can be estimated as

$$\mu_{d,K}^A = \mu_{1,K}^A - \mu_{0,K}^A \quad (2.3)$$

Then, a test of the null hypothesis of no treatment effect can be performed using a Wald statistic

$$\mu_{d,K}^A / SE(\mu_{d,K}^A) \quad (2.4)$$

where $SE(\mu_{d,K}^A)$ is a standard error of the $\mu_{d,K}^A$ estimate. A special consideration should be given to the estimation of this standard error. The estimate should take into account the fact that the proportions $\pi_{t,K}^C$, $\pi_{t,K}^{LTB}$, and $\pi_{t,K}^{OTH}$ are estimates of the multinomial random variables, and thus their variances should be incorporated in the overall variance estimate. This can be done using a delta approximation method for variance estimation [6,7]. In the next section, we describe the delta method and how to implement it using information readily available from PROC MIXED.

3. DELTA APPROXIMATION METHOD FOR VARIANCE ESTIMATION

Assume an LSM of interest μ (e.g., $\mu_{d,K}^A$ as described in the previous section) is expressed as a linear combination of LSMs $\mu_1, \mu_2, \dots, \mu_P$ from a standard MMRM:

$$\mu = \pi_1 \times \mu_1 + \pi_2 \times \mu_2 + \dots + \pi_P \times \mu_P \quad (3.1)$$

where π_p are proportions of subjects in specific groups (patterns) defined, for example, based on a combination of treatment arm, study completion status, and reason for discontinuation. LSMs μ_p ($p = 1, \dots, P$) are estimated from an MMRM with a vector of parameter estimates $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_M)$. Each of the LSMs μ_p can be expressed in terms of model parameters and contrast coefficients c_m^p :

$$\mu_p = c_1^p \times \beta_1 + c_2^p \times \beta_2 + \dots + c_M^p \times \beta_M \quad (3.2)$$

Variance of the overall LSM, $Var(\mu)$, can be estimated using a delta approximation method [6,7] as follows.

$$Var(\mu) = \nabla\mu [VarCov(\boldsymbol{\beta}) \oplus VarCov(\boldsymbol{\pi})] (\nabla\mu)' \quad (3.3)$$

where $\nabla\mu$ is a gradient vector of μ consisting of partial derivatives of μ with respect to model parameters β_m and proportions π_p . The elements of this gradient vector are as follows:

$$\nabla\mu = \left(\frac{\partial\mu}{\partial\beta_1}, \frac{\partial\mu}{\partial\beta_2}, \dots, \frac{\partial\mu}{\partial\beta_M}, \frac{\partial\mu}{\partial\pi_1}, \frac{\partial\mu}{\partial\pi_2}, \dots, \frac{\partial\mu}{\partial\pi_P} \right) \quad (3.4)$$

Variance-covariance matrix $VarCov(\boldsymbol{\beta})$ can be obtained directly from the estimates of the MMRM.

Variance-covariance matrix $VarCov(\boldsymbol{\pi})$ can be computed as variances and covariances of sampling proportions from a multinomial distribution:

$$\text{Diagonal elements (variances)} \quad VarCov(\boldsymbol{\pi})[i, i] = \pi_i \frac{1-\pi_i}{N}; \quad i = 1, \dots, P \quad (3.5)$$

where N is the total number of subjects in the sample based on which π_i are estimated (e.g., number of subjects in a given treatment arm).

$$\text{Off-diagonal elements (covariances)} \quad VarCov(\boldsymbol{\pi})[i, j] = -\frac{\pi_i\pi_j}{N}; \quad i, j = 1, \dots, P, i \neq j \quad (3.6)$$

Notation $[VarCov(\boldsymbol{\beta}) \oplus VarCov(\boldsymbol{\pi})]$ is used to indicate that the overall matrix is block-diagonal:

$$\begin{bmatrix} VarCov(\boldsymbol{\beta}) & \mathbf{0} \\ \mathbf{0} & VarCov(\boldsymbol{\pi}) \end{bmatrix} \quad (3.7)$$

Proportions π_p are estimated from a sample of study subjects as $\frac{n_p}{N}$, i.e., the number of subjects in group p divided by a total number of subjects N over all categories (typically within each treatment arm).

To facilitate the computation of $Var(\mu)$ as defined by equation (3.3), we will use the following data structure that can be represented and manipulated as a regular SAS dataset.

Table 1: Data Structure with Elements for Delta Method Variance Calculation

Model Effect	Model Parameter Estimate	μ_1 LSM Coefficients	π_1 Estimate	...	μ_P LSM Coefficients	π_P Estimate	$\frac{\partial \mu}{\partial \beta_m}$
<Model Term, Level>	β_1	c_1^1	π_1	...	c_1^P	π_P	$\frac{\partial \mu}{\partial \beta_1}$
<Model Term, Level>	β_2	c_2^1	π_1	...	c_2^P	π_P	$\frac{\partial \mu}{\partial \beta_2}$
			
<Model Term, Level>	β_M	c_M^1	π_1	...	c_M^P	π_P	$\frac{\partial \mu}{\partial \beta_M}$
	$\frac{\partial \mu}{\partial \pi_p}$		$\frac{\partial \mu}{\partial \pi_1}$...		$\frac{\partial \mu}{\partial \pi_p}$	

The elements of the gradient vector $\nabla \mu$ are represented by the last column and last row of this data structure. They can be computed as follows using other values from this data structure.

$$\frac{\partial \mu}{\partial \beta_m} = \sum_{p=1}^P c_m^p \times \pi_p \quad (3.8)$$

$$\frac{\partial \mu}{\partial \pi_p} = \sum_{m=1}^M c_m^p \times \beta_m \quad (3.9)$$

PROC MIXED can be used to obtain parameter estimates $\beta = (\beta_1, \beta_2, \dots, \beta_M)$ for the MMRM. We can use LSMESTIMATE statement in PROC MIXED to request the LSMs μ_p and the overall LSM μ , the result of which will be stored in the ODS output dataset LSMESTIMATES. Specification of LSMs is made easier with the new non-positional syntax of LSMESTIMATE statement [8]. When this simplified syntax is used (see examples in Section 5), option E allow us to obtain automatically all the contrast coefficients c_m^p (corresponding to a positional syntax with the full model parameterization) for the requested LSMs. These coefficients can be captured in the ODS dataset COEF. Parameter estimates β_m , and their variance-covariance matrix $VarCov(\beta)$ can also be readily obtained as ODS output datasets produced by PROC MIXED (SOLUTIONF and COVB respectively).

If all the necessary information is pulled from the output ODS datasets produced by PROC MIXED and stored in a dataset as described above, calculation of partial derivatives using equations (3.8) and (3.9) can be done in a SAS data step using arrays to compute $\frac{\partial \mu}{\partial \beta_m}$ on each row of the dataset, and using RETAIN statements to calculate $\frac{\partial \mu}{\partial \pi_p}$ in columns. Partial derivatives computed in this way can be transferred into a row vector (3.4) and stored as SAS dataset with one record and M+P variables.

Then matrix (3.7) needs to be constructed using ODS output dataset COVB for the upper left block and performing calculations in equations (3.5 and (3.6) for the lower right block. This matrix can be stored as a SAS dataset as well.

After that, the only thing that remains to be done is to perform vector and matrix multiplication operations according to equation (3.3). This can be done either by using PROC IML or implemented with core SAS functionality. In Appendix 1, we provide a macro that performs these operations using SAS/STAT procedure PROC FCMP.

This section presented a general definition and data structures needed to compute the variance of a linear combination of LSMs using the delta approximation method. In Section 5, we provide a detailed example and SAS code for the analysis based on the MNAR assumptions based on reasons for subject discontinuation described in the previous section.

4. EXAMPLE DATASET

Implementation of a PMM-based analysis will be illustrated using an example study where efficacy endpoint is stored in a SAS dataset, DATAIN, with the following variables:

- SUBJID – subject identification number;
- TRT – treatment arm (0=control and 1=experimental);

- DISCRSNN – completion or reason for discontinuation (0=completer, 1=lack of therapeutic benefit, including AEs or lack of efficacy, and 2=other reason for discontinuation);
- TIMEPTN – timepoint (0=Baseline, 1, 2,3,4, 5 for post-baseline study visits 1 through 5);
- SCORE – a continuous efficacy score, where higher values correspond to better outcomes;
- SCOREB – efficacy score at baseline

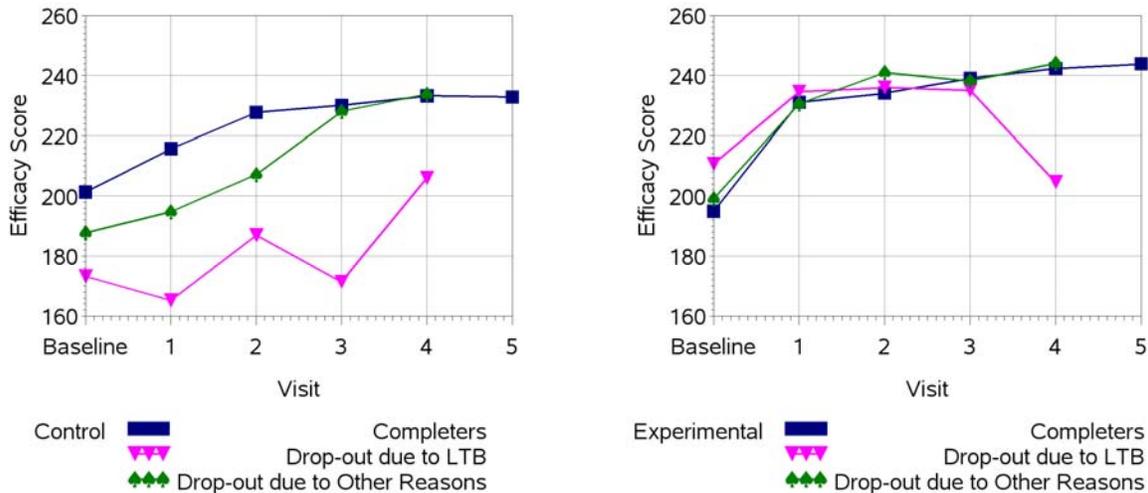
In this example dataset, percentages of drop-outs are very similar between the control and experimental arms, as well as across categories of reasons for discontinuations considered here, with just a slightly higher percentage of subjects discontinuing due to LTB from the control arm (see Table 2).

Table 2: Percentage of Subjects Completing and Discontinuing from the Study

Reason for Discontinuation	Discontinued Subjects (%)	
	Control Arm	Experimental Arm
Completers	79%	80%
Lack of Therapeutic Benefit	7%	5%
Other Reason for Discontinuation	14%	15%

Figure 1 shows mean efficacy scores across visits plotted separately for study completers and two groups of drop-outs. Subjects dropping out due to reasons other than LTB, follow a similar trajectory to completers. Subjects dropping out due to LTB from the control group showed some improvement but a less steady one compared to completers. Drop-outs due to LTB from the experimental arm, although improved between baseline and Visit 1, show a subsequent plateau and deterioration prior to discontinuation.

Figure 1: Mean Efficacy Scores by Visit and Completer/Drop-out Groups



5. SAS CODE AND RESULTS FOR THE EXAMPLE DATASET

In this section we illustrate how to perform the PMM analyses described in Section 2 and 3 for the example dataset introduced in Section 4.

The overall difference between the experimental and control treatments based on assumptions expressed by equations (2.2) and (2.3) is defined as follows:

$$\mu_{d,5}^A = \pi_{1,5}^C \times \mu_{1,5} + \pi_{1,5}^{LTB} \times \mu_{1,0} + \pi_{1,5}^{OTH} \times \frac{1}{2}(\mu_{1,1} + \mu_{1,2}) - \pi_{0,5}^C \times \mu_{0,5} - \pi_{0,5}^{LTB} \times \mu_{0,0} - \pi_{0,5}^{OTH} \times \frac{1}{2}(\mu_{0,1} + \mu_{0,2}) \quad (5.1)$$

We will employ the following SAS-friendly notation (which we will later use in SAS code) for the elements of equation (5.1):

- $\mu_{1,5} =$ mu1: LSM for experimental arm at time-point 5 (post-baseline Visit 5)
- $\mu_{1,0} =$ mu2: LSM for experimental arm at time-point 0 (baseline)
- $\frac{1}{2}(\mu_{1,1} + \mu_{1,2}) =$ mu3: average of LSMs for experimental arm at time-points (post-baseline Visits) 1 and 2
- $-\mu_{0,5} =$ mu4: negative of LSM for control arm at time-point 5 (post-baseline Visit 5)
- $-\mu_{0,0} =$ mu5: negative of LSM for control arm at time-point 0 (baseline)
- $-\frac{1}{2}(\mu_{0,1} + \mu_{0,2}) =$ mu6: negative of average of LSMs for control arm at time-points (post-baseline Visits) 1 and 2
- $\pi_{1,5}^C =$ &pi1: proportion of subjects in the experimental arm completing study through the final time-point 5
- $\pi_{1,5}^{LTB} =$ &pi2: proportion of drop-outs from the experimental arm by time-point 5 due to LTB
- $\pi_{1,5}^{OTH} =$ &pi3: proportion of drop-outs from the experimental arm by time-point 5 due to other reasons
- $\pi_{0,5}^C =$ &pi4: proportion of subjects in the control arm completing study through the final time-point 5
- $\pi_{0,5}^{LTB} =$ &pi5: proportion of drop-outs from the control arm by time-point 5 due to LTB
- $\pi_{0,5}^{OTH} =$ &pi6: proportion of drop-outs from the control arm by time-point 5 due to other reasons

Proportions of completers and drop-outs will be assumed to be stored in SAS macro variables &pi1 through &pi6, computation of which can be facilitated by a macro %nobscount provided in Appendix 1 which counts the number of observations in a specified subset of records from an input dataset.

Using this notation, the LSM difference between treatment arms ($\mu_{d,5}^A$) at the final time-point 5 can be expressed as follows:

$$\text{mu_diff_PMM} = \&\text{pi1} * \text{mu1} + \&\text{pi2} * \text{mu2} + \&\text{pi3} * \text{mu3} + \&\text{pi4} * \text{mu4} + \&\text{pi5} * \text{mu5} + \&\text{pi6} * \text{mu6} \quad (5.2)$$

This lends itself easily to the general notation used in equation (3.1), so that we can now follow equations (3.2) – (3.7) to calculate the variance of mu_diff_PMM using the delta method.

SAS Code Fragment 1 illustrates how the LSMs involved in the linear combination (5.2) can be computed using PROC MIXED and LSMESTIMATE statements. In this procedure call, we are using a repeated measures model for the efficacy score variable SCORE with a repeated factor SUBJID and a Toeplitz type covariance matrix for correlated errors. Model includes treatment (TRT), time-point (TIMEPTN), and TRT*TIMEPTN interaction as fixed effects as well as baseline score SCOREB as covariate. (The interaction of baseline with time could also be included.) Data includes all 6 levels of TIMEPTN (baseline plus 5 post-baseline visits). Including baseline time-point allows us to estimate LSM of each treatment arm at baseline (mu2 and mu5) from the TRT*TIMEPTN effect at TIMEPTN=0.

We are using a non-positional syntax in the LSMESTIMATE statement [8,10] which allows us to request contrast (LSM) estimates in a format where only non-zero coefficients involved in the contrast are specified, without regard to the ordering of parameters in the model parameterization. For example, when requesting an LSM estimate for the fixed effect TRT*TIMEPTN, the format would be as follows:

```
LSMESTIMATE TRT*TIMEPTN '<Label>' [<coefficient>, <level# of TRT> <level# of TIMEPTN>] /
<options>;
```

Note that “level #” for each factor indeed refers to the level and not the value of the factor. For example, for TRT which takes values 0 and 1, to estimate LSM for the control arm, level # is 1, and for the experimental arm, level # is 2. In order to estimate LSM of the experimental arm at Visit 5, values in square brackets would be [1, 2 6], indicating level #2 for TRT and level #6 for TIMEPTN. Nothing else would need to be specified, as this would be the only term with a non-zero coefficient in the corresponding contrast. In the code below, labels in each LSMESTIMATE statement describe the LSM that is calculated using this non-positional syntax.

We can use the output from this PROC MIXED to compute variance of the LSM difference between treatments as described in equations (3.2) – (3.7). We will pull information from different ODS output datasets to construct a dataset containing information described in Table 1.

Model parameter estimates $\beta = (\beta_1, \beta_2, \dots, \beta_M)$ are located in the ODS output dataset SOLUTIONF. From this dataset, we need to use the variables as shown in the left panel of Table 3, where the values of the ESTIMATE variable contain parameters β_m for each model effect identified by EFFECT variable and levels of CLASS effects TRT and TIMEPTN.

SAS CODE Fragment 1

```

ODS OUTPUT CovParms=CovParms CovB=CovB SolutionF=SolutionF Coef=Coef
LSMEstimates=LSMEstimates;
PROC MIXED DATA=DATAIN METHOD=reml EMPIRICAL;
  CLASS SUBJID TRT TIMEPTN;
  MODEL SCORE = TRT TIMEPTN TRT*TIMEPTN SCOREB/ SOLUTION COVB;
  REPEATED timeptn / TYPE=toep SUBJECT=SUBJID;

* Non-positional syntax for LSMs: TRT*TIMEPTN [<coeff>,<TRT_level#, TIMEPTN_level#>];
LSMESTIMATE TRT*TIMEPTN '1. mu1: Experimental at Timepoint 5' [1, 2 6] / E;
LSMESTIMATE TRT*TIMEPTN '2. mu2: Experimental at Baseline' [1, 2 1] / E;
LSMESTIMATE TRT*TIMEPTN '3. mu3: 1/2*(Experimental at Timepoint 1 + Experimental
at Timepoint 2)' [0.5, 2 2] [0.5, 2 3] / E;
LSMESTIMATE TRT*TIMEPTN '4. mu4: -Control at Timepoint 5' [-1, 1 6] / E;
LSMESTIMATE TRT*TIMEPTN '5. mu5: -Control at Baseline' [-1, 1 1] / E ;
LSMESTIMATE TRT*TIMEPTN '6. mu6: -1/2*(Control at Timepoint 1 + Control at
Timepoint 2)' [-0.5, 1 2] [-0.5, 1 3] / E;
LSMESTIMATE TRT*TIMEPTN 'mu_diff_PMM: (Experimental-Control) at Time-point 5'
[&pi1, 2 6] [&pi2, 2 1] [&pi3_half, 2 2] [&pi3_half, 2 3]
[-&pi4, 1 6] [-&pi5, 1 1] [-&pi6_half, 1 2] [-&pi6_half, 1 3] / E;

run;

```

LSM (contrast) coefficients c_m^p (see equation 4) for each LSM involved in the linear combination `mu_diff_PMM` can be obtained from the ODS output dataset `COEF`, the structure of which is shown in the right panel of Table 3. The `LMATRIX` variable in this dataset corresponds to the number of the `LSMESTIMATE` statement that was specified in the call to `PROC MIXED`. In Table 3, a fragment of the `COEF` dataset is shown corresponding to the LSM “mu1” (records with `LMATRIX=1`). `ROW1` variable represents the coefficients (c_m^{mu1}) for all model effects for this LSM estimate.

Table 3: Structure of the ODS output datasets from PROC MIXED

ODS Dataset SOLUTIONF				ODS Dataset COEF (fragment)				
Effect	TRT	TIMEPTN	Estimate	LMatrix	Effect	TRT	TIMEPTN	Row1
Intercept			87.6462	1	Intercept			1
TRT	0		-16.6337	1	TRT	0		0
TRT	1		0	1	TRT	1		1
TIMEPTN		0	-47.8973	1	TIMEPTN		0	0
TIMEPTN		1	-13.1025	1	TIMEPTN		1	0
TIMEPTN		2	-9.4208	1	TIMEPTN		2	0
TIMEPTN		3	-5.1077	1	TIMEPTN		3	0
TIMEPTN		4	-1.4617	1	TIMEPTN		4	0
TIMEPTN		5	0	1	TIMEPTN		5	1
TRT*TIMEPTN	0	0	16.8269	1	TRT*TIMEPTN	0	0	0
TRT*TIMEPTN	0	1	-6.2583	1	TRT*TIMEPTN	0	1	0
TRT*TIMEPTN	0	2	2.8260	1	TRT*TIMEPTN	0	2	0
TRT*TIMEPTN	0	3	1.4565	1	TRT*TIMEPTN	0	3	0
TRT*TIMEPTN	0	4	1.1285	1	TRT*TIMEPTN	0	4	0
TRT*TIMEPTN	0	5	0	1	TRT*TIMEPTN	0	5	0
TRT*TIMEPTN	1	0	0	1	TRT*TIMEPTN	1	0	0
TRT*TIMEPTN	1	1	0	1	TRT*TIMEPTN	1	1	0
TRT*TIMEPTN	1	2	0	1	TRT*TIMEPTN	1	2	0
TRT*TIMEPTN	1	3	0	1	TRT*TIMEPTN	1	3	0
TRT*TIMEPTN	1	4	0	1	TRT*TIMEPTN	1	4	0
TRT*TIMEPTN	1	5	0	1	TRT*TIMEPTN	1	5	1
scoreb			0.7976	1	scoreb			197.43
			

We can transpose dataset `COEF` (using `PROC TRANSPOSE` for variable `ROW1`, by `EFFECT`, `TRT`, `TIMEPTN`, and with `ID=LMatrix`), to get coefficients c_m^p for different LSMs ($p=\text{mu1}, \text{mu2}, \dots, \text{mu6}$) as columns (variables `COEFMU1` – `COEFMU6`):

			c_m^{mu1}	c_m^{mu2}	...	c_m^{mu5}	c_m^{mu6}
Effect	TRT	TIMEPTN	COEFMU1 (COEF.ROW1 where LMATRIX=1)	COEFMU2 (COEF.ROW1 where LMATRIX=2)	...	COEFMU5 (COEF.ROW1 where LMATRIX=5)	COEFMU6 (COEF.ROW1 where LMATRIX=6)
Intercept			1	1	...	-1	-1
TRT	0		0	0	...	-1	-1
TRT	1		1	1	...	0	0
TIMEPTN		0	0	1	...	-1	0
TIMEPTN		1	0	0	...	0	-0.5
TIMEPTN		2	0	0	...	0	-0.5
TIMEPTN		3	0	0	...	0	0
TIMEPTN		4	0	0	...	0	0
TIMEPTN		5	1	0	...	0	0
TRT*TIMEPTN	0	0	0	0	...	-1	0
TRT*TIMEPTN	0	1	0	0	...	0	-0.5
TRT*TIMEPTN	0	2	0	0	...	0	-0.5
TRT*TIMEPTN	0	3	0	0	...	0	0
TRT*TIMEPTN	0	4	0	0	...	0	0
TRT*TIMEPTN	0	5	0	0	...	0	0
TRT*TIMEPTN	1	0	0	1	...	0	0
TRT*TIMEPTN	1	1	0	0	...	0	0
TRT*TIMEPTN	1	2	0	0	...	0	0
TRT*TIMEPTN	1	3	0	0	...	0	0
TRT*TIMEPTN	1	4	0	0	...	0	0
TRT*TIMEPTN	1	5	1	0	...	0	0
scoreb			197.43	197.43	...	-197.4	-197.4

If we merge this dataset with the ESTIMATE variable from SOLUTIONF (merge by Effect, TRT and TIMEPTN) and add variables containing proportions of completers and drop-outs, we will have assembled all the data described in Table 1 needed to compute partial derivatives using equations (3.8) and (3.9). The full dataset, which we refer to as MU_ALL, would have the following structure:

Table 3: Structure of the Dataset MU_ALL Containing Information for Computing a Vector of Partial Derivatives

			β_m	c_m^{mu1}	...	c_m^{mu6}	$\pi_1=\&pi1$...	$\pi_6=\&pi6$
Effect	TRT	TIMEPTN	Estimate	COEFMU1	...	COEFMU6	Pi1	...	Pi6

Partial derivatives of mu_diff_PMM with respect to model parameters, as defined in equation (3.8), can now be computed in a data step as shown in SAS Code Fragment 2, processing data in each row in order to compute each derivative $\frac{\partial \text{mu_diff_PMM}}{\partial \beta_m}$.

SAS Code Fragment 2

```

DATA DER_PARAM(drop=i);
  set MU_ALL;

  array c(6) C1 - C6 ;
  array pi(6) Pi1 - Pi6 ;

  DER_BETA=0;
  do i=1 to 6; DER_BETA = DER_BETA +c[i]*pi[i]; end;
run;

```

Partial derivatives of mu_diff_PMM with respect to proportions of completers and drop-outs, as defined in equation (3.9), can be computed in a data step as shown in SAS Code Fragment 3, processing data in columns with the help of a RETAIN statement in order to compute each derivative $\frac{\partial \text{mu_diff_PMM}}{\partial \pi_p}, p = 1, \dots, 6$.

SAS Code Fragment 3

```

DATA DER_PI (keep=DER_Pi1 - DER_Pi6);
  set MU_ALL;

  retain DER_Pi1 - DER_Pi6 0;

  array c(6) C1 - C6 ;
  array pi(6) Pi1 - Pi6 ;
  array der_pi(6) DER_Pi1 - DER_Pi6 ;

  do i=1 to 6; der_pi[i] = der_pi[i]+c[i]*Estimate; end;

  *** At the last row, output derivatives with respect to proportions ***;
  *** &M is a macro variable containing number of parameters in the model ***;
  if paramn=&M then output;

run;

```

Partial derivatives with respect to model parameters computed in the dataset DER_PARAM, variable DER_BETA (see SAS Code Fragment 2) and partial derivatives with respect to proportions computed in the dataset DER_PI (see SAS Code Fragment 3) need to be placed together in a row vector as defined in equation (3.4). A dataset DER_ALL can be created containing variables COL1 through COL28 as follows:

Table 4: Structure of Dataset DER_ALL Containing Vector of Partial Derivatives

DER_PARAM .DER_BETA where PARAMN=1	...	DER_PARAM. DER_BETA where PARAMN=22	DER_PI. DER_PI1	...	DER_PI. DER_PI6
COL1	...	COL22	COL23	...	COL28

Variance-covariance matrix defined in (3.7) consists of two components: variance-covariance matrix of the model parameter estimates, and variance-covariance matrix for proportions of completers and drop-outs.

Variance-covariance matrix of the model parameter estimates is readily available in the ODS output dataset COVB from PROC MIXED. This dataset contains variables EFFECT, TRT, TIMEPTN (which identify model parameters similar to other ODS output datasets), a variable ROW with a sequential number for each parameter, and variables COL1 through COL<M>, where M is the number of parameters in the model. A value of COL<m> on a record where ROW=<l> represents variance (covariance) $VarCov(\beta_l, \beta_m)$. Thus, variables COL1 through COL<M> could be used directly as the upper left block of the matrix defined in (3.7).

The lower right block of the matrix in (3.7), variance-covariance matrix $VarCov(\boldsymbol{\pi})$ for proportions has two diagonal blocks itself: one a 3x3 matrix $VarCov(\boldsymbol{\pi}_{exp})$ corresponding to the multinomial proportions in the experimental arm $\boldsymbol{\pi}_{exp} = [pi1, pi2, pi3]$, and the other, $VarCov(\boldsymbol{\pi}_{con})$, in the control arm $\boldsymbol{\pi}_{con} = [pi4, pi5, pi6]$. We can use a macro %varmultinom provided in Appendix 1, which implements formulas in (3.5) and (3.6) to compute the elements of these matrices using multinomial proportions as input.

The entire variance-covariance matrix will have the following structure:

$$\begin{bmatrix} VarCov(\boldsymbol{\beta}) = \text{COVB (COL1 through COL<M>)} & 0 & 0 \\ 0 & VarCov(\boldsymbol{\pi}_{exp}) & 0 \\ 0 & 0 & VarCov(\boldsymbol{\pi}_{con}) \end{bmatrix}$$

Assuming that this matrix is stored in a SAS dataset, VARMAT, the last step in the calculations is defined by equation (3.3) and requires vector-matrix multiplication DER_ALL x VARMAT x DER_ALLT, where DER_ALLT is a transposed dataset obtained from DER_ALL (column vector). This operation can be performed using macro %vmvt provided in Appendix 1 which uses functionality of PROC FCMP.

Results provided in Table 5 represent the difference between experimental and control arms at time-point 5 as obtained from three analysis methods. MMRM-PMM represents the approach described above in this paper. MMRM-LOCF/BOCF represents the analysis where missing values were first imputed using single imputation BOCF for subjects discontinued due to LTB and single imputation LOCF for subjects that discontinued due to other reasons. The imputed dataset was then analyzed using the same mixed model with repeated measures as for the MMRM-

PMM (identical MODEL and REPEATED statements), but without the PMM component for the estimate of the treatment effect at time-point 5 (LSESTIMATE with coefficients [1, 2,6] [-1,1 6]). It should be noted that MMRM-PMM and MMRM-LOCF/BOCF approaches rely on similar clinical assumptions with respect to the treatment effect (benefit) attributed to drop-outs, but the computational/statistical methodology of implementing these assumptions are different. The MMRM-MAR analysis represents a MAR-based MMRM with the model identical to the one used in the MMRM-PMM and MMRM-LOCF/BOCF analyses, but with no prior imputations or PMM component. This method differs from the first two in the underlying clinical assumptions (under MAR, it is assumed that drop-outs will follow a trajectory similar to completers after discontinuation).

Table 5: Analysis results for the example dataset.

Method	Difference (Experimental - Control)		
	Least Squares Mean Estimate	Standard Error	P-value
MMRM-PMM	16.39	3.65	<0.0001
MMRM-LOCF/BOCF	17.05	4.01	<0.0001
MMRM-MAR	16.63	4.10	<0.0001

The results presented in Table 5 reveal two, perhaps surprising, findings. The MMRM-PMM approach provides a smaller estimate of the difference between treatments than both the MMRM-LOCF/BOCF and MMRM-MAR methods. Interestingly, the MMRM-PMM estimate is closer to the MMRM-MAR estimate than to that of the MMRM-LOCF/BOCF, despite the fact that the assumptions of MMRM-PMM and MMRM-LOCF/BOCF are clinically more similar than those of MMRM-PMM and MMRM-MAR.

A second surprising finding is that, although MMRM-PMM estimate of the treatment difference is smaller than that of MMRM-LOCF/BOCF, the MMRM-PMM method has a smaller standard error compared to the single imputation-based method MMRM-LOCF/BOCF. This standard error is also smaller than that of the MMRM-MAR.

These are phenomena that we observed with other PMMs (based on different clinical assumptions) implemented as linear combinations of LSMs. It is important to note that the MMRM-PMM analysis is rooted in an MMRM model that is estimated using MAR assumptions and unimputed data in the first place, and then the LSMs from this model are "reassigned" to subgroups of patients at various time-points using appropriate "weights". This is quite different from imputing data first, and then estimating a statistical model based on data containing imputations. The use of "weights" involved in the linear combination of LSMs results not only in weighting of the corresponding LSMs, but also of their individual standard errors, which results in a decreased overall standard error. A correction factor added by the delta method accounts for the fact the "weights" (proportions of completers and drop-outs) are estimates of random variables and does not offset the reduction in the overall standard error caused by "weights" in the linear combination of LSMs. In general, the standard error for treatment effect from the MMRM-based PMM approach is often smaller than that from standard MMRM using the MAR assumption. This goes against expectations: the PMM approach generally introduces assumptions for the missing data that diverge from what is seen in the observed data, so one would expect the standard error of the estimate of treatment effect under the PMM to be larger, rather than smaller. The standard error is often smaller under MMRM-based PMM because the standard error of MMRM-based PMM does not reflect the variability of the data postulated by the MNAR scenario; rather it reflects the variances and correlations of the data at time points as estimated from the standard MAR model. A more in-depth discussion of these aspects is beyond the scope of this paper and the authors hope to elaborate them in another publication in a near future.

6. CONCLUSION

In this paper, we demonstrated how certain types of MNAR assumptions regarding missingness in clinical trials can be implemented within a PMM-type framework using linear combinations of LSMs obtained from a standard MMRM. This analysis, including estimation of the standard error based on the delta approximation method, can be performed based on the information that is readily available from the PROC MIXED output with some additional relatively simple computations using core SAS/STAT functionality. These computations can be effectively implemented using data structures presented in this paper. The proposed implementation strategy is a general one that can be used to impose different sets of MNAR assumptions on a standard MMRM and is not confined to a specific example used for illustration in this paper.

Maximum likelihood-based and MI-based implementations of PMM each have their advantages and disadvantages that a statistician may need to weigh in the context of a given study. Maximum likelihood-based strategies are attractive because of a "single result" that they provide at the end of the analysis and the fact that missing data are not explicitly imputed ("invented"). This is in contrast to the MI-based approaches that perform explicit imputations and rely on random sampling and thus can produce slightly variable results depending on the random number generator sequence used. Such variations, however, are quite minor especially if a sufficiently large number of imputations are produced. Currently there is also no consensus on the appropriate way to treat non-monotone

missing data (subjects with intermediate missing visits) with MI in the context of MNAR assumptions. MI-based implementations, on the other hand, have an advantage that factors that are relevant to missingness but not relevant to analysis can be included in the imputation model in order to better estimate the distribution of unobserved outcomes or express MNAR assumptions that rely on such factors, and then subsequently be dropped from the analysis model. This cannot be achieved with the maximum likelihood-based PMM implementation. As pointed out in Section 5, an MMRM-based and imputation-based implementation of similar clinical assumptions about missingness can result in somewhat different estimates of the treatment difference. Also, while the MMRM-based PMM approach would result in a smaller estimate of the treatment difference compared to a standard MAR-based MMRM (provided that the underlying MNAR assumptions are more conservative than MAR), the standard error from the MMRM-based PMM approach can often be smaller than that of a MAR-based MMRM, as discussed above, and possibly smaller than that of a single imputation-based approach with clinically similar assumptions. Thus, results from the MMRM-based PMM analysis would impact the clinical interpretation of the magnitude of the treatment effect (with MMRM-based PMM estimate being more conservative), but likely not the hypothesis test for statistical significance based on the Wald statistic.

REFERENCES

- [1] European Medicines Agency. "Guideline on Missing Data in Confirmatory Clinical Trials. 2 July 2010. EMA/CPMP/EWP/1776/99 Rev. 1".
http://www.ema.europa.eu/docs/en_GB/document_library/Scientific_guideline/2010/09/WC500096793.pdf
- [2] National Research Council. 2010. The Prevention and Treatment of Missing Data in Clinical Trials. Panel on Handling Missing Data in Clinical Trials. Committee on National Statistics, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academic Press.
- [3] Little R.J.A. 1993. Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association* 88: 125-134.
- [4] Ratitch B., O'Kelly M. "Implementation of Pattern-Mixture Models Using Standard SAS/STAT Procedures." PharmaSUG 2011. <http://www.pharmasug.org/proceedings/2011/SP/PharmaSUG-2011-SP04.pdf>
- [5] Hedeker D., Gibbons, R.D. 1997. Application of Random-Effects Pattern-Mixture Models for Missing Data in Longitudinal Studies. *Psychological Methods* 2(1):64-78.
- [6] Oehlert, G. W. 1992. A note on the delta method. *American Statistician* 46: 27–29.
- [7] Hedeker, D. "Missing Data in Longitudinal Studies." Presentation slides.
<http://www.uic.edu/classes/bstt/bstt513/missbLS.pdf>
- [8] SAS Institute Inc. 2008. SAS/STAT® 9.2 User's Guide. Cary, NC: SAS Institute Inc.
- [9] Roger, J. Discussion of Incomplete and Enriched Data Analysis and Sensitivity Analysis presented by Geert Molenberghs. Drug Information Association (DIA) Meeting, Special Interest Area Communities (SIAC) - Statistics, January 2010.
- [10] Kiernan, K., Tobias, R., Gibbs, P., Tao, J. CONTRAST and ESTIMATE Statements Made Easy: The LSMESTIMATE Statement. SAS Global Forum 2011, paper 351-2011.
<http://support.sas.com/resources/papers/proceedings11/351-2011.pdf>
- [11] Molenberghs G., Kenward M.G. 2007. *Missing Data in Clinical Studies*. Wiley.
- [12] Thijs H., Molenberghs G. 2002. Strategies to Fit Pattern-Mixture Models. *Biostatistics* 3(2): 245-265.
- [13] Roger, J., Kenward, M. Repeated measures approach to "What if" questions for longitudinal studies with withdrawal. PSI Conference, Brighton. 18th May 2009.

APPENDIX 1

This appendix contains several utility macros referred to in Section 5. The full code of the implementation described in Section 5 can be made available upon reader's request.

Macro %nobscount

```
%macro nobscount(datain=, wherein=, mvname=);

    *** Macro "nobscount" counts the number of observations in the input dataset.
        Output is saved in a macro variable.
        Macro arguments:
            "datain"    : input dataset to count observation in
            "wherein"   : conditions to select a subset of observation in the input
dataset
            "mvname"    : macro variable name to store the computed number of
observations

        Sample call - counting completers in control arm:
        %nobscount(datain=DATAIN1, wherein=%str(TRT=0 and TIMEPTN=1 and DISCRSNN=0),
mvname=n0_com);
    ***;

    %global &mvname;
    %local nobsloc;

    data temp1; set &datain; where &wherein; run;

    %let dsid=%sysfunc(open(temp1,in));
    %if &dsid > 0 %then %do;
        %let nobsloc=%sysfunc(attrn(&dsid,nobs));
        %let &mvname = &nobsloc;
        %let rc=%sysfunc(close(&dsid));
    %end;

    %else %put open for data set &datain failed - %sysfunc(sysmsg());
%mend nobscount;
```

Macro % varmultinom

```
%macro varmultinom(datain=, dataout=, pi_list=, var_list=, n=, tot=);

    *** Macro "varmultinom" computes variance-covariance matrix for a multinomial
distribution.
        Input consists of a dataset containing one record and <n> variables
containing <n> multinomial
        proportions pi_i.
        Output is saved in a a dataset that has <n> records and <n> variables, where
each value
        corresponds to the variance/covariance of pi_i and pi_j.
        Macro arguments:
            "datain"    : input dataset containing one record and <n> variables
containing <n>
                        multinomial proportions
            "dataout"   : output dataset name to store the computed variance/covariance
matrix
            "pi_list"   : list of variable names, separated by space, containing
multinomial
                        proportions in the input dataset
            "var_list"  : list of variable names, separated by space, to be used in
the output dataset
                        "dataout"
            "n"         : number of proportions in the input dataset
            "tot"       : sample size (N) based on which multinomial proportions were
estimated

        Sample call - computing variance-covariance matrix for proportions in
experimental arm:
        %varmultinom(datain=mult1, dataout=mult11, pi_list=%str(PI1-PI3),
var_list=%str(COL23-COL25), n=3, tot=&tot1);
```

&tot1 is a macro variable containing number of subjects in experimental arm.

```
***;

data &dataout(keep=&var_list);
  set &datain;

  array pi(&n) &pi_list;
  array varl(&n) &var_list;

  do i=1 to &n;
    do j=1 to &n;
      if i=j then varl[j]=pi[j]*(1-pi[j])/&tot;
      else varl[j]=-pi[i]*pi[j]/&tot;
    end;
    output;
  end;

  drop i j;
run;
%mend varmultinom;
```

Macro %obsnvars

```
%macro obsnvars(ds=);
  *** Macro "obsnvars" counts the number of observations and variables in the input
  dataset.
  Output is saved in global macro variables "nvars" and "nobs".
  Macro arguments:
    "ds" : input dataset to count observation and variables in
  ***;
  %global nvars nobs;

  /* Open input data set */
  %let dsid = %sysfunc(open(&ds));

  /* If the data set exists, then get the number of observations and variables */
  %if &dsid %then
    %do;
      %let nobs = %sysfunc(attrn(&dsid,nobs));
      %let nvars=%sysfunc(attrn(&dsid,nvars));
      %let rc = %sysfunc(close(&dsid));
    %end;

  /* Otherwise, write a message that the data set could not be opened */
  %else %put open for data set &dset failed - %sysfunc(sysmsg());
%mend obsnvars;
```

Macro %vmvt

```
%macro vmvt(vec=, mat=, prod=);

  *** Macro "vmvt" performs multiplication of VECTOR * MATRIX * VECTOR' where
  MATRIX is a square
  matrix of the same dimentions as the length of row VECTOR and VECTOR' is a
  transposed column
  vector. VECTOR and MATRIX are assumed to be stored in SAS datasets, the names
  of which are
  passed as arguments to the macro. The product is a scalar stored in a macro
  variable.
  PROC FCMP gives access to a number of built-in CALLs and functions for
  manipulating matrices.
  The code below uses FCMP's READ_ARRAY function and its MULT call.
```

```

Macro arguments:
  "vec"      : Name of the dataset containing the VECTOR
  "matrix"   : Name of the dataset containing the MATRIX
  "prod"     : Name of a global macro variable that will contain
               the result of multiplication
***;

%local ld1 ld2 rd1 rd2;

%obsnvars(ds=&vec);
%let ld1=&nobs;
%let ld2=&nvars;

%obsnvars(ds=&mat);
%let rd1=&nobs;
%let rd2=&nvars;

%if (&rd1 ne &rd2) or (&ld1 ne 1) or (&ld2 ne &rd1) %then
  %put Vector or matrix dimentions are not suitable - operation not performed;

%if (&rd1 = &rd2) and (&ld1 = 1) and (&ld2 = &rd1) %then %do;
  proc fcmp;
    array left[&ld1,&ld2] / nosymbols;
    rc = read_array("&vec", left);

    array right[&rd1,&rd2] / nosymbols;
    rc = read_array("&mat", right);

    array result[&ld1,&rd2] / nosymbols;
    call mult(left, right, result);

    array left_t[&ld2,1] / nosymbols;
    call transpose(left, left_t);

    array prod[1,1] / nosymbols;
    call mult(result, left_t, prod);

    %global &prod;
    call symput("&prod", trim(left(put(prod[1,1],best.)))) );
  run;
  quit;
%end;
%mend vmvt;

```

ACKNOWLEDGMENTS

We would like to thank Quintiles Inc. for encouraging and supporting this work as well as conference participation. We would like to sincerely thank James Roger, Gary Koch, and Geert Molenberghs for numerous helpful discussions.

CONTACT INFORMATION

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

Bohdana Ratitch
 Quintiles Inc.
 100 Alexis-Nihon Blvd., Suite 800
 Saint-Laurent, Québec, Canada, H4M 2P4
 E-mail: Bohdana.Ratitch@quintiles.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.