Journal of Statistical Software

December 2015, Volume 68, Issue 8. doi: 10.18637/jss.v068.i08

A SAS Program Combining R Functionalities to Implement Pattern-Mixture Models

Pierre Bunouf
Laboratoires Pierre Fabre

Geert Molenberghs
I-BioStat, Universiteit Hasselt &
Katholieke Universiteit Leuven

Jean-Marie Grouin
Université de Rouen

Herbert Thijs
I-BioStat, Universiteit Hasselt &
Katholieke Universiteit Leuven

Abstract

Pattern-mixture models have gained considerable interest in recent years. Pattern-mixture modeling allows the analysis of incomplete longitudinal outcomes under a variety of missingness mechanisms. In this manuscript, we describe a SAS program which combines R functionalities to fit pattern-mixture models, considering the cases that missingness mechanisms are at random and not at random. Patterns are defined based on missingness at every time point and parameter estimation is based on a full group-by-time interaction. The program implements a multiple imputation method under so-called identifying restrictions. The code is illustrated using data from a placebo-controlled clinical trial. This manuscript and the program are directed to SAS users with minimal knowledge of the R language.

Keywords: MAR, MNAR, pattern-mixture model, identifying restriction, multiple imputation.

1. Introduction

Many experiments are concerned with analysis of longitudinal studies with outcomes suffering from missing values. In clinical trials, a main cause of missingness is subject dropout. Outcome values are missing at random (MAR) if the dropout mechanism is independent of missing outcome values, conditionally on the observed ones. If the covariates are fully observed, additional dependence on covariates is allowed for too. When MAR fails to hold, missing outcome values are said to be missing not at random (MNAR). MNAR means that

the probability of dropout depends on an outside variable not in the model or is related to unobserved outcome values at the time of dropout and possibly afterwards. The consequence of MNAR is that missing outcome values cannot be reliably imputed using observed measurements (i.e., covariate and outcome values). More precisely, explicit modeling of the dropout mechanism is needed. Pattern-mixture modeling (PMM) is a framework that can be considered when the dropout mechanism is MNAR; see, e.g., Verbeke and Molenberghs (2000). PMM stratifies the sample of subjects based on outcome patterns and formulates distinct models to estimate parameters within each stratum. The program described in this manuscript implements PMM analysis using a multiple imputation (MI) method under so-called identifying restrictions. Patterns are defined based on dropout at every time point. The estimation models incorporate a full group-by-time interaction for fixed effects and an unstructured error covariance matrix.

As the parameters of the estimation models identify each time point, some of these parameters are unidentified in incomplete patterns. These consequences can be overcome by using information available in other patterns. The identifying restrictions simply indicate from which patterns missing information is borrowed. The program described in this manuscript implements PMM analysis under several identifying restrictions. The complete-case missing values (CCMV) approach, introduced by Little (1993), stipulates that missing information is borrowed from completers (i.e., from subjects with complete outcome profile). In the neighboring-case missing values (NCMV) approach, the closest neighboring pattern is used. In other words, missing information at a time point is borrowed from the nearby pattern for which outcome values are observed at this time point, but unobserved later. The available-case missing values (ACMV) approach offers a compromise between CCMV and NCMV as missing information is borrowed from all available patterns weighted by occurrence of each pattern. ACMV has a particular status since this is the natural counterpart of MAR in the PMM framework. In practice, analysis under MAR can be a starting point for sensitivity analyses under MNAR. Another type of identifying restrictions, termed non-future missing values (NFMV), offers a relevant alternative because the user has full freedom to choose the distributions of the first unobserved values (which are termed the present values), given the previous measurements. Under NFMV, the missingness mechanism depends on the past and the present, but not on future unobserved outcome values.

The American National Academy of Sciences underscored the need to develop programs of analysis assuming MNAR; see National Research Council (2010, p. 114). New features in version 9.4 of SAS `PROC MI` (SAS Inc. 2014) allow one to generate imputations in the PMM framework under CCMV and NCMV. In earlier work, I-BioStat (2007) developed a SAS program which generates multiple imputations based on a mixed model for repeated measures (MMRM). However, some computations use approximations to ease matrix operations. The program described in this manuscript takes up the main aspects of algorithm but performs exact calculations. To do so, we use new features in SAS `PROC IML` (SAS Inc. 2011) that enable to call R functions from within the IML procedure. This automatic link is available from version 9.3 of SAS under Windows and Linux. Nevertheless, if another operating system (OS) is used and/or if SAS `PROC IML` version 9.3 or later is not available, thorough instructions are provided to execute the program anyway. The case study, which is used to illustrate the code, is a placebo-controlled clinical trial to assess the effect of a test drug in the treatment of macular degeneration.

The program is directed to SAS users with minimal knowledge of the R language (R Core

Team 2015b). Different aspects of PMM analysis are summarized in the next section. The algorithm and environment requirements are exposed in Section 3. Information to initiate program execution is provided in Section 4, whereas Section 5 describes the different stages of a case study analysis. Section 6 shows results of analyses under all the identifying restrictions available in the program. Additional information to specify other model parameterizations is given in Section 7.

2. PMM analysis

2.1. Identifying restrictions

In our context, patterns are defined based on dropout at every time point, except baseline. More precisely, if the tth outcome value is the last observed one and subject drops out after this point, this subject belongs to pattern t, $t = 1, \ldots, T$. The rationale for PMM stems from a particular decomposition of the joint distribution of the outcome variable together with the dropout indicator.

The pattern-mixture distribution of complete outcome values $y_1, \ldots, y_T$ is given by:

$$f(y_1, \ldots, y_T) = \sum_{t=1}^{T} \alpha_t f_t(y_1, \ldots, y_T), \tag{1}$$

where α_t denotes the proportion of pattern t and $f_t(y_1, \ldots, y_T)$ stands for $f(y_1, \ldots, y_T|t)$. In (1), the distribution of the whole population is expressed in terms of a mixture of the distributions of pattern populations. These in turn can be decomposed as:

$$\begin{aligned} f_t(y_1, \ldots, y_T) &= f_t(y_1, \ldots, y_t) f_t(y_{t+1}, \ldots, y_T|y_1, \ldots, y_t) \\ &= f_t(y_1, \ldots, y_t) \prod_{s=t+1}^{T} f_t(y_s|y_1, \ldots, y_{s-1}). \end{aligned} \tag{2}$$

The first component in (2) is identified from the observed outcome values. The second component is a product of conditional pattern distributions that are unidentified because the values of y_s are unobserved. Identification of unidentified parameters is a crucial aspect as the conditional pattern distributions form the basic framework for the imputation process. The identifying restrictions enable to overcome this problem: Unidentified parameters of incomplete patterns are set equal to appropriate functions of the parameters describing the distributions of other patterns.

We now describe the identifying restrictions available in the program. Further descriptions can be found in Thijs, Molenberghs, Michiels, Verbeke, and Curran (2002). Under CCMV, identification is based on the pattern of completers, which is pattern T. This identification can be formalized as:

$$f_t(y_s|y_1, \ldots, y_{s-1}) = f_T(y_s|y_1, \ldots, y_{s-1}), \quad s = t+1, \ldots, T. \tag{3}$$

Under NCMV, the neighboring pattern is used. We then have:

$$f_t(y_s|y_1, \ldots, y_{s-1}) = f_s(y_s|y_1, \ldots, y_{s-1}), \quad s = t+1, \ldots, T. \tag{4}$$

Identification can also be based on all identified patterns as specified in the formulation:

$$f_t(y_s|y_1,\ldots,y_{s-1}) = \sum_{j=s}^{T} \omega_{sj} f_j(y_s|y_1,\ldots,y_{s-1}), \quad s = t+1,\ldots,T. \tag{5}$$

We will use ω_s as shorthand for the set of positive ω_{sj}'s. Every ω_s that sums to 1 provides a valid identification scheme. Note that (5) results in the special case (3) by setting $\omega_{sT}=1$ and all other $\omega_{sj}=0$ and to the special case (4) by setting $\omega_{ss}=1$ and all other $\omega_{sj}=0$.

Molenberghs, Michiels, Kenward, and Diggle (1998) determined ω_s such that (5) corresponds to ACMV. The coefficients are defined as:

$$\omega_{sj} = \frac{\alpha_j f_j(y_1,\ldots,y_{s-1})}{\sum_{l=s}^{T} \alpha_l f_l(y_1,\ldots,y_{s-1})}, \quad j = s,\ldots,T. \tag{6}$$

Alternatively, non-future dependence is an assumption under which missingness is allowed to depend on the past and the present, but given these not on the future. Kenward and Molenberghs (2003) determined the identifying restriction, named NFMV, which corresponds to non-future dependence. Under NFMV, the conditional pattern distributions of present outcome values are left unconstrained. For the sake of clarity, these are noted g_t in what follows.

Our program implements two types of NFMV. Under NFMV_{CC}, the g_t functions are set equal to their f_T counterparts in the spirit of CCMV. In addition, the user has the possibility to introduce a location parameter Δ. An explicit formulation of this is given by:

$$g_t(y_{t+1}|y_1,\ldots,y_t) = f_T(y_{t+1}+\Delta|y_1,\ldots,y_t).$$

Under NFMV_{NC}, the g_t functions are set equal to their f_{t+1} counterparts in the spirit of NCMV. We then have:

$$g_t(y_{t+1}|y_1,\ldots,y_t) = f_{t+1}(y_{t+1}+\Delta|y_1,\ldots,y_t).$$

Once the g_t functions are determined, the conditional pattern distributions of y_s for $s = t+2,\ldots,T$ are given by:

$$f_t(y_s|y_1,\ldots,y_{s-1}) = \omega_{s,s-1} g_{s-1}(y_s|y_1,\ldots,y_{s-1}) + \sum_{j=s}^{T} \omega_{sj} f_j(y_s|y_1,\ldots,y_{s-1}) \tag{7}$$

with

$$\begin{aligned} \omega_{s,s-1} &= \frac{\alpha_{s-1} g_{s-1}(y_1,\ldots,y_{s-1})}{\alpha_{s-1} g_{s-1}(y_1,\ldots,y_{s-1}) + \sum_{l=s}^{T} \alpha_l f_l(y_1,\ldots,y_{s-1})}, \\ \omega_{sj} &= \frac{\alpha_j f_j(y_1,\ldots,y_{s-1})}{\alpha_{s-1} g_{s-1}(y_1,\ldots,y_{s-1}) + \sum_{l=s}^{T} \alpha_l f_l(y_1,\ldots,y_{s-1})}, \quad j \geq s. \end{aligned} \tag{8}$$

2.2. Multiple imputation method

The program implements a PMM analysis for Gaussian outcome variables, following the standard three-stage MI way, as described in Rubin (1987).

Pattern parameter estimation

Distinct models are formulated to fit the outcome variable within each pattern. The program uses MMRMs with a full group-by-time interaction at every time point for the fixed effects and an unstructured error covariance matrix. Time is treated as a class variable and no random effects are specified.

Let us denote by $\mathbf{Y}_i = (y_{i,1}, \ldots, y_{i,T})$ the complete outcome vector for the ith subject of pattern t and $\mathbf{Y}_{i,obs} = (y_{i,1}, \ldots, y_{i,t})$ its observed part. The MMRMs per pattern can be expressed as:

$$\mathbf{Y}_{i,obs} = X_i \boldsymbol{\beta}_t + \boldsymbol{\epsilon}_i, \tag{9}$$

where $\boldsymbol{\epsilon}_i \sim \mathrm{N}(\mathbf{0}, \Sigma_t)$, Σ_t is unstructured, and the $\boldsymbol{\epsilon}_i$'s are independent. The matrix X_i contains the known fixed-effects covariates whereas $\boldsymbol{\beta}_t$ contains the unknown parameters.

This first stage provides the estimates $\widehat{\boldsymbol{\beta}}_t$, $\widehat{\mathsf{VAR}}(\widehat{\boldsymbol{\beta}}_t)$, $\widehat{\Sigma}_t$, and $\widehat{\mathsf{VAR}}(\mathrm{col}(\widehat{\Sigma}_t))$, where $\mathrm{col}(\widehat{\Sigma}_t)$ is the vector containing the coefficients of the diagonal and the lower part of $\widehat{\Sigma}_t$.

Imputation

Imputation of missing outcome values is conducted sequentially by value. We describe here-below how to obtain a run of M imputed values of $y_{i,t+1}$. Multiple imputation of $y_{i,t+2}, \ldots, y_{i,T}$ follows the same process by considering the previous imputed values as observed ones.

The imputed values of $y_{i,t+1}$ are drawn from conditional pattern distributions. The selection of the pattern of imputation is driven by the identifying restriction chosen. In our illustration, we first suppose that pattern r $(t+1 \leq r \leq T)$ is used. Let us introduce $\boldsymbol{\mu}_{i,r}$ for the mean of $\mathbf{Y}_i$, which is $\boldsymbol{\mu}_{i,r} = X_i \boldsymbol{\beta}_r$. Based on appropriate parts of $\boldsymbol{\mu}_{i,r}$ and Σ_r, we further define the distributions of $\mathbf{Y}_i$ components as $\mathbf{Y}_{i,obs} \sim \mathrm{N}(\boldsymbol{\mu}_{i,r,1}, \Sigma_{r,11})$ and $y_{i,t+1} \sim \mathrm{N}(\mu_{i,r,2}, \Sigma_{r,22})$. Their covariances are denoted $\Sigma_{r,12}$ and $\Sigma_{r,21}$. Using $2|1$ as notation for $y_{i,t+1}|y_{i,1}, \ldots, y_{i,t}$, the conditional pattern distribution of $y_{i,t+1}$ given $y_{i,1}, \ldots, y_{i,t}$ is described by:

$$f_r(y_{i,t+1}|y_{i,1}, \ldots, y_{i,t}) \sim N(\mu_{i,r,2|1}, \Sigma_{r,2|1}),$$

where

$$\begin{aligned} \mu_{i,r,2|1} &= \mu_{i,r,2} + \Sigma_{r,21}[\Sigma_{r,11}]^{-1}(\mathbf{Y}_{i,obs} - \boldsymbol{\mu}_{i,r,1}), \\ \Sigma_{r,2|1} &= \Sigma_{r,22} - \Sigma_{r,21}[\Sigma_{r,11}]^{-1}\Sigma_{r,12}. \end{aligned} \tag{10}$$

Uncertainty pertaining to the pattern parameters $\boldsymbol{\beta}_r$ and Σ_r is incorporated through their Bayesian posterior predictive distributions. On the basis of Gaussian distributions and non-informative Jeffreys' priors, the values of $\widehat{\boldsymbol{\beta}}_r^{(m)}$ and $\widehat{\Sigma}_r^{(m)}$, $m = 1, \ldots, M$, are respectively randomly drawn from the posterior distributions $\mathrm{N}(\widehat{\boldsymbol{\beta}}_r, \widehat{\mathsf{VAR}}(\widehat{\boldsymbol{\beta}}_r))$ and $\mathrm{N}(\mathrm{col}(\widehat{\Sigma}_r), \widehat{\mathsf{VAR}}(\mathrm{col}(\widehat{\Sigma}_r)))$. After the derivation of $\widehat{\boldsymbol{\mu}}_{i,r}^{(m)}$, the imputed values of $y_{i,t+1}$ are drawn from the conditional pattern distributions which are expressed by:

$$f_r^{(m)}(y_{i,t+1}|y_{i,1}, \ldots, y_{i,t}) \sim N(\widehat{\mu}_{i,r,2|1}^{(m)}, \widehat{\Sigma}_{r,2|1}^{(m)}), \qquad m = 1, \ldots, M. \tag{11}$$

Under ACMV and NFMV, (5) and (7) indicate that imputation of a missing $y_{i,t+1}$ value is based on a sum of conditional distributions of several patterns weighted by occurrence of each pattern. In the MI setting, this weighted summation is handled via random pattern selection over imputations. For each imputation, we calculate the coefficients ω_{sj} in (6) and (8), which characterize pattern probabilities. Random pattern selection by imputation is based on coefficient values, noted $\omega_{sj}^{(m)}$, and another value $U^{(m)}$ which is drawn from the uniform distribution. So, pattern p is selected if:

$$\Sigma_{j=s}^{p-1}\omega_{sj}^{(m)} \leq U^{(m)} < \Sigma_{j=s}^{p}\omega_{sj}^{(m)}. \tag{12}$$

Pooled analysis

The complete data sets are fitted using the modeling strategy described in (9) for pattern parameter estimation. The MMRM approach incorporates a full group-by-time interaction at every time point for the fixed effects and an unstructured error covariance matrix. The analysis model can be expressed as:

$$\mathbf{Y}_i = X_i\boldsymbol{\beta}^* + \boldsymbol{\epsilon}_i^*, \tag{13}$$

where $\boldsymbol{\epsilon}_i^* \sim \mathrm{N}(\mathbf{0}, \Sigma)$, Σ is unstructured, and the $\boldsymbol{\epsilon}_i^*$'s are independent. The vector $\boldsymbol{\beta}^*$ contains the unknown parameters of fixed effects.

The principle of MI is to combine the inferences made on the M imputations into a single one. Let us define $\widehat{\boldsymbol{\beta}}^{*(m)}$, $m = 1, \ldots, M$, the estimators of $\boldsymbol{\beta}^*$ by imputation and $\widehat{\boldsymbol{\beta}}^*$ the pooled estimator of $\boldsymbol{\beta}$. In the usual Rubin's formulation

$$\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^* \sim N(\mathbf{0}, V),$$

$\widehat{\boldsymbol{\beta}}^*$ is the average of $\widehat{\boldsymbol{\beta}}^{*(m)}$'s whereas the pooled variance V is given by

$$V = W + \left(\frac{M+1}{M}\right) B,$$

where W is the average of $\widehat{\mathrm{VAR}}(\widehat{\boldsymbol{\beta}}^{*(m)})$ and B is the between-imputation variance:

$$B = \sum_{m=1}^{M} \frac{(\widehat{\boldsymbol{\beta}}^{*(m)} - \widehat{\boldsymbol{\beta}}^*)(\widehat{\boldsymbol{\beta}}^{*(m)} - \widehat{\boldsymbol{\beta}}^*)^\top}{M-1}.$$

The pooled estimator $\widehat{\boldsymbol{\beta}}^*$ is consistent under MNAR. The F-distributions of the tests of fixed effects are given in Li, Raghunathan, and Rubin (1991).

3. Algorithm and environment

3.1. Algorithm

The programming strategy was driven by the respective strengths of SAS and R. The use of SAS to handle and analyze data is widespread in some industries, such as the pharmaceutical

industry. Consequently, many statisticians are familiar with pre-programmed SAS procedures such as SAS `PROC MIXED` to fit continuous longitudinal outcomes and SAS `PROC MIANALYZE` to combine inferences in the MI setting. On the other hand, R provides an optimized environment for operations on matrices and, in our context, is more indicated for the imputation stage which requires many calculations of this type.

The algorithm is described for each of the three tasks of standard MI. For each task, we specify which functionalities of SAS and R are used. The algorithm is described as follows:

1. *Pattern parameter estimation*:
   - ▷ Use SAS `PROC MIXED` to fit outcome per pattern and estimate parameters.
2. *Imputation*: Use R to
   - ▷ Generate pattern parameter values by imputation.
   - ▷ Draw missing outcome values from conditional pattern distributions of all available patterns.
   - ▷ Impute using the patterns specified by the identifying restriction chosen.
3. *Pooled analysis*:
   - ▷ Use SAS `PROC MIXED` to analyze complete data sets.
   - ▷ Use SAS `PROC MIANALYZE` to combine inferences.

Two important remarks concern the imputation stage:

- Intermittent missing values (i.e., values missing not due to dropout) are multiply imputed by drawing values from the conditional distribution of the subject pattern, given all the other available outcome values. So, if a subject belongs to pattern t, imputation is based on parameters of this pattern.
- Some cases may require a large number of imputations (information about the number of imputations needed is provided in Section 6). This number strongly impacts program execution time because of data handling with R, which slows down the program process as the number of imputations increases. To gain time, multi-step functionality that enables execution of successive batches of imputations has been introduced. The gain becomes rapidly substantial: In the case study, a total of 500 imputations is set by default; the program execution with five batches of 100 imputations is twice faster than with a unique batch of 500 imputations.

The call of R from SAS `PROC IML` is available from version 9.3 of SAS under Linux and Windows OS. If the user's environment does not enable an automatic link between SAS and R, the structure of the program nevertheless allows for execution. However, the multi-step functionality for imputation is no longer available. To do so, the three stages of the MI procedure must be executed separately as follows:

1. *Pattern parameter estimation*:
   - ▷ Select lines 1–223 in the SAS code and execute.

2. *Imputation*:

   ▷ Select lines 2–374 in the R script and execute.
   This selection excludes the statements related to the link with SAS `PROC IML` which are `submit / R;` and `endsubmit;`.

3. *Pooled analysis*:

   ▷ Select lines 251–279 in the SAS code and execute to analyze complete data sets.

   ▷ Remove the character string '`+ &nImputations * (&batch - 1)`' in lines 280 and 281, and execute them. Also select and execute lines 284–285.

   ▷ Select lines 307–310 and execute to combine inferences.

The combination of the SAS and R functionalities and the call of R from SAS was a deliberate choice, as the program is primarily targeted at SAS users. However, the clear separation between the three stages, i.e., (1) pattern parameter estimation, (2) imputation, and (3) pooled analysis, of the program allows alternative implementation strategies. For example, it is quite feasible to invert the initial user's environment and call SAS from R.

3.2. Environment requirements

Some environment parameters must be set-up prior to the first program execution. To call R, the SAS system must be launched with the `-RLANG` command. This command can definitively be inserted in the file `SASV9.cfg`, which is stored in the folder `\nls\en` of the user's SAS environment. The `-RLANG` command allows the execution of the `RLANG` option in the SAS program using the syntax: `proc options option=RLANG;`. Under Windows, a reason of failure may be that the path to the R directory (and not to the file `R.exe`) is not registered as a Windows environment variable. In this case, the user must enter this information manually (e.g., `Variable=R_HOME, Value=C:\Program\Files\R R-2.15.1`).

The functionalities of SAS `PROC IML` which enable to call R are available from SAS 9.3. The interface with R until version 2.15.3 is supported in SAS 9.3. However, the interface with R 3.0.x is not supported in SAS 9.3, but is supported in SAS 9.4.

The part of the program written in SAS is contained in the file `PMM.sas` whereas `PMM.R` contains the R script. SAS `PROC IML` calls the R script via the command:

```
\%include "\&Path2WorkDir/PMM.R";
```

The SAS data files produced throughout program execution are stored in the working directory. Import/export of data files between SAS and R is done using R functions. Note that such functionalities also exist in SAS `PROC IML`, but only concern numerical data in matrix format.

Program execution requires the three R packages **Hmisc** (Harrel 2015), **foreign** (R Core Team 2015a), and **MASS** (Venables and Ripley 2002; Ripley 2015) to be installed. If these packages are not available in the user's R environment, these can be downloaded from the Comprehensive R Archive Network at `http://CRAN.R-project.org/` by removing the hash sign `#` in the following three lines at the top of the R script:

```
# install.packages("Hmisc")
# install.packages("foreign")
# install.packages("MASS")
```

We strongly recommend the user to install the packages in the R environment, i.e., without using SAS to launch R, by executing the three commands here-above in the R editor. It is also important to put back the hash signs once the packages are installed. More generally, any substantial modification in the R script should be done in the R environment, using either the R editor or any R interface such as **RStudio** or **Tinn-R** (Faria, Grosjean, Jelihovschi, Pietrobon, and Silva Farias 2015).

4. Initiation of program execution

To initiate program execution, the user must put the three files `PMM.sas`, `PMM.R`, and `Data.sas7bdat` in a working directory, which is for example `C:/WorkDir`. The SAS file `PMM.sas` and the R file `PMM.R` contain the program code, whereas the SAS data file `Data` contains the data of the case study. In this section, we describe how to initiate execution of the case study. We also provide information to use the program in other circumstances.

4.1. Description of the case study

The case study arises from a randomized clinical trial comparing a test drug with a corresponding placebo in the treatment of subjects with age-related macular degeneration. Subjects with macular degeneration progressively lose vision. In the trial, the subjects' visual acuity was assessed through subjects' ability to read lines of letters on standardized vision charts. These charts display lines of 5 letters of decreasing size, which the patient must read from top (largest letters) to bottom (smallest letters). The subjects' visual acuity is the total number of letters correctly read.

Subjects were asked to undergo a pre-randomization visit (baseline) and four post-randomization visits, i.e., visit 1 at 4 weeks, visit 2 at 12 weeks, visit 3 at 24 weeks, and visit 4 at 52 weeks. Treatment-effect inferences on visual acuity at visit 4 is the primary focus of the statistical analysis.

An overview of the different missingness patterns is given in Table 1. Note that 188 of the 226 profiles are complete, which is a percentage of 83.2%, while 16.8% (38 subjects) exhibit monotone missingness.

Pattern	Visit 1	Visit 2	Visit 3	Visit 4	Pattern distribution	
					Freq	Percent
1	O	O	O	O	188	83.2
2	O	O	O	M	24	10.6
3	O	O	M	M	8	3.5
4	O	M	M	M	6	2.7

Table 1: Missingness patterns and the frequencies with which they occur. 'O' indicates observed and 'M' indicates missing.

4.2. Data file characteristics

Program execution requires a SAS data file with standard variable characteristics. SAS variables should be standardized as follows:

- ▷ `Subject`: Subject number (positive integer).
- ▷ `Rep`: Repetition or occasion number (positive integer).
- ▷ `Cov1`: First covariate (real number), if any.
- ▷ `Cov2`: Second covariate (real number), if any,
- ▷ ...
- ▷ `Group`: Group number (positive integer).
- ▷ `Outcome`: Outcome values (real number), where '.' indicates missing values.

It is important to note that the name of covariates should start with '`Cov`'. The program does not handle data sets with missing values in the first outcome value and/or in covariates, if any.
In the case study, the contents of `Data` for subjects 1, 2, 3 take the form:

Subject	Rep	Cov1	Group	Outcome
1	1	59	2	55
1	2	59	2	45
1	3	59	2	.
1	4	59	2	.
2	1	65	2	70
2	2	65	2	65
2	3	65	2	65
2	4	65	2	55
3	1	40	1	40
3	2	40	1	37
3	3	40	1	17
3	4	40	1	.

4.3. Specification of paths to access files

The program needs to access files. The path to access the working directory must be specified at the top of the SAS file `PMM.sas` in the SAS macro variable `&Path2WorkDir`. In the case study, the default path is specified as:

```
%let Path2WorkDir = "C:/WorkDir";
```

Then, the paths to access the working directory and the file `SAS.exe` must be specified at the top of the R file `PMM.R`. This can be done using the R editor or any other text editor (e.g., **WordPad**). In the case study, the default paths are specified as:

```
Path2WorkDir = "C:/WorkDir"
path2SASexe = "C:/Program Files/SASHome/SASFoundation/9.3"
```

Nothing else needs to be specified in the R script.

4.4. Specification of model parameters

The user must indicate the fixed-effect parameters in the SAS macro variable `&Covariates`. The full group-by-time interaction should be specified with parameters `Int1`, `Int2`, ... for each time point effect and `Group1`, `Group2`, ... for between-group differences at each time point. Covariates should be specified with `Cov1`, `Cov2`, ... as for the SAS variables in the data file.

In the case study, the fixed-effect parameters are specified as:

```
%let Covariates = Cov1 Int1 Int2 Int3 Int4 Group1 Group2 Group3 Group4;
```

The columns of the design matrix of fixed effects, except covariates, can be derived in the data step `DefineColumns`. In the case study, these are derived as follows:

```
data DefineColumns;
        set DataFile.Data;
        Int1 = 0; Int2 = 0; Int3 = 0; Int4 = 0;
        Group1 = 0; Group2 = 0; Group3 = 0; Group4 = 0;
        if Rep = 1 then Int1 = 1;
        if Rep = 2 then Int2 = 1;
        if Rep = 3 then Int3 = 1;
        if Rep = 4 then Int4 = 1;
        if Rep = 1 and Group = 2 then Group1 = 1;
        if Rep = 2 and Group = 2 then Group2 = 1;
        if Rep = 3 and Group = 2 then Group3 = 1;
        if Rep = 4 and Group = 2 then Group4 = 1;
run;
```

4.5. Parameterization of imputation stage

The user must specify parameter values for imputation in several SAS macro variables:

- `&nBatches`: Number of batches of imputations (positive integer).
- `&nImputations`: Number of imputations by batch (positive integer).
- `&Restriction`: Identifying restriction to be chosen among CCMV, ACMV, NCMV, NFMV-CC (for NFMV_{CC}), and NFMV-NC (for NFMV_{NC}).
- `&Delta`: Location parameter under NFMV_{CC} and NFMV_{NC}.
- `&Rounding`: Number of decimals of the imputed outcome values (positive integer).

A value of `&Delta` is required if the identifying restriction chosen is either NFMV_{CC} or NFMV_{NC}. Otherwise, `&Delta` is ineffective.

In the case study, the default values of imputation parameters are set to:

```
%let nSteps = 5;
%let nImputations = 100;
%let Restriction = NFMV-CC;
%let Delta = 2;
%let Rounding = 1;
```

This parameterization means that PMM analysis will be conducted under the identifying restriction NFMV-CC with the location parameter $\Delta = 2$ based on five batches of 100 imputation, that is 500 imputations totally. Missing outcome values will be imputed with a rounding of one decimal.

Nothing else needs to be specified for program execution.

5. Analysis of the case study

5.1. Pattern parameter estimation

Pattern parameters are estimated using SAS `PROC MIXED` which fits longitudinal outcome values with the fixed effects specified in the SAS macro variable `&Covariates` and an unstructured error covariance matrix. The syntax used for this is:

```
proc Mixed data = OutcomeSort method = ml noclprint noitprint asycov covtest;
        class Subject Rep;
        model Outcome = %str(&Covariates) / noint s covb;
        ods output solutionf = Solution;
        ods output covb = CovB;
        ods output covparms = CovParms;
        ods output asycov = AsyCov;
        repeated Rep / subject = Subject type = UN r;
        by Pattern;
run;
```

SAS `PROC MIXED` produces the SAS data files `Solution`, `CovB`, `CovParms`, and `AsyCov`, that respectively contain the estimates $\widehat{\boldsymbol{\beta}}_t$, $\widehat{\mathsf{VAR}}(\widehat{\boldsymbol{\beta}}_t)$, $\text{col}(\widehat{\Sigma}_t)$, and $\widehat{\mathsf{VAR}}(\text{col}(\widehat{\Sigma}_t))$, $t = 1, \ldots, T$. These estimates are then stored in the SAS data file `Estimates2R`.

At the end of this stage, three SAS data files are exported to R:

- `Estimates2R` which contains the pattern parameter estimates.
- `Outcome2R` which contains the outcome values and the derived variables:
  - `MissInter`: Indicator of intermittent missing values.
  - `MissDrop`: Indicator of missing values due to dropout.

  - `Pattern`: Subject pattern number.
  - `nValues`: Number of outcome values before subject dropout.

- `MIparameters2R` contains the parameter values for imputation.

5.2. Imputation

Imputation is entirely conducted using R. The file `PMM.R` consists of four parts which are described here-below. Further details about programming rules are given in the script.

R functions

Four R functions are defined at the top of the script:

▷ `rMVNorm(n, Mu, Cov)` generates `n` random vectors which are drawn from a multinormal distribution with mean `Mu` and variance matrix `Cov`.

▷ `MVCond(Y, X, Beta, S, IndicY1, IndicY2)` derives the conditional mean and variance matrix of `Y` vector values given others. `X` denotes the design matrix, `Beta` contains the parameter values, and `S` is the variance matrix of `Y`. `IndicY1` indicates the observed values of `Y` and `IndicY2` the given ones.

▷ `fMatrix(Col)` produces a complete symmetric matrix from the vector `Col` which contains the coefficients of the diagonal and the lower part of the matrix.

▷ Under ACMV and NFMV, `fRdraw(OutcomeS, BetaImput, SigmaImput, IndicColS, nPatterns, nPatternsSIv, imput, val)` is used to select the pattern of imputation for the `imput`th imputation of the `val`th missing outcome value. `OutcomeS` is a data frame which contains subject information, `BetaImput` contains the values of the $\widehat{\boldsymbol{\beta}}_t^{(m)}$'s and `SigmaImput` the values of the col$(\widehat{\Sigma}_t^{(m)})$'s, whereas `IndicColS` indicates the design matrix columns among the `OutcomeS` columns. Last, `nPatterns` is the total number of patterns and `nPatternsSIv` is the number of patterns available for imputation. To select the pattern of imputation, `fRdraw()` produces the vector `OmegaCumSIv` which contains the ascending values of $\Sigma_{j=s}^{k}\omega_{sj}^{(m)}$ and the scalar `U` which contains the value of $U^{(m)}$ as specified in (12).

Generation of $\boldsymbol{\beta}_t$ values by imputation

The values of $\widehat{\boldsymbol{\beta}}_t^{(m)}$, $m = 1, \ldots, M$, are randomly generated from distributions $\mathrm{N}(\widehat{\boldsymbol{\beta}}_t, \widehat{\mathsf{VAR}}(\widehat{\boldsymbol{\beta}}_t))$, $t = 1, \ldots, T$, using `rMVNorm()`. Values are then stored in the data frame `BetaImput` in which $\widehat{\boldsymbol{\beta}}_t^{(m)}$ coefficients are numbered according to the ordering number of the fixed effects specified in `&Covariates`.

In the case study, the nine fixed effects specified in `&Covariates` are `Cov1 Int1 Int2 Int3 Int4 Group1 Group2 Group3 Group4`. For example, `Cov1 Int1 Group1` are the three identified parameters in Pattern 1. These correspond to the first, the second, and the sixth fixed effects in `&Covariates`. This numbering is kept for $\widehat{\boldsymbol{\beta}}_1^{(1)}$ coefficients in `BetaImput` as shown here-below:

```
R> subset(BetaImput, pattern == 1 & imputation == 1)

pattern imputation number      value
   1          1      1      -0.45256755
   1          1      2      98.79422220
   1          1      3       0.00000000
   1          1      4       0.00000000
   1          1      5       0.00000000
   1          1      6     -25.40001710
   1          1      7       0.00000000
   1          1      8       0.00000000
   1          1      9       0.00000000
```

Generation of Σ_t values by imputation

The values of $\text{col}(\widehat{\Sigma}_t^{(m)})$, $m = 1, \ldots, M$, are randomly generated from the distributions $\text{N}(\text{col}(\widehat{\Sigma}_t), \widehat{\text{VAR}}(\text{col}(\widehat{\Sigma}_t)))$, $t = 1, \ldots, T$, using `rMVNorm()`. Values are then stored in the data frame `SigmaImput` in which $\text{col}(\widehat{\Sigma}_t^{(m)})$ coefficients are numbered according to their ordering number by column by row after transposition into a virtual $T \times T$-dimensional matrix.

In the case study, the virtual matrix is 4×4-dimensional and contains $\text{T(T+1)}/2 = 10$ coefficients in its diagonal and lower triangular part. The values and the numbering of the coefficients of $\text{col}(\widehat{\Sigma}_1^{(1)})$ and $\text{col}(\widehat{\Sigma}_2^{(1)})$ in `SigmaImput` are shown here-below:

```
R> subset(SigmaImput, (pattern == 1 | pattern == 2) & imputation == 1)

pattern imputation number       value
   1          1       1      19.4057844
   1          1       2       0.0000000
   1          1       3       0.0000000
.....................................
.....................................
   2          1       1     106.7157113
   2          1       2     117.3420276
   2          1       3       0.0000000
   2          1       4       0.0000000
   2          1       5     137.4193741
   2          1       6       0.0000000
   2          1       7       0.0000000
   2          1       8       0.0000000
   2          1       9       0.0000000
   2          1      10       0.0000000
```

Imputation of missing outcome values

The program imputes first intermittent missing outcome values using subject pattern. Suppose that the ith subject belongs to pattern t and denote by $\mathbf{Y}_{i,obs}$ the vector of observed

outcome values and $\mathbf{Y}_{i,miss}$ the vector of intermittent missing ones. The conditional mean $\widehat{\mu}^{(m)}_{i,t,2|1}$ and variance matrix $\widehat{\Sigma}^{(m)}_{t,2|1}$, where $2|1$ stands for $\mathbf{Y}_{i,miss}|\mathbf{Y}_{i,obs}$, are derived using `MVCond()`. Then, outcome values are randomly drawn from the conditional pattern distributions $f^{(m)}_t(\mathbf{Y}_{i,miss}|\mathbf{Y}_{i,obs}) \sim \mathrm{N}(\widehat{\mu}^{(m)}_{i,t,2|1}, \widehat{\Sigma}^{(m)}_{t,2|1})$, $m = 1, \ldots, M$, using `rMVNorm()`.

We now describe how the program imputes missing outcome values due to dropout. We illustrate the process with the case study through one of the imputations, say imputation 1 ($m = 1$), in subject 11. This subject has only one observed outcome value, $y_{11,1} = 50$, and so, the three outcome values $y_{11,2}, y_{11,3}, y_{11,4}$ need to be imputed.

Initial information about subject 11 is shown here-below:

```
R> subset(Outcome, subject == 11)

subject rep group cov1 outcome missdrop missinter pattern nmeasures
   11   1     2   58      50       0         0       1         1
   11   2     2   58      NA       1         0       1         1
   11   3     2   58      NA       1         0       1         1
   11   4     2   58      NA       1         0       1         1

int1 int2 int3 int4 group1 group2 group3 group4
  0    0    1    0     0      0      1      0
  0    0    0    1     0      0      0      1
  1    0    0    0     1      0      0      0
  0    1    0    0     0      1      0      0
```

After each imputation in each subject, the R script produces:

- A matrix `MatOutcomeSI` which contains the random draws of missing outcome values in all available patterns (row number indicates outcome value number and column number indicates pattern number).
- A vector `YSI` which contains the observed and the imputed values of outcome.

In the case study, matrices `MatOutcomeSI` are 4×4 dimensional. In subject 11, the first row of `MatOutcomeSI` is not filled as $y_{11,1}$ is observed. Then, imputation of missing outcome values is conducted sequentially by value.

If the identifying restriction chosen is either CCMV, NCMV, or ACMV, random values of $y_{11,2}$ are first drawn in all available patterns from:

$$\triangleright\ f_t(y_{11,2}|y_{11,1}) \sim \mathrm{N}(\widehat{\mu}^{(1)}_{11,t,2|1}, \widehat{\Sigma}^{(1)}_{t,2|1}), \qquad t = 2, 3, 4,$$

where $2|1$ stands for $y_{11,2}|y_{11,1}$. Values are stored in the second row of `MatOutcomeSI` which, in our example, takes the form:

```
R> MatOutcomeSI

     [,1]     [,2]     [,3]     [,4]
[1,]    0  0.00000  0.00000  0.00000
```

```
[2,]    0 39.76904 65.83838 45.91795
[3,]    0  0.00000  0.00000  0.00000
[4,]    0  0.00000  0.00000  0.00000
```

Then, the rule to select the pattern of imputation is driven by the identifying restriction chosen. Under CCMV, the missing $y_{11,2}$ value is imputed with 45.91795 of Pattern 4. Under NCMV, the imputed value is 39.76904 of Pattern 2.

Under ACMV, the pattern of imputation is selected as described in (12) using `fRdraw()`. `fRdraw()` produces the vector `OmegaCumSIv` which contains the cumulative values $\Sigma_{j=2}^{t}\omega_{2j}^{(1)}$. In our example, `OmegaCumSIv` takes the form:

```
R> OmegaCumSIv

[1] 0.00000000 0.09073205 0.33557261 1.00000000
```

`fRdraw()` also produces the scalar `U` which contains a random draw from the uniform distribution. In our example, `U` takes the value:

```
R> U

[1] 0.1322
```

Since `U` $= 0.1322$ is comprised between $\Sigma_{j=2}^{3}\omega_{2j}^{(1)} = 0.09073205$ and $\Sigma_{j=2}^{4}\omega_{2j}^{(1)} = 0.33557261$, $y_{11,2}$ is imputed with the value of Pattern 3, which is 65.83838. This value is put into `YSI` which becomes:

```
R> YSI

[1] 50.00000 65.83838 0 0
```

The procedure described here-above is repeated to impute $y_{11,3}$ and $y_{11,4}$ by replacing $\mathbf{Y}_{i,obs}$ by `YSI`. When the process is completed, `MatOutcomeSI` takes the form:

```
R> MatOutcomeSI

     [,1]     [,2]     [,3]     [,4]
[1,]    0  0.00000  0.00000  0.00000
[2,]    0 39.76904 65.83838 45.91795
[3,]    0  0.00000 36.42852 39.41290
[4,]    0  0.00000  0.00000 37.42869
```

The complete vector `YSI` obtained is:

```
R> YSI

[1] 50.00000 65.83838 39.41290 37.42869
```

Now, if the identifying restriction chosen is either NFMV_{CC} or NFMV_{NC}, $y_{11,2}$ is imputed with a random value drawn from:

- $g_1^{(1)}(y_{11,2}|y_{11,1}) \sim \text{N}(\widehat{\mu}_{11,4,2|1}^{(1)}+\Delta, \widehat{\Sigma}_{4,2|1}^{(1)})$ under NFMV_{CC},
- $g_1^{(1)}(y_{11,2}|y_{11,1}) \sim \text{N}(\widehat{\mu}_{11,2,2|1}^{(1)}+\Delta, \widehat{\Sigma}_{2,2|1}^{(1)})$ under NFMV_{NC},

where 2|1 stands for $y_{11,2}|y_{11,1}$.

Imputations of $y_{11,3}$ and $y_{11,4}$ are based on multiple random draws. Then, the pattern of imputation is selected as under ACMV using `fRdraw()`.

For $y_{11,3}$, the random values are drawn from:

- $g_2^{(1)}(y_{11,3}|y_{11,1}, y_{11,2}) \sim \text{N}(\widehat{\mu}_{11,4,2|1}^{(1)}+\Delta, \widehat{\Sigma}_{4,2|1}^{(1)})$ under NFMV_{CC},
- $g_2^{(1)}(y_{11,3}|y_{11,1}, y_{11,2}) \sim \text{N}(\widehat{\mu}_{11,3,2|1}^{(1)}+\Delta, \widehat{\Sigma}_{3,2|1}^{(1)})$ under NFMV_{NC},
- $f_3^{(1)}(y_{11,3}|y_{11,1}, y_{11,2}) \sim \text{N}(\widehat{\mu}_{11,3,2|1}^{(1)}, \widehat{\Sigma}_{3,2|1}^{(1)})$,
- $f_4^{(1)}(y_{11,3}|y_{11,1}, y_{11,2}) \sim \text{N}(\widehat{\mu}_{11,4,2|1}^{(1)}, \widehat{\Sigma}_{4,2|1}^{(1)})$,

where 2|1 stands for $y_{11,3}|y_{11,1}, y_{11,2}$.

For $y_{11,4}$, the random values are drawn from:

- $g_3^{(1)}(y_{11,4}|y_{11,1}, y_{11,2}, y_{11,3}) \sim \text{N}(\widehat{\mu}_{11,4,2|1}^{(1)}+\Delta, \widehat{\Sigma}_{4,2|1}^{(1)})$,
- $f_4^{(1)}(y_{11,4}|y_{11,1}, y_{11,2}, y_{11,3}) \sim \text{N}(\widehat{\mu}_{11,4,2|1}^{(1)}, \widehat{\Sigma}_{4,2|1}^{(1)})$,

where 2|1 stands for $y_{11,4}|y_{11,1}, y_{11,2}, y_{11,3}$.

We provide here-below an example of `MatOutcomeSI` obtained under NFMV_{CC} and $\Delta = 2$ which is the default program parameterization. `MatOutcomeSI` takes the form:

```
R> MatOutcomeSI

           [,1]     [,2]     [,3]     [,4]
[1,]    0.00000  0.00000  0.00000  0.00000
[2,]   47.02344  0.00000  0.00000  0.00000
[3,]    0.00000 48.68755 44.07891 51.08928
[4,]    0.00000  0.00000 51.45901 42.98866
```

At the end of this stage, imputed outcome values are stored in the R data file `OutcomeImput2SAS`, which is exported to SAS.

5.3. Pooled analysis

The program fits outcome values by imputation using MMRM which incorporates a full group-by-time interaction at every time point for the fixed effects and an unstructured error covariance matrix. The syntax of the SAS `PROC MIXED` for this is:

```
proc Mixed data = OutcomeForPooledAnalysis method = ml noclprint noitprint
covtest;
title2 "Longitudinal outcome analysis by imputation";
    class Subject Rep;
        model Outcome = %str(&Covariates) / noint s covb;
        ods output solutionf = Solution;
        ods output covb = CovB;
        repeated Rep / subject = Subject type = UN r;
        by Imputation;
run;
```

Results are stored in the two SAS data files:

- `SolutionTotal` which contains the parameter estimates of fixed effects.
- `CovBTotal` which contains the variance matrix coefficients.

In the case study, the contents of `SolutionTotal` for the first imputation takes the form:

Effect	Estimate	StdErr	DF	tValue	Probt	_Imputation_
Cov1	0.8940	0.03572	225	25.03	<.0001	1
Int1	4.7352	2.1080	225	2.25	0.0257	1
Int2	3.7895	2.2431	225	1.69	0.0925	1
Int3	0.005495	2.3485	225	0.00	0.9981	1
Int4	-5.2882	2.4977	225	-2.12	0.0353	1
Group1	-2.6765	1.0772	225	-2.48	0.0137	1
Group2	-3.7053	1.5354	225	-2.41	0.0166	1
Group3	-3.0734	1.8285	225	-1.68	0.0942	1
Group4	-5.1362	2.1944	225	-2.34	0.0201	1

Then, the pooled analysis is performed using SAS `PROC MIANALYZE` with the syntax:

```
proc MIanalyze parms = SolutionTotal covb(effectvar = rowcol) = CovBTotal;
title2 "Combined analysis for fixed effects (Proc MIanalyze)";
modeleffects &Covariates;
run;
```

An example of SAS output for the case study with the default program parameterization `nSteps = 5`, `nImputations = 100`, `Restriction = NFMV-CC`, `Delta = 2` is given in Table 1. Results exhibits a treatment-effect estimate at visit 4 of -4.45 (2.37) with a significance level of $p = 0.06$.

6. Results under different identifying restrictions

Any inference should account for the uncertainty attributable to missing data so that the type I error rate is valid under the assumptions made. In clinical trials, the primary statistical approach is often based on the MAR assumption, under which MMRM provides valid inferences.

Pattern-mixture model under NFMV-CC identifying restriction and location parameter Delta=2
Pooled analysis of fixed effects

The MIANALYZE Procedure

Model Information

PARMS Data Set	WORK.SOLUTIONTOTAL
COVB Data Set	WORK.COVBTOTAL
Number of Imputations	500

Variance Information

Parameter	Variance: Between	Variance: Within	Variance: Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Cov1	0.000058534	0.001277	0.001336	258883	0.045919	0.043911	0.999912
Int1	0.177738	4.447967	4.626061	336689	0.040039	0.038504	0.999923
Int2	0.172711	5.026062	5.199119	450385	0.034432	0.033290	0.999933
Int3	0.196343	5.577757	5.774493	429893	0.035272	0.034074	0.999932
Int4	0.367578	6.288040	6.656353	162982	0.058574	0.055344	0.999889
Group1	0.000013640	1.159998	1.160012	3595E9	0.000011782	0.000011782	1.000000
Group2	0.040866	2.337020	2.377968	1.68E6	0.017521	0.017221	0.999966
Group3	0.176145	3.460291	3.636788	211866	0.051006	0.048540	0.999903
Group4	0.715017	4.906453	5.622899	30736	0.146021	0.127473	0.999745

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Cov1	0.897253	0.036550	0.8256	0.96889	258883	0.874079	0.921698	0	24.55	<.0001
Int1	4.557453	2.150828	0.3419	8.77301	336689	3.210413	5.834442	0	2.12	0.0341
Int2	3.602813	2.280158	-0.8662	8.07185	450385	2.290362	4.732946	0	1.58	0.1141
Int3	-0.161689	2.403017	-4.8715	4.54815	429893	-1.385394	1.020038	0	-0.07	0.9464
Int4	-5.276138	2.579991	-10.3329	-0.21941	162982	-7.448545	-3.678467	0	-2.05	0.0409
Group1	-2.674981	1.077038	-4.7859	-0.56402	3595E9	-2.686167	-2.663180	0	-2.48	0.0130
Group2	-4.008669	1.542066	-7.0311	-0.98627	1.68E6	-4.567307	-3.457197	0	-2.60	0.0093
Group3	-3.033814	1.907036	-6.7716	0.70393	211866	-4.297201	-1.992340	0	-1.59	0.1116
Group4	-4.451317	2.371265	-9.0991	0.19646	30736	-6.861047	-1.153242	0	-1.88	0.0605

Figure 1: Example SAS output for the case study with the default program parameterization.

However, the sensitivity of inferences to departures from MAR should be thoroughly investigated via sensitivity analyses under plausible MNAR mechanisms, although MNAR cannot be tested. This latter aspect has been discussed, demonstrated and exemplified in Molenberghs, Beunckens, Sotto, and Kenward (2008). If the conclusion assuming MAR differs from conclusions under plausible MNAR mechanisms, then careful scrutiny is necessary.

In the PMM framework, ACMV is the natural counterpart to MAR whereas the other identifying restrictions describe MNAR mechanisms. Here-below, we provide results of PMM analyses of the case study data under different identifying restrictions available in the program. Analyses are based on 10,000 imputations, which is a large number compared to what is generally recommended. Arguments to justify the number of imputations are given first.

The number of imputations is an important issue as it directly influences result accuracy and program execution time. Several aspects need to be considered and there is no obvious formal answer. Beyond the intuitive rule that the greater the number of imputations, the greater is result accuracy, the identifying restriction chosen and the amount of observations per pattern (e.g., majority completer versus spread over many patterns) are two important aspects to

Identifying restriction	Δ	Mean	Standard error	p value
CCMV		−4.75	2.37	0.044
ACMV		−4.69	2.42	0.053
NCMV		−4.30	2.52	0.088
NFMV_{CC}	0	−4.66	2.39	0.051
NFMV_{NC}	0	−4.46	2.47	0.071
NFMV_{CC}	2	−4.44	2.38	0.062
NFMV_{NC}	2	−4.25	2.46	0.085
NFMV_{CC}	4	−4.24	2.38	0.075
NFMV_{NC}	4	−4.03	2.46	0.102

Table 2: Treatment-effect estimates at visit 4 under CCMV, ACMV, NCMV, NFMV_{CC}, and NFMV_{NC} obtained with 10,000 imputations.

be considered. CCMV is more precise if there are a lot of completers, as opposed to some neighboring patterns that are rather lightly filled.

As shown in Section 4.1, the case study exhibits a moderate dropout rate of 16.8%. The relative increase in variance due to missingness can be used to quantify how missingness contributes to inferential uncertainty. In the SAS output shown in Section 5, the value 0.146 for the treatment-effect parameter at visit 4 characterizes a moderate contribution. This information can be supplemented by the relative efficiency, introduced by Rubin (1987), which indicates the magnitude of the point estimate's standard error of the current run rather than based upon an infinite number of imputations. In our example, the relative efficiency is greater than 0.999 and tends to demonstrate that the default size of 500 imputations is largely sufficient to obtain accurate inferential results. However, the systematic use of such diagnostic tools to determine the number of imputations raises several criticisms because the true value of these quantities is unknown and varies across distinct runs for the same data and analysis; see, e.g., Bodner (2008). Moreover, one can easily execute several times the case study based on 500 imputations and observe that results exhibit some variation across different runs.

A practical alternative to this approach consists of pre-defining accuracy levels on desired inferential parameters in line with the objectives of the study. On the basis of several runs of different sizes, the user selects the number of imputations that guarantees the stability of results at the pre-defined accuracy levels. Such investigation should be conducted under NCMV or NFMV_{NC}, which generate greater between-imputation variances than CCMV or NFMV_{CC}, respectively.

A simple exploratory analysis requires less accuracy than an analysis directed to regulatory bodies. In the case study, a size of 100 to 500 imputations is perfectly conceivable for a quick exploration, whereas 1000 imputations guarantee an accuracy level of 1 decimal to the treatment-effect estimate at visit 4. A size of 10,000 imputations guarantees accuracy levels of two decimals to the estimate and three decimals to the significance level. Table 2 shows the treatment-effect estimates at visit 4 under CCMV, ACMV, NCMV, NFMV_{CC}, and NFMV_{NC} obtained with 10,000 imputations. Under NFMV_{CC} and NFMV_{NC}, Δ was set to $0, 2$ and 4.

The difference between the result under ACMV and the results under the MNAR mechanisms is moderate, although this difference may have an impact on the statistical conclusion if the significance level is set to 0.05. NFMV assumes that any dropout is associated with an

outcome decrease by Δ at the first unobserved visit. The decrease of treatment-effect estimates under NFMV as the value of Δ increases was expected since there are twice more dropout subjects in the test group, 25/111 (22.5%), than in the placebo group, 13/115 (11.3%).

7. Specifications of other PMM analyses

The great flexibility of the program allows different kinds of investigation. This flexibility is illustrated with the case study through the implementation of four PMM analyses.

7.1. Introduction of a class effect

Analysis of a class effect in the PMM framework requires the user to ensure that enough subjects are available by class modality per pattern to allow pattern parameter estimation. We now add a fictitious center effect with three modalities (i.e., three different investigation sites) to the statistical model of the case study. We also assume that a variable `Centre` identifies centers with values 1, 2, and 3 in the SAS data set `Data`.

The `Centre` effect is incorporated in the model via two fixed-effect parameters. According to Section 4, the names of these parameters must start by '`Cov`' as they will be analyzed as any other covariate. The center-effect parameters will be denoted by `CovInv1` and `CovInv2` in what follows. The model is specified in the SAS macro variable `&Covariates` as:

```
%let Covariates = Cov1 CovInv1 CovInv2 Int1 Int2 Int3 Int4
Group1 Group2 Group3 Group4;
```

Then, the columns of the design matrix associated with the center-effect parameters are obtained from the indicator SAS variables `CovInv1` and `CovInv2`. These can be derived in the data step `DefineColumns` using the syntax:

```
CovInv1 = 0; CovInv2 = 0;
if Centre = 1 then CovInv1 = 1;
if Centre = 2 then CovInv2 = 1;
```

7.2. Introduction of an interaction term

We now describe how to incorporate an interaction of baseline by visit in the case study. Let us denote by `Cov11 Cov12 Cov13 Cov14` the parameters of the baseline values by visit, the model is specified in the SAS macro variable `&Covariates` as:

```
%let Covariates = Cov11 Cov12 Cov13 Cov14 Int1 Int2 Int3 Int4
Group1 Group2 Group3 Group4;
```

Then, the columns of the design matrix associated with the interaction terms are obtained from the SAS variables `Cov11 Cov12 Cov13 Cov14`. These can be derived in the data step `DefineColumns` using the syntax:

```
Cov11 = 0; Cov12 = 0; Cov13 = 0; Cov14 = 0;
if Rep = 1 then Cov11 = Cov1;
```

```
if Rep = 2 then Cov12 = Cov1;
if Rep = 3 then Cov13 = Cov1;
if Rep = 4 then Cov14 = Cov1;
```

In the case study, this model implies that SAS `PROC MIANALYZE` does not provide pooled inferences for the treatment effect at visit 1 since the between-imputation variance is zero. Such a situation is not uncommon in a more general context as there may always be parameters for which there is no missing information. Anyway, treatment-effect inferences at visit 1 can be retrieved from any of the individual imputations, because all complete data sets will provide the same results for that particular parameter.

7.3. Use of different models for imputation and analysis

If different models are used for imputation and analysis, the SAS macro variable `&Covariates` cannot specify both models anymore. If `&Covariates` is used to specify the imputation model, the analysis model must be specified manually in SAS `PROC MIXED`, line 272, and in SAS `PROC MIANALYZE`, line 309.

7.4. Pooling of patterns

In applications with many time points, or when sample size by pattern is too low, it may be more reasonable (or necessary due to sparseness) to just define patterns of early, middle, and late dropouts. However, the program handles patterns that are defined based on dropout at every time point, except baseline. To overcome this, pattern numbers must be decoupled from time point numbers. We now show how to reduce the number of patterns in the case study to three by pooling Patterns 2 and 3.

The pooling of patterns can be specified in the SAS data step `Outcome2R` by incorporating first Pattern 3 into Pattern 2 and then Pattern 4 into Pattern 3. This is done, below line 105, by inserting the syntax:

```
data DataFile.Outcome2R;
        set DataFile.Outcome2R;
        if Pattern = 3 then Pattern = 2;
        if Pattern = 4 then Pattern = 3;
run;
```

No other modifications are needed in the SAS code.

Next, the instruction to calculate the number of available patterns for the imputation of each missing value must be modified in the R script. The default instruction `nPatternsSIv = nPatterns - val + 1` appears twice.

Line 274, to calculate the number of available patterns for imputation under CCMV, ACMV, and NCMV, the default instruction must be replaced by the syntax:

```
if (val <= 2) {
  nPatternsSIv = nPatterns - val + 1
} else
  nPatternsSIv = nPatterns - val + 2
```

Line 324, to calculate the number of available patterns for imputation under NFMV, the default instruction must be replaced by the syntax:

```
if (val <= 3) {
  nPatternsSIv = nPatterns - val + 1
} else
  nPatternsSIv = nPatterns - val + 2
```

7.5. Extension to linear mixed-effects models

In experiments with many time points, MMRM as specified in (9) may be inappropriate to fit outcome values over time. Alternatively, the so-called random-coefficients model toward extends the regression model to longitudinal outcomes.

Let us assume, in the case study, that the outcome values are described by an average trend in function of the time since randomization. We further base the model on the assumption that, for every subject i who belongs to group g, this trend can be modeled by a quadratic regression, but with subject-specific coefficients. This explicitly allows the outcome trajectories to vary by subject.

Denoting the time since randomization at visits by $Time_k$, $k = 1, \ldots, t$ $(t < T)$, one formally obtains

$$y_{gik} = (\beta_{0g} + b_{0gi}) + (\beta_{1g} + b_{1gi})\, Time_k + (\beta_{2g} + b_{2gi})\, Time_k^2 + \epsilon_{gik}, \tag{14}$$

where $\boldsymbol{\epsilon}_i = (\epsilon_{gi1}, \epsilon_{gi2}, \ldots, \epsilon_{git})$ is assumed to be normally distributed with mean vector zero and some covariance matrix Σ_i. Because subjects are randomly sampled from a population of subjects, it is natural to assume that the subject-specific coefficients in $\mathbf{b}_i = (b_{0gi}, b_{1gi}, b_{2gi})^\top$ are normally distributed with mean zero and covariance G.

The above model is a special case of the general linear mixed-effects model which assumes that $\mathbf{Y}_{i,obs}$ satisfies

$$\begin{aligned} \mathbf{Y}_{i,obs}|\mathbf{b}_i &= X_i\boldsymbol{\beta} + Z_i\mathbf{b}_i + \boldsymbol{\epsilon}_i, \\ \mathbf{b}_i &\sim N(\mathbf{0}, G), \end{aligned} \tag{15}$$

where X_i contains the known fixed-effects covariates and Z_i contains the known subject-specific covariates. We have $\mathbf{b}_i \sim N(\mathbf{0}, G)$ and $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \Sigma_i)$, where the $\mathbf{b}_i$'s and $\boldsymbol{\epsilon}_i$'s are independent. The error covariance matrix sometimes simplifies to $\Sigma_i = \sigma I_{ni}$ or other structures. To specify model (14), times of measurement must be incorporated into X_i and Z_i.

Implementation of linear mixed-effects models in the PMM framework is straightforward from Section 2.2. If patterns are based on missingness at every time point, PMM can be specified from (15) by replacing the parameters $\boldsymbol{\beta}$, G, and Σ_i by the pattern-specific parameters $\boldsymbol{\beta}_t$, G_t, and Σ_t. If patterns are based on combinations of visits, the decoupling between pattern number and visit number is accommodated in the program (see indications in the Section 7.4).

Missing outcome values are drawn from conditional pattern distributions where patterns are selected by the identifying restriction chosen. As in Section 2.2, we now suppose that a pattern r is used to impute $y_{i,t+1}$ whereas $y_{i,1}, \ldots, y_{i,t}$ are observed. The mean and variance of conditional pattern distributions are directly and simply obtained by replacing Σ_r by $V_i = Z_i G_r Z_i^\top + \Sigma_r$ in the formulas yielding (11).

The uncertainty pertaining to the matrix G_r is incorporated through the Bayesian posterior predictive distribution of coefficients, as for the error covariance matrix Σ_r. On the basis of non-informative Jeffreys' priors in the Gaussian setting, the values of $\widehat{G}_r^{(m)}$ by imputation, $m = 1, \ldots, M$, are randomly drawn from the posterior distributions $\mathrm{N}(\mathrm{col}(\widehat{G}_r)), \widehat{\mathsf{VAR}}(\mathrm{col}(\widehat{G}_r))$, where $\mathrm{col}(\widehat{G}_r)$ is the vector containing the coefficients of $\widehat{G}_r$.

The values of $y_{i,t+1}$ are imputed after the derivation of $\widehat{V}_i^{(m)} = Z_i \widehat{G}_r^{(m)} Z_i^\top + \widehat{\Sigma}_r^{(m)}$. As described in Section 2.2 and using the notation of (11), the imputed values are drawn from the conditional pattern distributions, which are expressed by:

$$f_r^{(m)}(y_{i,t+1}|y_{i,1}, \ldots, y_{i,t}) \sim N(\widehat{\mu}_{i,r,2|1}^{(m)}, \widehat{V}_{i,2|1}^{(m)}), \qquad m = 1, \ldots, M.$$

The implementation of model (14) with the program requires:

- In `PMM.sas`: To specify the random effects in SAS `PROC MIXED` with the statement `Random` and to estimate G_t's coefficients and correlations with the options `G GCORR`. Note that the specification of the error covariance matrix structure requires to copy the variable for time in another variable that will be included in the statements `Class` and `Repeated` (see Verbeke and Molenberghs 2000, p. 94).
- In `PMM.sas`: To put estimates into the export data file `Estimate2R` to R.
- In `PMM.R`: To generate the values $\widehat{G}_r^{(m)}$ by imputation and derive $\widehat{V}_{i,2|1}^{(m)}$.

7.6. Extension to other applications

The program implements MMRM with full group-by-time interaction and an unstructured error covariance matrix. As shown in the previous section, it is perfectly feasible to implement alternative modeling strategies such as linear mixed-effects models, but also non-linear models using SAS `PROC NLMIXED`. Under this perspective, the present version of the program can be seen as a general framework whose algorithm can be adapted to different contexts.

The combination of the SAS and R functionalities and the call of R from SAS was a deliberate choice, as the program is primarily directed to SAS users. However, multiple imputation under identifying restriction is implemented in the R script. SAS is only used for parameter estimation and pooled analysis and can be replaced by any other software, including R packages, that have capabilities similar to SAS `PROC MIXED` and SAS `PROC MIANALYZE`. Moreover, implementing all the program in R would save execution time by avoiding the calls from SAS to R.

Acknowledgments

The authors thank the two anonymous reviewers for their relevant and constructive comments on earlier draft versions that helped to improve the quality of this manuscript. Financial support from the IAP research network #P7/06 of the Belgian Government (Belgian Science Policy) is gratefully acknowledged. The research leading to these results has also received funding from the European Seventh Framework programme FP7 2007–2013 under grant agreement Nr. 602552.

References

Bodner T (2008). "What Improves with Increased Missing Data Imputation?" *Structural Equation Modeling*, **15**(4), 651–675. doi:10.1080/10705510802339072.

Faria J, Grosjean P, Jelihovschi E, Pietrobon R, Silva Farias P (2015). ***Tinn-R*** *Editor – GUI for R Language and Environment*. Version 4.0.3.5, URL http://nbcgib.uesc.br/lec/software/editores/tinn-r/en.

Harrel F (2015). ***Hmisc****: Harrell Miscellaneous*. R package version 3.16-0, URL http://CRAN.R-project.org/package=Hmisc.

I-BioStat (2007). *MNAR Analysis and Sensitivity Analyses*. URL http://ibiostat.be/online-resources/online-resources/uploads/mcar.zip.

Kenward M, Molenberghs G (2003). "Pattern-Mixture Models with Proper Time Dependence." *Biometrika*, **90**(1), 53–71.

Li K, Raghunathan T, Rubin D (1991). "Large Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics & an F Reference Distribution." *Journal of the American Statistical Association*, **86**(416), 1065–1073. doi:10.1080/01621459.1991.10475152.

Little R (1993). "Pattern-Mixture Models for Multivariate Incomplete Data." *Journal of the American Statistical Association*, **88**(421), 125–134. doi:10.2307/2290705.

Molenberghs G, Beunckens C, Sotto C, Kenward M (2008). "Every Missingness Not At Random Model Has a Missingness At Random Counterpart With Equal Fit." *Journal of the Royal Statistical Society B*, **70**(2), 371–388. doi:10.1111/j.1467-9868.2007.00640.x.

Molenberghs G, Michiels B, Kenward M, Diggle P (1998). "Missing Data Mechanisms and Pattern-Mixture Models." *Statistica Neerlandica*, **52**(2), 153–161. doi:10.1111/1467-9574.00075.

National Research Council (2010). *Prevention and Treatment of Missing Data in Clinical Studies*. The National Academies Press, Washington, DC. Panel on Handling Missing Data in Clinical Trials. Committee on National Statistics, Division of Behavioral and Social Sciences and Education.

R Core Team (2015a). ***foreign****: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...* R package version 0.8-66, URL http://CRAN.R-project.org/package=foreign.

R Core Team (2015b). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Ripley B (2015). ***MASS****: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-44, URL http://CRAN.R-project.org/package=MASS.

Rubin D (1987). *Multiple Imputation for Nonresponses in Surveys*. John Wiley & Sons. doi:10.1002/bimj.4710310118.

SAS Inc (2011). *SAS/IML 9.3 Software*. Cary. URL http://www.sas.com/.

SAS Inc (2014). *SAS/STAT 9.4 Software*. Cary. URL http://www.sas.com/.

Thijs H, Molenberghs G, Michiels B, Verbeke G, Curran D (2002). "Strategies to Fit Pattern-Mixture Models." *Biostatistics*, **3**(2), 245–265. doi:10.1007/978-1-4757-3625-0_27.

Venables W, Ripley B (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

Verbeke G, Molenberghs G (2000). *Linear Mixed Models for Longitudinal Data*. Springer-Verlag. doi:10.1007/978-1-4419-0300-6.

Affiliation:

Pierre Bunouf
Laboratoires Pierre Fabre
142, rue du village d'entreprises
31670 Labege, France
E-mail: pierre.bunouf@pierre-fabre.com

Geert Molenberghs, Herbert Thijs
I-BioStat, Universiteit Hasselt & Katholieke Universiteit Leuven
Agoralaan building D
3590 – Diepenbeek, Belgium
E-mail: geert.molenberghs@uhasselt.be, herbert.thijs@uhasselt.be

Jean-Marie Grouin
Université de Rouen – INSERM U 657
rue Lavoisier
76821 Mont Saint-Aignan, France
E-mail: Jean-Marie.Grouin@univ-rouen.fr

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
December 2015, Volume 68, Issue 8 *Submitted:* 2013-02-20
doi:10.18637/jss.v068.i08 *Accepted:* 2015-01-12