2016

Sensitivity Analysis for Non-Ignorable Dropout of Marginal Treatment Effect in Longitudinal Trials for G-Computation Based Estimators

Emin Tahirovic
University of Pennsylvania, emint@upenn.edu

Recommended Citation

Tahirovic, Emin, "Sensitivity Analysis for Non-Ignorable Dropout of Marginal Treatment Effect in Longitudinal Trials for G-Computation Based Estimators" (2016). *Publicly Accessible Penn Dissertations*. 2049.
https://repository.upenn.edu/edissertations/2049

Sensitivity Analysis for Non-Ignorable Dropout of Marginal Treatment Effect in Longitudinal Trials for G-Computation Based Estimators

Abstract

We specify identifying assumptions under which linear increments (LI) estimator can be used to estimate unconditional expectation for longitudinal data from a clinical trial in the presence of dropout. We show that these are analog conditions under which extended linear SWEEP estimator achieves unbiased estimation of the identical parameter in the same setting. Within a class of linear autoregressive models we specify how strategies implemented in LI and extended SWEEP relate to each other w.r.t. the conditional expectation of increments and outcomes respectively. We utilize conceptual overlap of these two methods to define a sensitivity analysis for both of them in presence of non-ignorable dropout. Interdependency of these two approaches offers a natural solution to a prominent problem of asynchronous association between outcome and dropout inevitably encountered in sensitivity analysis for dropout in longitudinal data. Validation of our approach is done on the data coming from a randomized, longitudinal trial of behavioral economic interventions to reduce CVD risk. We subsequently show that our approach to sensitivity analysis can be perceived as extension of the pattern mixture method defined by Daniels and Hogan in 2007. to longer sequences of observations. For T=3 we give the explicit expression for bias of our approach w.r.t. mentioned pattern mixture approach. We further show on a subset of the data from the same study that this bias does not invalidate our sensitivity analysis for LI when it comes to evaluating the robustness of findings under increasingly less ignorable dropout.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Epidemiology & Biostatistics

First Advisor

Andrea B. Troxel

Keywords

extended SWEEP, Granger-non causality, linear increments, longitudinal data, non-ignorable dropout, sensitivity analysis

Subject Categories

Biostatistics

SENSITIVITY ANALYSIS FOR NON-IGNORABLE DROPOUT OF MARGINAL TREATMENT EFFECT IN LONGITUDINAL TRIALS FOR G-COMPUTATION BASED ESTIMATORS

Emin Tahirović

A DISSERTATION

in

Epidemiology and Biostatistics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2016

Supervisor of Dissertation

Andrea B. Troxel

Professor of Biostatistics

Graduate Group Chairperson

John H. Holmes, Professor of Medical Informatics in Epidemiology

Dissertation Committee

Sharon X. Xie, Associate Professor of Biostatistics

Kevin G. Lynch, Associate Professor of Biostatistics in Psychiatry

Kevin Volpp, Professor of Medicine

SENSITIVITY ANALYSIS FOR NON-IGNORABLE DROPOUT OF MARGINAL TREATMENT EFFECT IN LONGITUDINAL TRIALS FOR G-COMPUTATION BASED ESTIMATORS

ACKNOWLEDGEMENT

I would like to thank my mother and fater as well as my sister and her family for their constant, unambiguous and unapologetic support. Their words of encouragement during our long Skype talks helped me on numerous occasions not to waver, but to preserve. I would like to thank my advisor Prof. Andrea B. Troxel for her time and patience and all the support and good advice during my PhD studies. Last, but not least, I would like to thank my thesis committee and my fellow students.

ABSTRACT

SENSITIVITY ANALYSIS FOR NON-IGNORABLE DROPOUT OF MARGINAL TREATMENT EFFECT IN LONGITUDINAL TRIALS FOR G-COMPUTATION BASED ESTIMATORS

Emin Tahirović

Andrea B. Troxel

We specify identifying assumptions under which linear increments (LI) estimator can be used to estimate unconditional expectation for longitudinal data from a clinical trial in the presence of dropout. We show that these are analog conditions under which extended linear SWEEP estimator achieves unbiased estimation of the identical parameter in the same setting. Within a class of linear autoregressive models we specify how strategies implemented in LI and extended SWEEP relate to each other w.r.t. the conditional expectation of increments and outcomes respectively. We utilize conceptual overlap of these two methods to define a sensitivity analysis for both of them in presence of non-ignorable dropout. Interdependency of these two approaches offers a natural solution to a prominent problem of asynchronous association between outcome and dropout inevitably encountered in sensitivity analysis for dropout in longitudinal data. Validation of our approach is done on the data coming from a randomized, longitudinal trial of behavioral economic interventions to reduce CVD risk. We subsequently show that our approach to sensitivity analysis can be perceived as extension of the pattern mixture method defined by Daniels and Hogan in 2007. to longer sequences of observations. For T=3 we give the explicit expression for bias of our approach w.r.t. mentioned pattern mixture approach. We further show on a subset of the data from the same study that this bias does not invalidate our sensitivity analysis for LI when it comes to evaluating the robustness of findings under increasingly less ignorable dropout.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

Incremental accrual of information over time is the shared characteristic for both survival analysis and longitudinal studies. Motivated by this important shared concept some authors have tried to adapt the ideas and methods from survival analysis to the longitudinal data setting. The hope is to benefit from similarities and differences in a way that would then contribute to better understanding of the methods and their assumptions in both frameworks. We will focus on the work done by Diggle, Farewell, and Henderson, 2007. In it, authors propose an adaptation of a technique used in survival analysis for making inference on the distribution of a counting process using right censored sample. Identification and interpretation of the parameters that quantify answers to questions of relevance to investigators are rendered possible through the prominent independent censoring assumption. The modeling part of the approach makes use of Aalens additive risk model (Aalen, 1980) adapted for the discontinuous nature of longitudinal data. The adapted version of the additive risk model replaces the continuous time notion of the risk $\lambda(t)$ with the expectation of discrete time increments $\Delta Y_t = Y_t - Y_{t-1}$. This results in a model that is an amalgamated version of transition and random effects model. When such a transfer of concepts is done from a setting in which time (of a jump of the counting process) is random to longitudinal data setting where time of observation is predetermined and not random, some caution is necessary. During this transfer a shift in the interpretation of the parameter that can be estimated from observed data can occur. In a stochastic process setting we have one probabilistic object that is a single counting process whose probabilistic characteristics are governed by jumps that can be realized only within individuals present in the study at that time. Nevertheless probabilistic and sampling characteristics are defined on the level of a single counting process. This risk $\lambda(t)$ has to remain “unblemished” by the act of dropout of any individual. This is reflected by the preservation of its counterfactual interpretation at each time t as a risk that would be there in case no one dropped out. This is achieved by a prominent independent censoring assumption (see Appendix) which serves as an ignorability condition for unbiased estimation/identification of dropout-free, counterfactually interpreted risk of a counting process. Heuristically speaking, randomness (and non-discrete characteristic) of time within this setting makes sure that independent censoring is all we need to preserve such interpretation of the risk in case of dropout. It works like a probabilistic “glue” w.r.t. different individuals, so that a single counting process (more accurately its risk) can be written as a sum of individual specific counting processes (risks) without any measure theoretic ambiguities. Probability of jumps within different individuals happening at the same time is 0 due to the continuous random time.

When transferring these concepts and applying them for longitudinal clinical trial data we have to make sure that the transfer of the identifiability/ignorability assumption is done appropriately so that the marginal interpretation of the parameter, prominently of interest in clinical trials, remains preserved. The most important aspect is in the change between random time of jumps and predetermined observation times in a longitudinal clinical trial. Now, without time as the probabilistic “glue”, our identifiability/ignorability assumption will have to compensate for the part that was covered by randomness of continuous time. As an anchor for defining the appropriate ignorability condition when such transfer of concepts is implemented we will use the relationship between linear increments approach (LI from now on, Diggle, Farewell, and Henderson, 2007) and a method suggested by Robins, Rotnitzky, and Zhao, 1995. Authors called it the extended SWEEP estimator and it was presented together with an accompanying ignorability/identifiability assumption (IIA from now on), under which the identification of a marginal (unconditional, we will use these interchangeably) treatment effect, using only observed data, in the case of dropout is possible. We will describe how this ignorability condition can be expressed in terms of increments and compare the identificational assistance offered by both of these estimators in the case of dropout. Any estimator that offers identificational benefit above and beyond the case of dropout happening completely at random should offer itself (if possible then intuitively) to interpretable sensitivity analysis w.r.t. non-ignorable dropout or equivalently to systematic departure from its IIA. This way the researchers using this estimator can get a sense of robustness of their findings to possible association of the dropout behavior and their outcome of interest not captured by the corresponding IIA.

From the new insight in how LI and extended SWEEP differ and coincide we are able to define a unified approach to sensitivity analysis for violation of ignorability conditions for both of these estimators. From the specific interdependence relationship between them a natural solution for the issue of asynchronous association between outcome process and dropout process becomes apparent. More precisely, we will see how for any sensitivity analysis in a longitudinal setting we have to survey experts on sensitivity parameters non-identifiable from the observed data and at the same time very hard intuitively to conceive. To exemplify this imagine describing the influence of the outcome Y_T at the end of the study on dropout behavior at the second time point. It is usually quite hard to conceive such asynchronous, maybe even counterfactual (outcome at the end of the study might be never observed) association, let alone quantify it accurately. We show how the proposed sensitivity analysis for LI solves this issue under some additional, but plausible conditions.

Sensitivity analysis for LI (we will sometimes use the acronym LISA instead) is a new addition to tools for evaluating robustness of the findings in a longitudinal trial to dropout. We believe that formally positioning it as precisely as possible into the existent landscape of similar tools can only be beneficial in the sense of better understanding LISA, as well as those techniques already in use. We show that LISA can be perceived as an extension of the approach presented originally in Daniels and Hogan, 2008 to longer sequences of observations without necessarily increasing the number of sensitivity parameters. This keeps the sensitivity analysis proposed by Daniels and Hogan interpretable even after some parameter reductions. We offer natural conditions under which this reduction in parameter number does not introduce a significant bias. For 3 time points we express the functional form of the bias from LISA and show how it can be evaluated and its influence curtailed w.r.t. to preserving the usefulness of LISA for longer sequences of observations.

We introduce some notation we will use throughout the paper; in what follows we suppress indices denoting individuals for simplicity, but later incorporate them as they prove important in differentiating between conditions on one individual and conditions on one time interval. Define baseline time as $t = 0$ and assume that at this time we measure a set of baseline covariates $\mathbf{X} = (X_1, \ldots, X_p)$. Assume observations Y_t are taken at $t = 1$ and then at regular, **non-random** intervals up to T. The response at time t is an s-dimensional vector $\mathbf{W}_t = (\mathbf{V}_t^T, Y_t)$ where Y_t, the outcome of interest, is a scalar, and $\mathbf{V}_t$ is a possibly vector valued set of endogenous covariates/outcomes measured at each time in addition to Y_t. Since $\mathbf{V}_t^T$ is not crucial for illustrating and conveying our main message we will leave out for now any contemporaneusly measured, endogenous variables. It will be convenient to define the notation for the history of a process in discrete time:

$\overline{\mathbf{W}}_t = \{\mathbf{X}^T, \mathbf{W}_0^T, \mathbf{W}_1^T, \ldots, \mathbf{W}_t^T\}$. The complementary concept w.r.t. the number of planned observations T is denoted $\underline{\mathbf{W}}_t = \{\mathbf{W}_{t+1}^T, \mathbf{W}_{t+2}^T, \ldots, \mathbf{W}_T^T\}$. Further define $\overline{\mathbf{W}}_0 = \mathbf{X}$ and note that dim$(\overline{\mathbf{W}}_t) = p + st$ where p is the dimension of $\mathbf{X}$. Dropout occurs at any time after $t = 1$ when all subjects are observed. Define $R_t = 1$ if $\mathbf{W}_t$ is observed at time t. We assume that the only missingness is due to dropout, i.e., once a subject leaves the study return is not possible.

In chapter 2 we define the IIA for linear increments estimator and describe its relationship to extended SWEEP estimator. We describe how identification becomes an artifact of specification for the LI and how is this to understand in the light of positivity assumption (see 2). In the same chapter we discuss simulation set up under which cogent comparison of LI and extended SWEEP on one side and the weighted estimator (with inverse probability of observation posing as weights) on the other is possible. After defining IIA for LI in 2, we proceed in 3 to conceptualize a structured way to depart from this IIA for LI. We describe the idea for our approach and showcase methodology involed, after which we validate the approach on a real data set coming from a randomized, longitudinal trial of behavioral economic interventions to reduce CVD risk. In 4 we express sensitivity parameters from LISA defined in 3 as a function of sensitivity parameters from Daniels and Hogan approach and show how these two coincide. Further we evaluate the bias of LISA w.r.t. to Daniels and Hogan approach and argue for its low influence w.r.t. the purpose LISA is meant to serve. For three time points we quantify this bias in the real data set and discuss its minor influence as captured by the possibility of decision discrepancy about robustness of the findings from LISA and Daniels and Hogan approach for same set of sensitivity parameters. We conclude with a short recap and possible future directions in chapter 5. R-code can be found in Apendix C.1 and C.2.

CHAPTER 2

IGNORABLE DROPOUT FOR G-COMPUTATION BASED ESTIMATORS

2.1. Introduction

As already mentioned our starting point will be the work done by Diggle, Farewell and Henderson (Diggle, Farewell, and Henderson, 2007). The dynamic machinery that the authors adapt for continuous longitudinal data in a randomized trial relies mainly on two theoretical concepts: Doob decomposition of a stochastic process (into a predictable compensator process and a zero-mean martingale, see Appendix) and Aalen's additive risk model Aalen, 1980. This method was established under the name linear increments and appeared in several papers (Aalen, 2012, Aalen and Gunnes, 2010) as a choice for analyzing data from longitudinal clinical trials with dropout. Underlying motivation is taken from analogy between the risk of a counting process λ_t at time t and an expected increment $\Delta Y_t = Y_t - Y_{t-1}$.

Longitudinal clinical trials are almost exclusively designed to consistently estimate either the complete time evolution of the **unconditional** treatment effect or the **unconditional** treatment effect on the outcome measured at end the trial. In that regard and by the non-randomness of the prespecified observation times the estimation target is different from capturing exactly the time-specific dynamics of a process. Both of these parameters can be expressed using parameters of the unconditional joint distribution of the outcomes $(Y_1, \ldots, Y_T)$. Dropout or intermittent missingness are almost inevitable in trials where repeated measurements are collected over time. In such settings, if the outcome of interest and reasons for missingness are related, the marginal treatment effect $E[Y_T \mid X = 1] - E[Y_T \mid X = 0]$ (we exclusively use X as treatment indicator), or more generally, the marginal expectation $E[Y_T \mid X = x]$ might not be estimable from the observable data only. An assumption enabling identification is often referred to as an identifiability/ignorability assumption (IIA from now on).

No modeling assumptions are needed for formulating an IIA. In fact, the work that set firm conceptual groundwork on the problem of identifiability (Koopmans and Reiersol, 1950) states:

> *One might regard problems of identifiability as a necessary part of the specification problem. We would consider such a classification acceptable, provided the temptation to specify models in such a way as to produce identifiability of relevant characteristics is resisted. Scientific honesty demands that the specification of a model be based on prior knowledge of the phenomenon studied and possibly on criteria of simplicity, but not on the desire for identifiability of characteristics in which the researcher happens to be interested.*

To avoid such conflation between modeling and identification, IIA should be model-agnostic and define non-parametrically how to express $E[Y_T \mid X = x]$ in terms of observable or known quantities alone. The IIA landscape in the literature is conceptualized primarily through the existence and interrelation between the distribution of complete data $F_C(\Theta_c)$ and distribution $F_O(\Theta_o)$ of observed data. IIA for estimators that adjust for dropout by specifying **only** the model for outcome (within of implicit or explicit parametric G-computation) don't always guarantee existence of $F_C(\Theta_c)$. In particular, the choice of models for expected increments that render increments exchangeable between dropouts and adherers at each scheduled time might not be congenial. In other words, there may not exist such $F_C(\Theta_c)$ within which all these models can be simultaneously true. Note that this issue is related, but different from the notion of correctness of our modeling choices w.r.t. the truth. Congeniality becomes an issue if we parametrize the same part of $F_C(\Theta_c)$ in at least two ways that cannot simultaneously be accommodated by any valid $F_C(\Theta_c)$. For these reasons in the case of such estimators (LI, extended SWEEP) it is necessary to *a priori* define which parameter w.r.t. $F_C(\Theta_c)$ is of interest when formulating IIA. Otherwise, as we will see, the concepts of identifiability and ignorability remain elusive and their definitions prone to logical loops within which we identify what we specify and/or specify what we can identify.

As a platform for illustrating these subtleties we will use the relationship between E-SEQ-MAR (defined below, presented as an unnamed IIA in Robins, Rotnitzky, and Zhao, 1995, from now on RRZ) and discrete time independent censoring (DTIC) (as presented in DFH; Aalen, 2012; Aalen and Gunnes, 2010). For estimation purposes we will use estimators that complement these two IIA's: (non-)linear extended SWEEP estimator (Appendix B of RRZ) and LI. E-SEQ-MAR yields a minimal requirement for using the G-computation formula (Robins, 1986) to identify and estimate

$E[Y_T \,|\, X = x]$. The relationship between E-SEQ-MAR/(non-)linear SWEEP and DTIC/LI exemplifies a subtle difference between censoring, from a stochastic process viewpoint, as a characteristic of a time interval w.r.t. a preceding time interval and the concept of dropout, more characteristic of longitudinal or panel data, as a patient level event. We will see how these two characterizations of missingness coincide under positivity assumptions in longitudinal trials with continuous data.

Another useful analogy for framing these conditions is the concept of justifiable reductions on the sample space as defined in Florens and Mouchart, 1985, where a series of reductions is considered justified if it *does not lose information on the parameters of interest*. This perspective on preserving identifiability of $E[Y_T \mid X = x]$ frames our problem as one of dynamic specification.

In what follows, we will illustrate the relationship between (non-)linear SWEEP and LI w.r.t. estimation of $E[Y_T \mid X = x]$ and/or $E[Y_T, Y_{T-1}, \ldots, Y_1 \mid X = x]$ and explain how congeniality of model choices for increments assumes a role analogous to the one positivity has when it comes to identification by estimators that relay on estimating equations weighted by inverse probability of observing.

2.2. Estimators

In this section we will describe two estimators, extended SWEEP and LI estimator, as representatives of two estimation paradigms: discrete time longitudinal (panel) data and continuous, random time, counting processes paradigm adjusted for discrete time continuous longitudinal data. Definition of these estimators is independent of any possible coarsening process complete data might be subjected to so we will introduce dropout as a form of data coarsening later. For now imagine we are dealing with complete data. (Non-)linear SWEEP and LI can each be presented in two different ways: as strategies based on imputation of observation-level missing data followed by the analysis of the full imputed data, and as standalone estimators. It will prove illustrative to present them in both ways.

2.2.1. Linear increments

Theoretical motivation

A very nice historical perspective of application of martingales in survival analysis is given by Aalen et al. (2009) while a comprehensive and more technical description of the rigorous development of this technique can be found in the monograph by Fleming and Harrington (1991). A basal tool in the analysis of counting (and even more general stochastic) processes is to break them down into separate martingale and drift terms. This is a powerful technique underlying a lot of martingale methods in survival analysis. It is simply described in the case in which $\{Y_t\}_{t=0,1,...}$ is a stochastic process adapted to the discrete-time filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t=0,1,...}, P)$. If Y is integrable, then it is possible to decompose it into the sum of a martingale M and another process A. The process A which starts from zero is such that A_t is $\mathcal{F}_{t-1}$-measurable (predictable) for each $t \geq 1$. Due to M being a martingale we have the identity

$$A_t - A_{t-1} = E\Big[A_t - A_{t-1}|\mathcal{F}_{t-1}\Big] = E\Big[Y_t - Y_{t-1}|\mathcal{F}_{t-1}\Big] \tag{2.1}$$

The first equality follows from the fact that A_t is $\mathcal{F}_{t-1}$-measurable, and the second by the martingale characteristic of the process $\{M_t\}_{t=0,1,...}$.

So, A is uniquely defined by

$$A_t = \sum_{k=1}^{t} E\Big[Y_k - Y_{k-1}|\mathcal{F}_{k-1}\Big] \tag{2.2}$$

and is referred to as the compensator of Y. We remark that this is an abstract existence result and that generally there is no concrete characterization of the process A. This is known as “Doob-Meyer decomposition” and transferring it from its application in continuous time counting process setting to discrete non-random time continuous longitudinal data is the theoretical background for linear increments (LI) estimator. This transfer can be perceived as taking 2 steps at the same time: continuous random time → discrete deterministic time and count outcome → continuous outcome.

We will look at it in detail and try to identify where it could yield some pathological situations w.r.t. the usual discrete non-random time framework within which evaluation of established IIA's for longitudinal data and their corresponding estimators is traditionally facilitated. In particular we will track how to best preserve comparability of the LI method and its assumptions to the already existent methods in the literature.

Modeling increments

Let $\Delta Y_t = Y_t - Y_{t-1}$ and $\Delta Y_1 = Y_1$ by convention. Then a basic building block of LI is a model for the expected increment

$$\boldsymbol{E}\left[\Delta Y_t \mid \overline{\mathbf{Y}}_{t-1}\right] = \omega_t(\overline{\mathbf{Y}}_{t-1}\,; \mathbf{b}_{\Delta Y_t}).$$

DFH stress that the function ω_t, although additive, does not have to be strictly linear (main effects) in history, given one uses random effects or other more complicated structures. It will be useful to state what models ω_t imply about models for corresponding expected outcomes. We draw attention to a self evident fact: by moving Y_{t-1} from $\Delta Y_t = Y_t - Y_{t-1}$ to the right hand side, we imply something about the model for $\boldsymbol{E}\left[Y_t \mid \overline{\mathbf{Y}}_{t-1}\right]$. Thus, by specifying a model for ω_t, no matter how unrestricted it might be in the functional sense, we implicitly impose a constraint on the model for the expected outcome at t: value of Y_{t-1} added to whatever is specified by ω_t. This correspondence between models for increments and models for outcomes is a very simple example of congeniality of model choices. In a simple linear independent effects model for ω_t that includes the whole history except the most recent Y_{t-1} this constraint translates into specifying $\boldsymbol{E}\left[Y_t \mid \overline{\mathbf{Y}}_{t-1}\right]$ by a linear independent effects model where the coefficient that corresponds to Y_{t-1} is held fixed at 1.

The LI estimator of β_{LI_t} for time t is defined as the following sum:

$$\hat{\beta}_{\mathsf{LI}_t} = \sum_{j=1}^{t} \hat{\omega}_j(\overline{\mathbf{Y}}_{j-1}\,;\mathbf{b}_{\Delta Y_j})$$

DFH give some general insight in how to interpret this parameter within the joint distribution of $\overline{\mathbf{W}}_T$ or $\overline{\mathbf{Y}}_T$

... parameters of our dynamic model have a marginal interpretation in the case where only exogenous covariates are used. This interpretation is lost when dynamic covariates are used.

Without further assumptions, as pointed out by DFH, β_{LI_t} cannot be interpreted as $E[Y_t \,|\, X = x]$. As one option, they suggest adapting classical path analysis introduced by Wright, 1921 to marginalize calculated $\hat{\beta}_{\mathsf{LI}_t}$ so that after this process is done we can interpret the resulting statistic as $\hat{E}[Y_t \mid X = x]$. Since we are dealing with non-random visit times this suggested "marginalization" will be feasible only within a set of congenial outcome models implied by models specified for increments (for a similar strategy within the domain of counting processes where time is random see Fosen et al., 2006). As we will see extended SWEEP estimator is an analog of the strategy from Fosen et al., 2006 for longitudinal data coming from a randomized trial.

The other approach for estimation of $E[Y_t \mid X = x]$ using LI was presented in Gunnes et al., 2009: let $\hat{Y}_{i1}$ be the resulting prediction (later, with introduction of dropout, this will be imputation for those missing) for i^{th} individual calculated as $\hat{\omega}_1(\boldsymbol{X}_i\,;\,\hat{\mathbf{b}}_{Y_1})$. Here, depending on the assumed functional form of ω_1 one can use linear or nonlinear least squares as estimation strategy. Define $\hat{Y}_{it}$ as

$$\hat{Y}_{it} = \sum_{j=1}^{t} \hat{\omega}_j(\overline{\hat{\mathbf{Y}}}_{i(j-1)}\,;\,\hat{\mathbf{b}}_{\Delta Y_j})$$

at $t = T, \ldots, 1$. If, for illustration, we assume that all ω_j are linear in history then $\hat{\mathbf{b}}_{\Delta Y_j}$ is an estimate from the linear regression of $\Delta\hat{Y}_j = \hat{Y}_j - \hat{Y}_{j-1}$ on $\overline{\hat{\boldsymbol{Y}}}_{(j-1)}$ where $\overline{\hat{\boldsymbol{Y}}}_{(j-1)}$ is the history of predictions (imputations) until and including time $j-1$. Then

$$\hat{E}(Y_t) = \frac{1}{n}\sum_{i=1}^{n}\hat{Y}_{it}$$

is interpreted in Gunnes et al., 2009 as the consistent estimate of the marginal mean $E\left[Y_t \mid X = x\right]$. This would mean that our interpretability issue involving β_{LI_t} from DFH has been solved by the decision to use an imputation implementation of LI. Such a “shortcut” for implementing formal path analysis via predictions/imputations, as we will see, hides assumptions that become even more important when dropout is introduced. Namely, correct interpretation hinges upon a specific characteristic of the sequence of models one specifies for ω_j’s in order to make observed and missing increments exchangeable at each time w.r.t. their mean. With such an approach it is then hard to decouple specification from identifying assumptions and conflation of identification and specification becomes unavoidable.

Implied data generating mechanism

After laying out the theoretical background, DFH illustrate what such a strategy implies about the underlying true data generating process (which we present in its original form from DFH).

$$Y_t = \sum_{j=1}^{t} E\Big[\Delta Y_j | \overline{\mathbf{Y}}_{j-1}\Big] + M_t + \varepsilon_t \tag{2.3}$$

$$\begin{pmatrix}\text{measured}\\ \text{response}\end{pmatrix} = \begin{pmatrix}\text{predictable}\\ \text{compensator}\end{pmatrix} + \begin{pmatrix}\text{zero mean}\\ \text{martingale}\\ \text{random effect}\end{pmatrix} + \begin{pmatrix}\text{measurement}\\ \text{error}\end{pmatrix}$$

We won't comment on this structure further except to contrast it to the approach where overdispersion in the data is captured by a classical Laird-Ware (Laird and Ware, 1982) random effect consisting of random intercept and possibly a random slope. In such a setting we assume a joint Gaussian distribution for latent intercept and slope. Further, none of the outcome models includes past outcomes or contemporaneous endogenous covariates so it remains reasonable to assume that the conditional (on $\mathbf{X}$) and marginal distribution of the random effect is the same. This allows us to specify how an estimate conditional on random effect can be marginalized over the distribution of the intercept and the slope to calculate its unconditional estimate. Compared to that, only a martingale assumption on a random effect at each time is too little to clearly define how and over which distribution to marginalize in the sense previously mentioned. The situation is further complicated by the fact that at each time we include most recent past in some form, which is inevitably associated with and in a way defines the next martingale random effect.

2.2.2. Extended SWEEP

(Non-)Linear extended SWEEP for $E\,[Y_t \mid X = 1, 0]$

Extended SWEEP estimator was introduced in RRZ as a generalization of a computation technique (Roderick, Little, and Rubin, 1986) for calculating MLE in the case when normally distributed repeated measurements are coarsened by MAR dropout. We first introduce its general form. We specify for each individual $i\;t$ (non-)linear models $\eta_{i\,t'\,t} \equiv \eta_{t'\,t}(\overline{\mathbf{Y}}_{t'-1}\,;\theta_{t'\,t})$ for $t^{'} = t+1, \ldots, 1$ where we choose the functional form of $\eta_{t'\,t}$ according to domain experts' knowledge about the association of outcome and dropout. Use $\hat{\eta}_{i\;1t}$ as an estimate of $E\,[Y_{it} \mid \mathbf{X_i}]$ (remember the convention $\overline{\mathbf{W}}_0 = \mathbf{X}$) where $\hat{\eta}_{i\,t'\,t}$ is recursively defined as

1) $\hat{\eta}_{i\;t+1,t} = Y_{it}$

2) $\hat{\eta}_{i\,t',t}$ is defined only for those individuals with $R_{i\,t'-1} = 1$ as the predicted value from a (non)linear least squares regression of $\hat{\eta}_{i\,t'+1,t}$ on $\overline{\mathbf{Y}}_{t'}$ among those subjects j with $R_{j\,t'} = 1$ according to specified (non-)linear function $\eta_{t'\,t}(\cdot\,;\theta_{t'\,t})$

Then, as already stated, by standard least squares theory if **all** $\eta_{t'\,t}(\cdot\,;\theta_{t'\,t})$'s **are correctly** specified we can use $\hat{\eta}_{i\;1t}$ as an unbiased asymptotically normal estimator of $E\,[Y_{it} \mid X_i = x]$. Another way to use the predictions from this series of models is

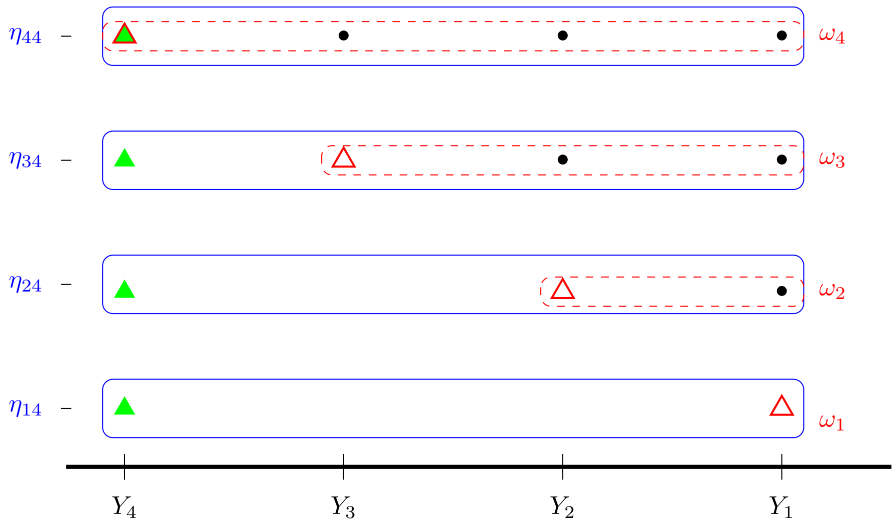


Figure 2.1: Different parts of the joint distribution of $\overline{\mathbf{Y_4}}$ parametrized byη' and ω's

$$\hat{E}(Y_t) = \frac{1}{n}\sum_{i=1}^{n} \hat{\eta}_{i\;1t}$$

which is more analogous to imputation oriented use of LI. Notice the “relaxation” of the (observed) history w.r.t. the time point t at which we are interested to estimate the marginal mean and how in the case of LI we don’t specify such “lagged”, but only contemporaneous association. See Figure 2.1 or illustration of different ways η’s and ω’s constrain the joint distribution for T=4.

Linear extended SWEEP for $E\left[\overline{\mathbf{Y}}_t \mid X = 1, 0\right]$ and congeniality of η’s

If we try to use the above approach to estimate $E\left[\overline{\mathbf{Y}}_T \mid X = x\right]$, the expectation over the whole study period, we have to acknowledge the possible issue of lack of congeniality (see Section 2.3.1) for assumed non-linear models. The above non-linear SWEEP approach amounts to $T(T-1)/2$ specified (believed to be true) non-linear models $\eta_{t'\,t}$ for $1 \le t \le T$ and $1 \le t^{'} \le t+1$ ($\eta_{t+1,t}$ and $\eta_{1,t}$ is degenerate, implied form respectively) . Then for a pair (Y_k, Y_j) and $m < \min(k,j)$, the assumed functional forms $\eta_{mk}(\cdot;\boldsymbol{\theta}_{mk})$ and $\mu_{mj}(\cdot;\boldsymbol{\theta}_{mj})$ might not be simultaneously possible within any joint distribution of $\overline{\mathbf{Y}}_T$. These issues were originally warned against in RRZ as well for the case where non-linear SWEEP is used to estimate $E\left[\overline{\mathbf{Y}}_t \mid X = 1, 0\right]$. This also implies that the interpretation of $\hat{\eta}_{i\,1k}$ **and** $\hat{\eta}_{i\,1j}$ simultaneously as marginal expectations within some joint distribution of $\overline{\mathbf{Y}}_{iT}$ dominated by a valid probability measure is not possible. We say, following Florens and Mouchart, 1985, that the collection of **sequential** reductions on the sample space of observed data in the form of $\eta_{t'k}$ and $\mu_{t''j}$ for $t^{'} = k+1, \dots, 1$ and $t^{''} = j+1, \dots, 1$ is not justifiable w.r.t. identification of $E\left[Y_{ik}, Y_{ij} \mid X = 1, 0\right]$.

If we choose $\eta_{tt}(\cdot\,; \theta_{tt})$'s for $t \in T, \dots, 1$ above to be linear we are able to write an estimate of the extended SWEEP as a series of matrix multiplications. (We will assume, in what follows, that we are in a more general setting where $\mathbf{V}_t \neq \emptyset$. For the imputation based representation of the extended SWEEP we kept $\mathbf{V}_t = \emptyset$ for the sake of notational simplicity. It is not hard to do a mental experiment where η's specify not only a model for a conditional expectation of a specific Y_t but for a vector $\mathbf{W}_t$. Then η's become s-dimensional vectors of functions.) Let $\hat{\theta}_{t\,t} = (\hat{\theta}_{Y_t}, \hat{\theta}_{t1}, \dots, \hat{\theta}_{t(s-1)})^T)$ to be the least squares estimate from the multivariate linear regression of $\boldsymbol{W}_{it}$ on $\overline{\boldsymbol{W}}_{i(t-1)}$ Dimension of $\hat{\theta}_{t\,t}$ is $s \times (p + s(t-1))$. Let further $\hat{B}_t$ denote the $(p+st) \times (p + s(t-1))$-dimensional matrix $(I_{p+s(t-1)}, \hat{\theta}_{t\,t}^T)^T$. The extended SWEEP estimator of $E(\overline{\boldsymbol{W}}_{iT} \mid \mathbf{X})$ is equal to

$$\hat{E}(\overline{\boldsymbol{W}}_{iT} \mid \mathbf{X}) = \mathbf{X}\,\hat{\boldsymbol{\theta}}_{\textsf{sw}}$$

where $\hat{\boldsymbol{\theta}}_{\textsf{sw}} = \hat{B}_T \times \hat{B}_{T-1} \times \cdots \times \hat{B}_2 \times \hat{B}_1$ which is a $(p+sT) \times p$ dimensional matrix. Out of $(p+sT)$ rows of this matrix p correspond to baseline covariates and have a degenerative distribution (coefficients will all be 1), T rows correspond to the outcome of interest $\overline{\mathbf{Y}}_T$ while $(s-1) \times T$ are for auxiliary (possibly endogenous) covariates $\mathbf{V}$. Congeniality of $T(T-1)/2$ implicitly defined η models within this linear implementation of the extended SWEEP can still be discussed. Within a linear class of models, η_{tt}'s for $t \in T, \dots, 1$ will imply numerical values for the coefficients of all of the remaining $T(T-1)/2 - T = T(T-3)/2$ η models.

We want to point out that although extended SWEEP does not model probability of dropout it has a built in mechanism for recovering a marginal treatment effect, or more general, marginal mean of $E(\overline{\boldsymbol{Y}}_{iT} \mid \mathbf{X})$. This is, at the same time its parameter of interest. Through the above defined series of matrix multiplications it is possible to recover marginal treatment effect when the series of conditional mean models $\eta_{tt}(\cdot\,; \theta_{tt})$'s for $t \in T, \ldots, 1$ is linear w.r.t. history. This way one compactly implements a general concept of path analysis (Wright, 1921), precursor of structural equation modeling. "Padding" the coefficient matrix $\hat{\theta}_{t\,t}^{T}$ with the identity matrix $I_{p+s(t-1)}$ is done with the purpose of making this matrix multiplication mathematically defined while its statistical characteristics remain unchanged.

We can of course always use any single markov assumptions we might believe to be true when it comes to estimating $\hat{\theta}_{t\,t} = (\hat{\theta}_{Y_t}, \hat{\theta}_{t1}, \ldots, \hat{\theta}_{t(s-1)})^T)$. We would "code" any markov assumption in $\hat{B}_j$'s matrices by setting the part of $\hat{\theta}_{t\,t}$ that a particular markov assumption annuls to zero.

Congeniality remains to be an issue even if we decide to model increments instead of outcomes. Since LI is inherently incremental technique it is not possible to estimate β_{LI_j} without acknowledging necessity for all of the previous β_{LI_k} $(k < j)$. ω_t's specified for LI introduce a set of analogous sequential reductions on the sample space as well. Heuristically, for a pair (Y_k, Y_j) and $m < \min(k, j)$, a single ω_m has to be correct w.r.t. to $E\left[Y_{ik} \mid X = x\right]$, $E\left[Y_{ij} \mid X = x\right]$ **and** $E\left[Y_{ik}, Y_{ij} \mid X = x\right]$ if we want to interpret β_{LI_j} (w.l.o.g. $k < j$) as $E\left[Y_{ij} \mid X = x\right]$. All of these congeniality considerations are defined w.r.t. $F_O(\Theta_o)$.

To end this section we note an important difference between non-linear SWEEP and LI. The above approach for non-linear SWEEP can be used to identify and estimate a single $E\left[Y_{it} \mid X = x\right]$ without any issues related to lack of congeniality of the chosen models. On the other hand, the LI estimating approach, due to its incremental, inherently conditional paradigm, implicitly specifies $t(t-1)/2$ models by deciding on a model for an expected increment at each $m < t$. In other words, writing Y_t as a sum of increments $\Delta Y_1 + \Delta Y_2 + \cdots + \Delta Y_t$ hard-wires the estimation of $\boldsymbol{E}\left[\,\overline{\mathbf{Y}}_t \mid \mathbf{X}\right]$ as the only option, and it is not possible to decouple and concentrate only on one outcome without (implicitly) specifying models for all those that are measured before it. This way we are making identification and specification of the parameter more dependent than it is necessary when using non-linear SWEEP to estimate $E\left[Y_{it} \mid X = x\right]$ in case of non-ignorable dropout.

To reiterate, in the case in which we choose all $\eta_{t^{\prime}t}$'s to be **linear** and **autoregressive** in observed history there is no need to differentiate between $E\left[Y_t \mid X = x\right]$ and $E\left[\,\overline{\mathbf{Y}}_t \mid X = x\right]$ as parameters of interest. That is, t linear models $\eta_{t^{\prime}t}(\cdot\,;\theta_{t^{\prime}t})$ for $t^{'} = t, \ldots, 1$ implicitly specify all $t(t-1)/2$ models that include those $\eta_{\cdot j}$ for $j \in t-1, \ldots, 1$.

2.3. Extended SWEEP and LI as strategies for ignorable dropout

2.3.1. Congeniality, specification, identification and ignorability for extended SWEEP and LI

Ignorability and identification can, when left too vague, lead to non-trivial caveats when it comes to evaluating one estimator w.r.t. the other in presence of missing data. The modeling assumptions implied by ω's and/or η's about different (conditional) moments within the joint distribution of longitudinal data $\overline{Y}_T$ are made independently and dynamically. These modeling assumptions can't be ordered and combined in a way that allows us to conclude what set of joint distributions can they be accommodated by. There is no structured way of doing this, but, what we in some cases can do is recognize when are these assumptions in the form of dynamic specification not congenial among each other. In general the more assumptions we make, the easier it gets to recognize that no joint distribution is able to accommodate them. In other words, we can sometimes have a good enough idea about if they can simultaneously be true in **any** valid joint distribution of the complete data $(\overline{Y}_T, \overline{R}_T)$ including the dropout indicators. This is captured by the concept of congeniality of these parametric assumptions. A prominent example (we give it without proof for illustration purposes only) of a set of uncongenial constraints is : $(Y_1, Y_2) \sim N((\mu_1, \mu 2)^T, \Sigma)$ and

$$P(R_2 = 0 \mid Y_1, Y_2) = \text{expit}\{\alpha + \gamma_1 Y_1 + \gamma_2 Y_2\}$$

$$P(R_2 = 0 \mid Y_1) = \text{expit}\{\alpha^{'} + \gamma_1^{'} Y_1\}$$

Assuming that the complete data distribution is bivariate normal and that $P(R_2 = 0 \mid Y_1, Y_2)$ has the presented logistic form preserving this form for $P(R_2 = 0 \mid Y_1)$ when we "marginalize" $P(R_2 = 0 \mid Y_1, Y_2)$ w.r.t. $P(Y_2 \mid Y_1)$ is not possible.

We are not offering any algorithmic solution for establishing or disproving congeniality (it is not a trivial task), but merely pointing it out as something that one should be aware of. Precise effect of congeniality on identification and ignorability becomes relevant when we compare one estimation strategy to another w.r.t. their ability to adjust for same modality of dropout. At that moment we should either render congeniality irrelevant for both strategies or allow it to have same consequences for both. This, when it is possible, is also necessary because lack of congeniality has a "confounding" effect on a relationship between specification and identification in the framework of missing data (or any framework where one needs to make untestable assumptions). Related to that we give a quote from the work that set firm conceptual groundwork on the problem of identifiability (Koopmans and Reiersol, 1950):

> *One might regard problems of identifiability as a necessary part of the specification problem. We would consider such a classification acceptable, provided the temptation to specify models in such a way as to produce identifiability of relevant characteristics is resisted. Scientific honesty demands that the specification of a model be based on prior knowledge of the phenomenon studied and possibly on criteria of simplicity, but not on the desire for identifiability of characteristics in which the researcher happens to be interested.*

This is in alignment with our notion of "confounding" effect of the lack of congeniality on the relationship between identification and specification. Without considerations what lack of congeniality means for our estimation strategy w.r.t. specification and identification, ignorability and identification can become a concept meaningful only within the set of semi-parametric assumptions entailed in functional forms of η's and/or ω's and lose it's meaning within some/any valid underlying joint distribution of complete data. Specifying what is the target of estimation w.r.t. complete (counterfactual) data distribution is a good first step in detecting lack of congeniality.

So for a clear definition of ignorability and identification it is imperative to **specify** some question of interest we are trying to answer using our data, before data collection and coarsening occurs. In particular, one should be able to map this question of interest onto a parameter of the true joint distribution of $\overline{\mathbf{Y}}_t$. Specification therefore secures the unique domain within which ignorability and identification preserve unambiguous meaning. In the setting for which the use of LI and extended SWEEP is suggested, namely longitudinal data coming from a traditional clinical trial, one is mostly interested in marginal treatment effect on the outcome at each, or often, at the last of the scheduled measurement times. (Non-)linear extended SWEEP as originally introduced does preserve this as its parameter of interest, while LI's parameter of interest is not clearly defined and thus becomes a result of collection of models ω we choose for each increment. We will always achieve unbiased estimation of β_{LI_t} in this case, only the meaning of this parameter within any possible joint distribution of $\overline{\mathbf{Y}}_T$ will depend on congeniality of our modeling choices. In the next section we specify IIA under which β_{LI_t} can be interpreted as a marginal mean.

2.3.2. Identifiability and ignorability assumptions (IIA)

Without imposing any additional regularity conditions (see *measurable separability* in Florens, Mouchart, and Rolin, 1993 or condition (T3) in Eichler and Didelez, 2010) , the relationship between $F_O(\Theta_o)$ and $F_C(\Theta_c)$ hinges only on a set of constraints on conditional expectations, implied by $\eta_{t^{'}t}$ $1 \leq t \leq T$ and $1 \leq t^{'} \leq t+1$ (and/or ω_t's $1 \leq t \leq T$). Both E-SEQ-MAR and DTIC are constraints analogously relating observed (w.r.t. $F_O(\Theta_o)$) and marginal (w.r.t. $F_C(\Theta_c)$) moments.

E-SEQ-MAR for identification of $E[Y_T \mid X = x]$ is formulated as

$$E[Y_{i,\,T} \mid \overline{\mathbf{Y}}_{i,\,j-1}, R_{i,\,j} = 1] = E[Y_{i,\,T} \mid \overline{\mathbf{Y}}_{i,\,j-1}, R_{i,\,j-1} = 1], \;\; 2 \leq j \leq T. \tag{2.4}$$

Index j relaxes the history so that at each time t the conditional expectation $E[Y_{i,\,T}|\overline{\mathbf{Y}}_{i,\,j-1}, R_{i,\,j-1} = 1]$ is equated with its observed component

$E[Y_{i,\,T} \mid \overline{\mathbf{Y}}_{i,\,j-1}, R_{i,\,j} = 1]$. Notice that $\eta_{t^{'}T}(\cdot\,; \theta_{t^{'}T})$, $1 \leq t^{'} \leq T+1$ is the functional form of the left hand side of (2.4). We then use (2.4) as a justification for the imputation algorithm in (2.2.2). DTIC, as pointed out in Aalen and Gunnes, 2010, is, in its original form, an assumption about missingness of responses and not about censoring of the individual. This is a remnant of its motivation within theory of counting processes where differentiating between probability measure for a time interval and a probability measure for an individual cannot result in any ambiguous behavior w.r.t. null sets. In a longitudinal clinical trial observation times are not random any more and we will see how this reflects on DTIC when we write it out using notation accommodating discrete non-random observation time. For DTIC in longitudinal clinical trial we write

$$E[\Delta Y_t \mid \overline{\mathbf{Y}}_{t-1}, R_t = 1] = E[\Delta Y_t \mid \overline{\mathbf{Y}}_{t-1}, R_t = 0, R_{t-1} = 1], 2 \leq t \leq T \tag{2.5}$$

We can see that the conditioning (and exchangeability of mean increments) in DTIC is specified only w.r.t. to the most immediate past, while the “relaxation” of this “gap” as present in (2.4) is not part of this IIA. Notice that (2.5) consists of the all the “first” constraints that (2.4) includes for different $t \in 2, \ldots, T$. This is naturally not enough to imply (2.4) in general for any $t > 2$. Thus, transitioning from random observation time to pre-scheduled one makes a more explicit specification of DTIC necessary for identification of $E(Y_T|X)$. Loosely speaking, there are parts of DTIC necessary for identification of $E(Y_T|X)$ that were “covered” implicitly in process notation by randomness of observation times. For now we express those parts, that were before implicit in randomness of observation time, explicitly and the version of DTIC that implies (2.4) is

$$\begin{aligned} E[\Delta Y_{i,\,k} \mid \overline{W}_j, R_j = 1] &= E[\Delta Y_{i,\,k} \mid \overline{\mathbf{W}}_j, R_{j-1} = 1], \\ & j, k \quad 1 \leq j \leq k, \;\; 1 < k \leq t \end{aligned} \tag{2.6}$$

We show (see Appendix B.1) how within a restricted class of linear autoregressive models for ω_t's equivalence of (2.5) and (2.4) depends on positivity assumption (for positivity see Appendix A and Laan and Rose, 2011) in the data. Heuristically, positivity ensures that every individual that contributes to ignorability of dropout w.r.t. two consecutive time intervals, contributes also to the same extent to making dropout ignorable across individuals. In other words, adding up “ignorability preserving contributions" across individuals and across intervals is exchangeable under positivity. If positivity does not hold for each $t \leq T$ then a) it is still possible to interpret $\hat{\beta}_{\mathsf{LI}_t}$ as a marginal parameter of the joint distribution as long as the $T(T-1)$ model implicit in ω_t's are congenial (with non-linear models it is not clear how congeniality is defined) and b) identification is facilitated exclusively through model choices we make for η's and/or ω's. Notice that a) and b) are opposite goals, and the art is to achieve balance between them, while still making plausible assumptions w.r.t. the association between dropout and the outcome of interest. This means that, without including positivity assumption, unambiguous characterization of the relationship between LI and non-linear SWEEP estimator w.r.t. identification of $E\left[Y_{iT} \mid X_i = x\right]$ is not possible.

2.4. Simulations

2.4.1. Set up

We will define our simulation scenario by accentuating how it differs from the one presented in DFH. Equation (2.4) depicts coarsely the full data generation process, which follows closely that in DFH. The precise structure for one individual and one time point is

$$
\begin{aligned}
Y_t^M &= \mu_t + M_t + \varepsilon_t \\
Y_t^S &= \mu_t + S_t + \varepsilon_t
\end{aligned}
$$

for $t \in (0, 1, 2, 4, 6, 8)$. We use $\mu_1 = \cdots = \mu_6 = 0$ and

$$
\begin{aligned}
M_t &= U_1 + \cdots + U_t \\
S_t &= U_1 + t \times U_2
\end{aligned}
$$

where $U_i \sim N(0, \sigma_i^2)$. Thus M_t is a zero-mean random walk (martingale) while S_t is a Laird-Ware random effects model with intercept and slope. The σ_i^2's are chosen as in DFH to facilitate comparison. We show results for the sixth observation time point, which corresponds to the sixth column of Table 1 in DFH.

LI and extended SWEEP are evaluated for two choices of linear independent effects models w.r.t: full history and intercept only. Further on, we evaluate a hybrid strategy consisting of prediction/imputation of the whole data set by $\hat{Y}_{ij}$ using models $\omega_k(\cdot\ ; \mathbf{b}_{\Delta Y_k}),\ 2 \leq j \leq 6, 1 \leq k \leq j$ followed by applying linear SWEEP with $\eta_{t'6}(\cdot\ ; \theta_{t'6})$ for $1 \leq t^{'} \leq 6$. We do this for each combination of full history and intercept only model. In addition we show IPW EE estimator from RRZ with true and incorrectly modeled probability of dropout. We assume intercept to be the only baseline covariate $\mathbf{X}$ (we observe and compare the estimate of the mean outcome Y_6 only in one treatment group) and no auxiliary contemporaneous covariates (outcomes) $\mathbf{V}_t^T$ are measured.

For each choice of the random effect (martingale or Laird-Ware) we introduce four types of dropout:

1. (SEQ)-MAR / no positivity
2. NMAR / no positivity
3. (SEQ)-MAR / positivity
4. NMAR / positivity

The second dropout model is the same as described in DFH. We generate dropout at time t based on the logistic model that includes only the latent random effect at $t-1$. No observations are missing at baseline, so Y_1 is completely observed. This, in general, results in NMAR dropout.

$$\begin{aligned} P(R_t = 0 \mid R_{t-1} = 1, \overline{\mathbf{Y}}_{t-1}, \overline{\mathbf{M}}_{t-1}) &= P(R_t = 0 \mid R_{t-1} = 1, M_{t-1}) \\ &= \text{expit}\{\alpha_{t-1} + \gamma_{t-1} M_{t-1}\} \end{aligned} \tag{2.7}$$

$$
\begin{aligned}
P(R_t = 0 \mid R_{t-1} = 1, \overline{\mathbf{Y}}_{t-1}, \overline{\mathbf{S}}_{t-1}) &= P(R_t = 0 \mid R_{t-1} = 1, S_{t-1}) \\
&= \text{expit}\{\alpha_{t-1} + \gamma_{t-1} S_{t-1}\} \qquad (2.8)
\end{aligned}
$$

for $t \in 1, 2, 4, 6, 8$. The first dropout model differs only from the second in that the predictor is not the previous random effect but previous outcome Y_{t-1}, which makes it SEQ-MAR dropout as defined in RRZ.

For this dropout model we used

$$
\begin{aligned}
(\alpha_0, \alpha_1, \alpha_2, \alpha_4, \alpha_6) &= (-8, -6, -6, -6, -4) \\
(\gamma_0, \gamma_1, \gamma_2, \gamma_6, \gamma_8) &= (0.2, 0.3, 0.3, 0.5, 0.6)
\end{aligned}
$$

To evaluate estimators under the assumption of positivity we introduce two additional modes of dropout. These are positivity preserving versions of (2.7) and (2.8) and their positivity preserving MAR counterparts. At each time point probability of dropping out is truncated at 0.3 so that the minimum probability of being observed at $t = 6$ is approximately $0.7^5 = 0.17$.

$$
P(R_t = 0 \mid R_{t-1} = 1, \overline{\mathbf{Y}}_{t-1}, \overline{\mathbf{M}}_{t-1}) = 0.3 \times \Big[\text{expit}\{\alpha_{(p,\,t-1)} + \gamma_{t-1} M_{t-1}\}\Big]
$$

$$
P(R_t = 0 \mid R_{t-1} = 1, \overline{\mathbf{Y}}_{t-1}, \overline{\mathbf{S}}_{t-1}) = 0.3 \times \Big[\text{expit}\{\alpha_{(p,\,t-1)} + \gamma_{t-1} S_{t-1}\}\Big].
$$

The $\alpha_{(p,\,t-1)}$ are chosen such that the proportions of the missing observations at each time $t = 2, 3, 4, 5, 6$ are equal to the proportions in the original set-up (2.7). These are ($\approx$ 1%, $\approx$ 11%, $\approx$ 16%, $\approx$ 30%, $\approx$ 50%) for $t = 2, 3, 4, 5, 6$.

We generate full data with sample size $n = 500$ and results correspond to Monte Carlo (MC) means and standard deviations of 1000 iterations.

2.4.2. Results

The first four columns of Table 2.1 show results for all models when the positivity assumption is upheld. π denotes an average of the minimum over 1000 iterations of $\big(\Pi_{j=2}^{6} P(R_{i,\,j} = 1 \mid R_{i,\,j-1} = 1, \overline{\mathbf{Y}}_{i,\,j-1})\big)$. E-SEQ-MAR and DTIC in this case have the same contribution w.r.t. identification of the marginal mean of Y_6. We can see that given a correct collection of models $\eta_{t'6}(\cdot\,; \theta_{t'6})$ for $1 \leq t^{'} \leq 6$ extended SWEEP estimator will be unbiased for $E[Y_{i,\,6}^{M} \mid X]$ and $E[Y_{i,\,6}^{S} \mid X]$. For SEQ-MAR dropout, the correct choice of $\eta_{t'6}$'s are linear independent effects in complete history. The same choice of ω_t's for $t = 1, \ldots, 6$ will yield unbiased estimates for both types of random effect structure. When dropout is NMAR the correct $\eta_{t'6}$'s might no longer be linear in observed history for data generated using Laird-Ware latent effect so using such models will not yield an unbiased estimate of $E[Y_{i,\,6}^{S} \mid X]$. Correct choice of $\eta_{t'6}$'s in the case of martingale random effect is a constrained version of a linear model where the coefficient next to the most recent outcome is held fixed at 1 while others are fixed at 0 with the intercept remaining to estimate from the data. For such a choice we can estimate $E[Y_{i,\,6}^{M} \mid X]$ unbiasedly. We show the equivalent LI estimator (row 5) that will be unbiased for $E[Y_{i,\,6}^{M} \mid X]$ if ω_t's for $t = 1, \ldots, 6$ are all chosen to be intercept only models. This correct specification comes only due to our exogenous knowledge of a correct functional form (of outcomes and/or increments) that will make dropout ignorable. So in this case we have an MNAR but still ignorable dropout. In these columns the IPW EE estimator that uses true probability of dropout in the weights (row 9) is biaswise comparable to correct choice of ω_t's. In the case of LI though, we need to keep in mind that congeniality of $\eta_{t'\,t}$'s ($1 \leq t \leq 6$ and $1 \leq t^{'} \leq t$) implied by these "intercept only" choices for ω_t's still remains an issue to consider.

In columns 5 to 8 we have a near loss of positivity and congeniality of $\eta_{t'\,t}$'s ($1 \leq t \leq 6$ and $1 \leq t^{'} \leq t$) becomes the only criteria under which we can claim identification, and, for correct choice of models, an unbiased estimation of the unconditional mean. $F_C(\Theta_c)$ without positivity condition cannot be unambiguously defined only from $F_O(\Theta_o)$ and its mere theoretical existence is in question. Identification here becomes even on a semantical level an artifact of modeling choices for $\eta_{t'\,t}$'s or ω_t's ($1 \leq t \leq 6$). Under such conditions we risk to conflate identification and specification to the unpredictable extent. Again, we have correct linear choices for $\eta_{t'6}$'s and ω_t's if dropout is SEQ-MAR. In contrast to the set up with positivity, the IPW estimator exhibits very large standard errors which is

Table 2.1: MC mean and standard deviation for 1000 iterations for sample size 500

		SEQ-MAR		*NMAR*		*SEQ MAR*		*NMAR*	
		$\pi \approx 0.18$	$\pi \approx 0.18$	$\pi \approx 0.18$	$\pi \approx 0.18$	$\pi \approx 0.07$	$\pi \approx 0.07$	$\pi \approx 0.07$	$\pi \approx 0.07$
Est.		Y_M	Y_S	Y_M	Y_S	Y_M	Y_S	Y_M	Y_S
1 LI-fh	Mean	-0.06	0.00	-1.52	-2.87	0.04	-0.02	-6.09	-8.30
	SD	1.98	1.022	1.99	1.84	2.97	2.30	2.01	1.54
2 SWLI-fh,fh	Mean	-0.02	0.06	-0.38	-0.56	0.06	-0.05	-3.59	-3.33
	SD	1.68	1.28	1.63	1.28	2.60	1.65	2.99	1.34
3 SWLI-c,fh	Mean	-0.00	0.05	-0.20	-0.82	0.06	-0.03	-2.70	-3.44
	SD	1.72	1.31	1.64	1.29	2.66	1.66	1.97	1.29
4 SW-fh	Mean	-0.02	0.06	-0.38	-0.56	0.06	-0.05	-3.59	-3.33
	SD	1.69	1.28	1.63	1.28	2.60	1.66	1.99	1.34
5 LI-c	Mean	2.40	-1.30	0.03	-4.10	6.17	-2.08	0.01	-7.73
	SD	1.90	1.80	1.90	1.79	2.20	1.80	1.80	1.46
6 SWLI-fh,c	Mean	0.60	-1.53	0.05	-2.00	1.95	-2.84	-0.01	-4.38
	SD	1.58	1.22	1.54	1.19	1.68	1.17	1.50	1.11
7 SWLI-c,c	Mean	0.40	-1.46	0.05	-2.10	1.70	-2.69	-0.02	-4.21
	SD	1.62	1.27	1.56	1.21	1.69	1.20	1.55	1.14
8 SW-c	Mean	-8.55	-12.22	-9.00	-12.95	-20.00	-24.10	-19.42	-22.29
	SD	2.38	2.46	2.40	2.40	2.08	1.81	1.80	1.65
9 IPW	Mean	-0.08	-0.08	-0.06	-0.09	-14.50	-17.64	-14.22	-16.35
	SD	2.30	2.48	2.33	2.53	6.85	5.83	5.16	5.78
10 IPW-Y	Mean	6.95	12.32	1.00	2.22	-14.87	-18.1	-12.78	-15.24
	SD	9.15	13.96	4.51	5.60	5.63	5.00	3.13	2.03

Note : Est = estimator; fh-full history; c = only intercept;
SW = ext. SWEEP; SWLI-fh,c = impute missing using only intercept
model for LI then use SWEEP with fh;

a known drawback of this estimator. What is somewhat surprising is that even when we use true probabilities for SEQ-MAR dropout, IPW EE still highly underestimates the marginal mean. With the sample size of 500 that would in practice correspond to a large clinical trial observing such high finite sample bias should be an aspect of IPW EE to consider. When comparing IPW EE to other estimators a special attention should be given not only to inflated standard errors, but also a finite sample bias coming from near loss of the positivity in the simulated sample. On the other side, we can see that with the correct value for weights and a positivity preserving sample (column 3) IPW EE can serve as an alternative to the LI "intercept only" series of models. In column 7 this "intercept only" choice for ω_t's result in unbiased estimation of β_{LI_6}. Interpreting this parameter as $E[Y_{i,\,6}^{M} \mid X]$ in this case hinges on congeniality of the models chosen, since identification becomes inseparable from specification. Figure 2.2 shows the relationship between specification and identification in the light of (observed) positivity, congeniality and IIA for LI and extended SWEEP.

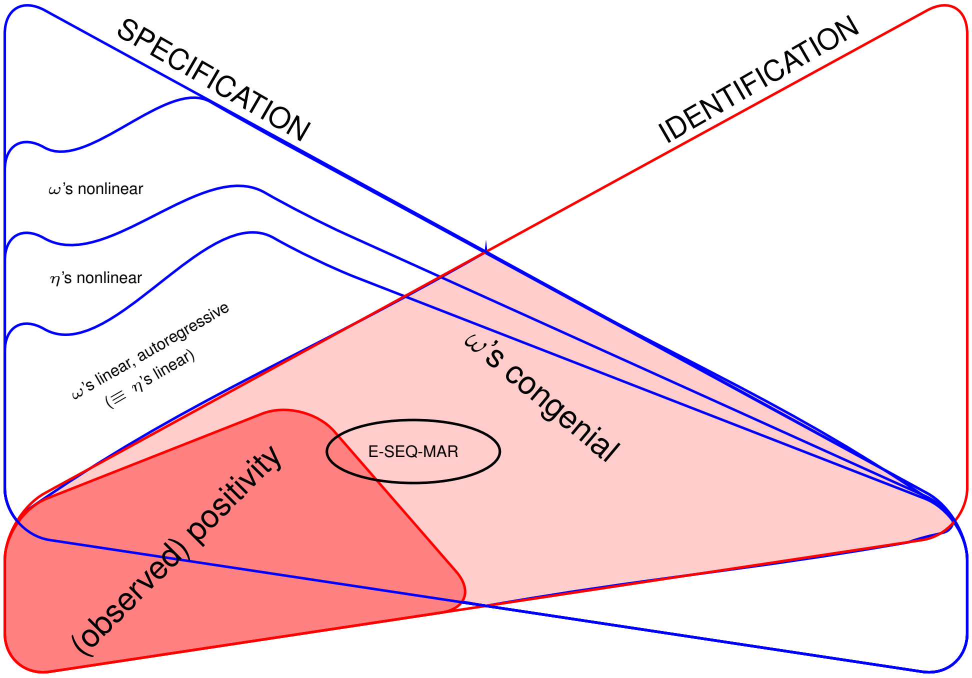


Figure 2.2: Identification by specification of the outcome model ((observed) positivity, congeniality and IIA for LI and extended SWEEP

2.5. Discussion

LI was proposed as a method for accurate description of time evolution of longitudinal data when faced with dropout. For this purpose it is a very easily implemented and powerful technique. DTIC as presented in DFH and expressed as (2.5) in notation characteristic for non-random observation times is enough to achieve this goal. We showed that under E-SEQ-MAR, LI can be used for a more ambitious task of identifying the unconditional mean or the unconditional treatment effect in longitudinal clinical trial with dropout. Difference between these two applications of LI are on some level analogous to difference between Pearls's intervention based causality (Pearl, 2009) and Granger's causality (Granger, 1969). Our goal is comparable to the one authors in Eichler and Didelez, 2010 had. There, conditions on a multivariate time series are specified that are needed in addition to Granger's non-causality (Granger, 1969, Florens and Mouchart, 1982), to identify average causal effect (ACE) of interventions from observed data. The similarities go only to certain extent since fitting dropout as any single type of intervention defined in Eichler and Didelez, 2010 is not possible. This is due to a special characteristic that only dropout process exhibits, issue that has been implicitly acknowledged in Dawid and Didelez, 2010

But now we also have the collection U of unobservable domain variables (for simplicity we suppose throughout that which variables are observed or unobserved is the same under all regimes considered).

That is to say that latent variables remain latent from the beginning till the end of the study. This is impossible to assume for our setting because negative version of the statement above is the definition of dropout. Y or $\mathbf{V}$ can become latent at some point, if a person drops out.

Without the help of measure theoretic set up we discuss the intricate relationship between positivity, congeniality and identification. LI, due to its incremental nature, implicitly specifies more of the $F_C(\Theta_c)$ than it is necessary for extended SWEEP to identify the unconditional mean at t. Thus, congeniality in general becomes a larger issue for LI than for extended SWEEP. In the case where we can rely on (observed) positivity in our data this is a lesser issue, but it becomes important when positivity ceases to be a guarantee of existence of some/any $F_C(\Theta_c)$. Then congeniality of $T(T-1)/2$ models implied by ω_t's remains the only footing on which the existence of some/any $F_C(\Theta_c)$ hinges. Usually, in situations when identification is not an issue, congeniality can be classified as only one of the ways to parametrically missspecify true $F_C(\Theta_c)$, but when we try to facilitate identification by specifying only models for expected outcome (or its expected increments) congeniality becomes the criteria that determines if there is anything at all, unambiguous enough, left to identify. For these reasons, we argue that the comparison of the actual ability of the estimators to adjust for dropout in simulations should be judged using simulated data that preserves all the characteristics that both (or more) estimators need to preserve mutually equivalent interpretation of their estimands within any/some existent $F_C(\Theta_c)$. In the case of IPW EE this characteristic is (observed) positivity, since without it IPW EE is unusable. For extended SWEEP and LI congeniality of (implied) models specified has a similar role, but somewhat different consequences w.r.t. these estimators. While absence of some level of observed positivity makes IPW EE unusable, absence of congeniality of models makes numerically unbiased estimates possible at the cost of interpretation of the parameter estimated. In other words, we could have a sequence of ignorably missing increments with their mean unbiasedly estimated at each time 1 to t from observed data, but still in principle have unidentified unconditional mean at t (this notion is somewhat related to the concept of difference between initial and sequential Bayesian cut in Florens and Mouchart, 1985). This can happen if ω_t's, that made sure increments in adherers remain representable w.r.t. mean, imply t(t-1) $\eta-$models that can't be accommodated simultaneously by any single $F_C(\Theta_c)$.

Issues of congeniality are hardly a problem in complete parametric approaches since we rely on the complete likelihood. With that in mind, we suggest classifying missingness in longitudinal trials via level of observed positivity in the data in addition to classical dropout modes MCAR, MAR, NMAR if the estimators used rely only on semiparametric constraints as LI does.

Sensitivity analysis for LI is now possible to conceptualize. In fact the incremental machinery in this case could serve as a possible (conditional) solution to a problem often encountered in strategies for global sensitivity analysis. In a longitudinal trial this manifests in asking a subject-matter expert to speculate how the outcome at the end of the study affects the dropout at early stages of the study. Such a task goes against the ingrained human perception of causality as it relates to temporal order, where cause precedes effect. It is much easier to conceptualize how the simultaneous outcome or the one just preceding the event of dropout is causing dropout to occur. This retrospective aspect of eliciting the experts' opinion is an acknowledged issue in the work dealing with global sensitivity analysis for dropout (Scharfstein, Rotnitzky, and Robins, 1999, Scharfstein et al., 2014).

Eliciting only contemporaneous information about the association of the outcome process and dropout process and using linear increments to add this shift between dropouts and adherers over the whole course of study could be a strategy to deal with the above issue. Of course this kind of strategy hides assumptions and we would have to make sure that these don't take the sensitivity analysis too far from its goal of robustness in the sense of a non-parametrically defined model (Scharfstein, Rotnitzky, and Robins, 1999).

CHAPTER 3

Sensitivity analysis for linear increments (LISA)

In this chapter we define a structured way of departure from E-SEQ-MAR for LI. This way we can evaluate how robust estimates from LI are to dropout behavior that departures from the one captured by E-SEQ-MAR (see Figure 2.2).

3.1. Introduction

In recent years (see National Research Council report “The Prevention and Treatment of Missing Data in Clinical Trials”) continued efforts have been made to promote sensitivity analysis w.r.t. to missing data in clinical trials from its anecdotal and optional character to an indispensable part of the primary data analysis and a mandatory component of its reporting. For this reason we will refrain from further accentuating its contribution to acquiring a complete picture of the estimated effect in a clinical trial. A theoretical gold standard for sensitivity analysis would fall into the category of *global* sensitivity analysis (for a concise exposition on *local* and *global* characterization see Scharfstein et al., 2014; for a more detailed one see Daniels and Hogan, 2008, DH from now on, and/or Scharfstein, Rotnitzky, and Robins, 1999) and it can be conceptualized, with considerable simplification, by three components: sensitivity parameter(s) α, distribution F_O of observed data, and a functional relationship ξ between F_O and α that uniquely determines the distribution of the complete data F_C. This characterization roughly aligns ξ with the definition of non-parametrically identified model from Scharfstein, Rotnitzky, and Robins, 1999. Such a completely non-parametric gold standard is almost never achievable in a non-trivial applied setting for few different reasons. In longitudinal clinical trial with a continuous outcome (which will be our focus) and/or at least one continuous, contemporaneously measured covariate, these reasons can be classified under the category of a) feasible non-parametric estimation (see “curse of dimensionality” Robins and Ritov, 1997); b) preserving interpretability of a low-dimensional α through the whole sequence of planned repeated measurements (it can be very difficult to interpret and cogently present the influence of a three- or higher-dimensional α on one’s findings); and c) eliciting confident information about that particular α from domain experts (since α is not identified from the observed data we need to consult subject matter experts in order to decide on plausible values for α). Nevertheless, this gold standard is still useful as a reference and a starting point from which we can then move in a controlled manner, by sequentially imposing restrictions on ξ and/or F_O. Obstacles coming from these three categories are in no way non-overlapping and imposing restrictions in one usually changes the domain within which the other two interact. When it comes to estimating a treatment effect in longitudinal clinical trials, goals under b) and c) are particularly exclusive of each other. In contrast to the setting with repeated measurements, the goal of balancing all three of these aspects of the sensitivity analysis is much more achievable when we are interested in an effect of a one-time intervention instead of a vector sequence of treatments whose compound effect over time is of interest (see Diaz and Laan, 2013) for sensitivity analysis for a single time point in causal setting).

Scharfstein et al., 2014 gives a very nice exposition of the trade-off between these three goals in prominent methods for sensitivity analysis for dropout in longitudinal clinical trials. We will use this work to position our proposed method within the established literature. Roughly, it can be viewed as an extension of the pattern mixture approach of DH to longer sequences of repeated measurements while keeping the number of sensitivity parameters manageable, but still interpretable. Since we are defining sensitivity analysis for estimators that don't specify a model for probability of dropout, we, like Daniels and Hogan, use observed dropout proportions throughout. Due to conclusions from Robins and Ritov, 1997 the interpretation is confined within a certain set of parametric assumptions about conditional means given the past for observed data. In this regard it is different from Scharfstein et al., 2014. Similarity to this work comes from a mutual goal to allow experts to offer more confident information about the contemporaneous association between dropout and the outcome process. This endeavor is also made easier by the fact that our sensitivity parameter is defined on the scale of the outcome of interest and not, as in Scharfstein et al., 2014, on the log-odds ratio scale. We will not elaborate on the difficulty related to thinking about time shifted association, beyond quoting one of the authors from Scharfstein et al., 2014:

> *... we have found that subject matter experts who have been exposed to the RRS technology have difficulty quantifying how the distal outcome scheduled to be measured at the end of the study affects the risk of dropping out at intermediate time points. Rather, we found that these experts were more comfortable thinking about how the outcome scheduled to be measured at assessment k+1 affects the risk of dropping out between assessments k and k + 1.*

Resolving this issue is equivalent to simultaneously achieving goals b) and c) and this often happens inevitably at some expense of goals in category a).

The baseline setting to which our sensitivity framework collapses when the sensitivity parameter is set to zero is E-SEQ-MAR. This is the assumption we introduced in the preceding chapter under which the extended SWEEP estimator and linear increments (LI) estimator deliver consistent and unbiased estimates of the sequence of unconditional (marginal) means $E\left[\,\overline{\mathbf{Y}}_t \mid X = 1, 0\right] := (\mu_t, \ldots, \mu_1)$ These two estimation strategies offer same identificational assistance within the class of linear autoregressive mean models. This class is also the one within which a cogent sensitivity analysis w.r.t. non-ignorable dropout for these estimators is possible without modeling the probability of dropout. Further, outside of this class of models it is not immediately clear how to specify any lower dimensional α as a meaningful sensitivity parameter.

3.2. Methodology

3.2.1. Identification and sensitivity functions

We assume in this chapter that $\mathbf{V} = \emptyset$ and $X \in 0, 1$. All of the following assumptions and mathematical manipulations are assumed to be done separately in each treatment group $X \in \{0, 1\}$, thus conditioning event $X = x$ is implicitly omnipresent and propagated through each conditioning step. Let $E(Y_3) := \eta_{13} \equiv \eta_{13}(\overline{\mathbf{Y}}_0\,; \theta_{0,3})$ and $T = 3$. Assume further that $\eta_{23} \equiv \eta_{23}(\overline{\mathbf{Y}}_1\,; \theta_{1,3}) := E(Y_3 \,|\, R_1 = 1, \overline{\mathbf{Y}}_1)$ and $\eta_{33} \equiv \eta_{33}(\overline{\mathbf{Y}}_2\,; \theta_{2,3}) := E(Y_3 \mid R_2 = 1, \overline{\mathbf{Y}}_2)$. The treatment effect is then identified and estimated as $E(Y_3 \mid X = 1) - E(Y_3 \mid X = 0)$. Using only iterative expectation we have

$$\begin{aligned} E(Y_3) &= E(E[Y_3 \mid Y_1]) \\ &= E\Big(E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1]\, P(R_2 = 1 \mid R_1 = 1, Y_1) + \\ &\qquad E[Y_3 \mid R_2 = 0, R_1 = 1, Y_1] P(R_2 = 0 \mid R_1 = 1, Y_1)\Big) \;\square \end{aligned} \tag{3.1}$$

In the above expression $E[Y_3 \mid R_2 = 0, R_1 = 1, Y_1]$ is not identified from observed data. $P(R_2 =$

$0 \mid R_1 = 1, Y_1)$ can be estimated given the correct model, while $E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1]$ can still be segmented recursively through iterative expectation to see which parts of it are identifiable. For now we assume that

$$E[Y_3 \mid R_2 = 0, R_1 = 1, Y_1] = E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1] + \varphi_{31}(\boldsymbol{\gamma}; Y_1) \ \square \tag{3.2}$$

This makes

$$\eta_{23}(\overline{\mathbf{Y}}_1; \theta_{23}) = E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1] + \varphi_{31}(\boldsymbol{\gamma}; Y_1) P(R_2 = 0 | R_1 = 1, Y_1) \ \square \tag{3.3}$$

We can write $E(Y_3 \mid R_2 = 1, R_1 = 1, Y_1)$ using iterative expectation in an analogous manner as we did with $E(Y_3)$:

$$\begin{aligned} E(Y_3 \mid R_2 = 1, R_1 = 1, Y_1) &= E\Big(\quad E[Y_3 \mid R_2 = 1, R_1 = 1, Y_2, Y_1] \quad \Big| R_2 = 1, R_1 = 1, Y_1 \Big) \\ &= E\Big(E[Y_3 \mid R_3 = 1, R_2 = 1, R_1 = 1, Y_2, Y_1] \times \\ &\qquad P(R_3 = 1 \mid R_2 = 1, R_1 = 1, Y_2, Y_1) \qquad + \\ &\qquad E[Y_3 \mid R_3 = 0, R_2 = 1, R_1 = 1, Y_2, Y_1] \times \\ &\qquad P(R_3 = 0 \mid R_2 = 1, R_1 = 1, Y_2, Y_1) \quad \Big| R_2 = 1, R_1 = 1, Y_1 \Big) \ \square \end{aligned} \tag{3.4}$$

Again we have parts identifiable given correct models: $E[Y_3 \mid R_3 = 1, Y_2, Y_1]$ and $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$ and the non-identifiable part $E[Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1]$. Thus, we specify analogously

$$E[Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1] = E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1) \ \square \tag{3.5}$$

Similarly, this makes

$$\eta_{33}(\overline{\mathbf{Y}}_2\,; \theta_{33}) = E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1) P(R_3 = 0 | R_2 = 1, Y_2, Y_1) \ \square \tag{3.6}$$

Substituting (3.5) into (3.4) and (3.4) into (3.1) we get

$$\begin{aligned}
E(Y_3) &= E(E[Y_3 \mid Y_1]) \\
&= E\Big\{ \ E\Big(\ E[Y_3 \mid R_3 = 1, Y_2, Y_1] \ \times \ P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \ + \\
&\qquad [E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)] \ \times \ P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) \\
&\qquad \Big| \ R_2 = 1, R_1 = 1, Y_1 \Big) \times \ P(R_2 = 1 \mid R_1 = 1, Y_1) \qquad + \\
&\qquad \Big(E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1] + \varphi_{31}(\boldsymbol{\gamma}; Y_1) \Big) \ \times \ P(R_2 = 0 \mid R_1 = 1, Y_1) \Big\} \ \square
\end{aligned} \tag{3.7}$$

Now substitute (3.2) into (3.7)

$$E(Y_3) = E(E[Y_3 \mid Y_1])$$

$$\begin{aligned}
= E\Big\{ \; & E\Big(\; E[Y_3 \mid R_3 = 1, Y_2, Y_1] \; \times \; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \; + \\
& \Big[E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)\Big] \; \times \; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) \\
& \Big| \, R_2 = 1, R_1 = 1, Y_1\Big) \times P(R_2 = 1 \mid R_1 = 1, Y_1) \qquad + \\
& \Big(E\Big(\; E[Y_3 \mid R_3 = 1, Y_2, Y_1] \; \times \; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \; + \\
& \Big[E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)\Big] \; \times \; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) \\
& \Big| \, R_2 = 1, R_1 = 1, Y_1\Big) \; + \varphi_{31}(\boldsymbol{\gamma}; Y_1)\Big) \; \times P(R_2 = 0 \mid R_1 = 1, Y_1) \; \Big\} \; \square
\end{aligned}$$

Simplifying the above expression gives

$$\begin{aligned}
E(Y_3) = E\Big\{ \; & E\Big(\; E[Y_3 \mid R_3 = 1, Y_2, Y_1] \, \Big| \, R_2 = 1, R_1 = 1, Y_1\Big)\Big\} + \\
& E\Big\{E\Big(\varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1) \; \times \; P(R_3 = 0 \mid R_2 = 1, R_1 = 1, Y_2, Y_1) \, \Big| \, R_2 = 1, R_1 = 1, Y_1\Big)\Big\} + \\
& E\Big\{\varphi_{31}(\boldsymbol{\gamma}; Y_1) \, P(R_2 = 0 \mid R_1 = 1, Y_1) \quad \Big\} \; \square
\end{aligned} \tag{3.8}$$

In order to implement any sensitivity analysis for the treatment effect at $T = 3$ we need to specify functions $\varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)$ and $\varphi_{31}(\boldsymbol{\gamma}; Y_1)$ not identifiable from observed data. $\varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)$ can be recovered by surveying an expert about how the expected outcome Y_3 differs between those absent and those present at the third time point given identical histories up to and including time 2. This type of question focuses on the contemporaneous relationship of the dropout process R_t and the outcome process Y_t, and is natural to think about because outcome Y_3 and dropout indicator R_3 relate to the same time point in the study. A more complicated endeavor is to inform a plausible form and/or value of $\varphi_{31}(\boldsymbol{\gamma};Y_1)$. This sensitivity function relates in a time-shifted manner the outcome process Y_t and the dropout process R_t. Namely, an expert needs to think back in time about the influence of the third outcome on dropping out at the second time point as reflected by the shift in the conditional mean of Y_3 between those who drop out at time 2 and those who don't. Conversely, one can frame this as a question about how dropout at the second time point is governed by the yet-to-be-measured distant outcome at the end of the study. This is not a very intuitive way to think about association and it has in general proven to be hard for experts to conceive an intuitive and plausible choice for $\varphi_{kj}(\boldsymbol{\gamma};\overline{Y}_j)$ for $k-1 \gg j$ in case of such a lagged relationship. We suggest a way to incorporate and set $\varphi_{31}(\boldsymbol{\gamma};Y_1)$ implicitly by specifying analogously defined shifts $\delta_{21}(\boldsymbol{\alpha};Y_1)$ and $\delta_{32}(\boldsymbol{\alpha};Y_2,Y_1)$ on the scale of increments $\Delta Y_2 = Y_2 - Y_1$ and $\Delta Y_3 = Y_3 - Y_2$.

$$\begin{aligned}
\varphi_{31}(\boldsymbol{\gamma};Y_1) &= E(Y_3\,|R_2=0,Y_1) - E(Y_3 \mid R_2=1,Y_1) \\[2em]
&= E\Big(\Delta Y_3 + Y_2|R_2=0,R_1=1,\overline{Y}_1\Big) - E\Big(\Delta Y_3 + Y_2|R_2=1,R_1=1,\overline{Y}_1\Big) \\
&= E\Big(\Delta Y_3|R_2=0,R_1=1,\overline{Y}_1\Big) - E\Big(\Delta Y_3|R_2=1,R_1=1,\overline{Y}_1\Big) + \\
&\quad E\Big(Y_2|R_2=0,R_1=1,\overline{Y}_1\Big) - E\Big(Y_2|R_2=1,R_1=1,\overline{Y}_1\Big) \\[2em]
&= \delta_{31}(\boldsymbol{\alpha};Y_1) + \delta_{21}(\boldsymbol{\alpha};Y_1) \ \ \square
\end{aligned}$$

This does not solve the problem of specifying a shifted association, since a choice for $\delta_{31}(\boldsymbol{\alpha};Y_1)$ might be even harder to conceive than $\varphi_{31}(\boldsymbol{\gamma};Y_1)$. We do nevertheless make a step towards specifying $\varphi_{31}(\boldsymbol{\gamma};Y_1)$. The incremental paradigm allows us, formally for now, to chronologically partition the shift $\varphi_{31}(\boldsymbol{\gamma};Y_1)$ into a part contemporaneous with the specific dropout time 2 and any residual, lagged influence of the dropout at time 2 on the outcome at time 3. We can now be more confident at least about the part of $\varphi_{31}(\boldsymbol{\gamma};Y_1)$, since experts usually have a good idea about $\delta_{21}(\boldsymbol{\alpha};Y_1)$. For linear, autoregressive models η_{33} and η_{22} it trivially holds that $\delta_{32}(\boldsymbol{\alpha};Y_2,Y_1) = \varphi_{32}(\boldsymbol{\gamma};Y_2,Y_1)$ and $\delta_{21}(\boldsymbol{\alpha};Y_1) = \varphi_{21}(\boldsymbol{\gamma};Y_1)$. The form and value of $\varphi_{31}(\boldsymbol{\gamma};Y_1)$ for this class of models is then implicitly "completed" by our imputation algorithm (see beneath) based on a conditional normal distribution.

This implicit choice for $\delta_{31}(\boldsymbol{\alpha}; Y_1)$ is congenial with the assumption of *future independence* (see Section 3.2.2). Notice that, as soon as we move from φ's to sums of δ's and to models for increments implied by both η_{33} and η_{22}, we should move from identification of a single $E(Y_3 \mid \mathbf{X} = x)$ to that of a two-dimensional vector $E(Y_3, Y_2 \,|\, \mathbf{X} = x)$. It is theoretically possible to conceive a situation in which both η_{33} and η_{22} are correct but $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ and $\delta_{31}(\boldsymbol{\alpha}; Y_1)$ are both incorrectly specified in a way that their biases cancel out in their sum which makes $\varphi_{31}(\boldsymbol{\gamma}; Y_1)$ correct. With correctly chosen $\varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1)$ (and a model for probability of dropout) we can still identify $E(Y_3 \mid \mathbf{X} = x)$ although the estimate of $E(Y_2 \mid \mathbf{X} = x)$ based on LI would be biased. Within an incremental paradigm, such as LI, such situations are pathological and we will exclude such synergistic bias (or lack thereof) that can in general happen when we decide to estimate an unconditional expected outcome as a sum of unconditional expected increments previously marginalized in accordance with a valid joint distribution of the full data F_C. Thus, instead of coding ignorable dropout for T=3 in terms of values for φ_{31} and φ_{32}, we do it using $\delta_{32}(\boldsymbol{\alpha}; Y_2, Y_1) = \delta_{31}(\boldsymbol{\alpha}; Y_1) = \delta_{21}(\boldsymbol{\alpha}; Y_1) = 0$. Then, (3.3) is equivalent to first two rows of (3.9) while the third characterizes ignorable dropout w.r.t. $E(Y_2 \mid \mathbf{X} = x)$:

$$E[Y_3 \mid R_3 = 1, R_2 = 1, R_1 = 1, Y_2, Y_1] = E[Y_3 \mid R_2 = 1, R_1 = 1, Y_2, Y_1]$$

$$E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1] = E[Y_3 \mid R_1 = 1, Y_1]$$

$$E[Y_2 \mid R_2 = 1, R_1 = 1, Y_1] = E[Y_2 \mid R_1 = 1, Y_1] \quad \square \tag{3.9}$$

3.2.2. Model for observed increments and non-ignorable imputation algorithm

Anchor the sensitivity analysis at $\delta_{21}(\boldsymbol{\alpha}; Y_1)$

Specifying the sensitivity function $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ can be formulated as the following question: Given the same value at time 1, what is the shift w.r.t. the next expected incremental change introduced by the act of dropping out? (inform δ_{21}). Further we could ask an expert if she thinks this shift changes with time. A plausible belief would be that the longer subjects with the same history remain on study, the “closer” will they be with respect to the characteristics determining time of dropout. In our conditional pattern mixture approach, this will be reflected by a smaller shift in the next expected increment within a drop out-adherer pair. This sort of behavior can be captured by introducing an attenuation parameter $\rho \in [l, 1]$

$$\delta_{t,t-1} = \rho^{t-2} \times \delta_{21}$$

(3.10)

The effect of the dropout at a later study phase, as reflected by the shift in the conditional mean increment between adherers and dropouts, is smaller than at the beginning $(0.8 \leq l \leq 1)$. It will be hard for a domain expert to be able to specify $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ as a function of Y_1. Thus the loss of generality reflected by assuming $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ is a constant α in the light of experts simplified input should not be regarded as dramatic in practical sense.

Non-ignorable imputation algorithm

Our sensitivity analysis is based on imputation of the missing increments. For the expected observed increment ΔY_t at each time $t \in \{2, \ldots T\}$ we fit a linear autoregressive model. For $t = 1$ $\Delta Y_1 = Y_1$ per definition and the model for the expectation includes only the intercept.

$$E[\Delta Y_t \mid R_t = 1, \overline{Y}_{t-1}] = b^0_{\Delta Y_t} + b^{t-1}_{\Delta Y_t} Y_{t-1} + \cdots + b^1_{\Delta Y_t} Y_1 \ \square \qquad (3.11)$$

Notice that the coefficients $b^j_{\Delta Y_t}$ $j \in \{0, 1, \ldots, j-1\}$ are all equal to coefficients collectively denoted as $\theta_{t-1,t}$ in η_{tt} except for the coefficient in η_{tt} corresponding to $b^{t-1}_{\Delta Y_t}$. Corresponding θ is then equal to $b^{t-1}_{\Delta Y_t} + 1$. Estimates of the coefficients from these models are used to impute the value of the missing increment at time t for individual i according to the following recursive imputation algorithm:

1. For $\mathbf{t} = \mathbf{1}$ set $\hat{E}(\Delta Y_{i1} | R_{i1} = 1) = Y_{i1}$

   For $\mathbf{t} = \mathbf{2}, \ldots \mathbf{T}$

2. Fit $E(\Delta Y_t|R_t = 1, \overline{Y}_{t-1})$ on observed data

3. calculate $\hat{E}(\Delta Y_{i,t}|R_t = 0, \overline{Y}_{t-1}) + \rho^{t-2} \times \alpha$

4. sample $\varepsilon_{it} \sim N(0, \hat{\sigma}_t)$ where $\hat{\sigma}_t$ is the estimate of the residual standard error from $E(\Delta Y_t|R_t = 1, \overline{Y}_{t-1}) = b^0_{\Delta Y_t} + b^{t-1}_{\Delta Y_t} Y_{t-1} + \cdots + b^1_{\Delta Y_t} Y_1$

5. use $\Delta\hat{Y}_{it} = \hat{E}(\Delta Y_{i,t}|R_t = 0, Y_{t-1}, \mathbb{I}^{t-1}_{k_{t-1}}) + \rho^{t-2} \times \alpha + \varepsilon_{it}$ as the imputed increment

Notice that this is a recursive procedure since for a person i dropping out at t, $\Delta Y^{\text{miss}}_{it}$ is the only data point that will be imputed exclusively from his/her observed data. All subsequent ones will use their data imputed at the previous step by the algorithm. $\rho^{t-2} \times \alpha$ is a specific shift in the intercept of the model in (3.11). For patients that drop out at $t = 2$ this recursive procedure introduces an implied shift for $E(Y_3 \mid R_2 = 0, Y_2, Y_1) - E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1) = \rho\alpha$ also. One way of interpreting this implied shift is that, after imputing the data on missing Y_2, we are done correcting the outcome Y_3 w.r.t. dropping out at time 2, since at the next step, we will use that imputed Y_2 to impute $\Delta\hat{Y}_{i3}$. What's left is to specify how the act of dropping out at 3 is reflected in the shift δ_{32}. This implies that δ_{31} is 0. This can be interpreted as shifting or accumulating the whole $\varphi_{31} = \delta_{31} + \delta_{21}$ in δ_{21}, a parameter our domain experts can realistically have accurate input on. In general δ_{31} of course does not have to be 0 and it might not be realistic to expect that the probabilistic structure of F_C is simple enough to capture the whole effective shift φ_{31} in outcome through only δ_{21}. There is though an assumption, which we call *future independence given present*, about the joint distribution of the complete data F_C that does allow, in principle, such aggregation without introducing bias. A version of future independence (under the name *future ignorability*) in causal setting is introduced by Joffe, Yang, and Feldman, 2010, while its version for longitudinal clinical trial data with dropout implies assumption 1 in Scharfstein et al., 2014. We will state it for general number of observations T as

$$Y_t \perp R_j \mid \overline{\mathbf{Y}}_j, \text{ for } 1 \leq j < t, 3 \leq t \leq T \ \square \tag{3.12}$$

For $T = 3$ this means that $f(Y_3|R_2 = 0, Y_2, Y_1) = f(Y_3|R_2 = 1, Y_2, Y_1)$.

This assumption allows us, even with parameter aggregation implied by non-ignorable imputation algorithm, to interpret $\sum_{i=1}^{n}\sum_{j=1}^{3}\hat{\Delta Y}_{ij}$ as $E(Y_3)$. We show a sketch of a proof in the Appendix B.2.

3.3. Behavioral economic interventions to reduce CVD risk

3.3.1. Observed data

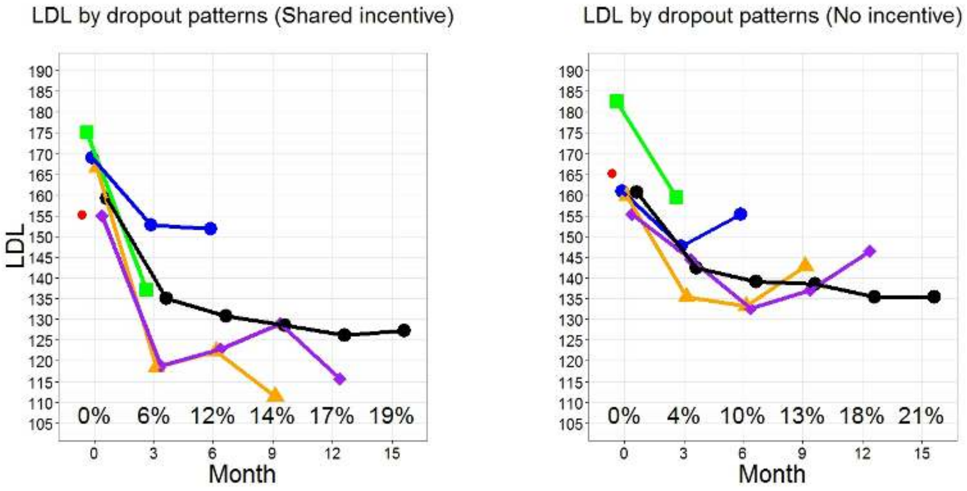
A modified dataset coming from a multi-center cluster-randomized controlled trial comparing three alternative economic interventions for reducing LDL cholesterol among patients with high cardiovascular risk will be used to exemplify the use of our method. Participating primary care physicians (PCPs) from the University of Pennsylvania, Geisinger Health System, and Harvard Vanguard Medical Associates were randomly assigned to one of four study arms: control (C), physician incentives (PHYS), patient incentives (PAT), and shared physician-patient incentives (shared). The outcome of interest was low-density lipoprotein (LDL). Eligible and participating patients of these PCPs were allocated to the arm to which their PCP had been randomized. Longitudinal data on LDL was collected every 3 months over 15 months ($t = 1, \ldots, 6$). We restrict our sensitivity analysis to the effect of the shared intervention (n=347) compared to control (n=365). We will further modify any intermittent missingness into drop-out by leaving out any outcome recorded after the first occasion a subject has missed. Sensitivity analysis is presented for the mean change in LDL at month 15. There was a modest to considerable portion of missing observations in the observed data with around 20% of observations missing at 15 (t=6) months (see Table 3.1).

	Shared-incentive arm (n=347)	Control arm (n=365)
t=2	5.7 %	3.8 %
t=3	11.5 %	10.1 %
t=4	14.4 %	13.4 %
t=5	17.0 %	17.5 %
t=6	19 %	21 %

Table 3.1: Marginal dropout proportions

We do not distinguish among reasons for dropout. Figure 3.1 shows mean LDL values per dropout pattern. Each profile line corresponds to the single dropout pattern so the groups used for the calculation of the mean are disjoint. We can see a differential tendency for dropout, (assuming explainable dropout) in two treatment groups. It seems that subjects in the shared group usually

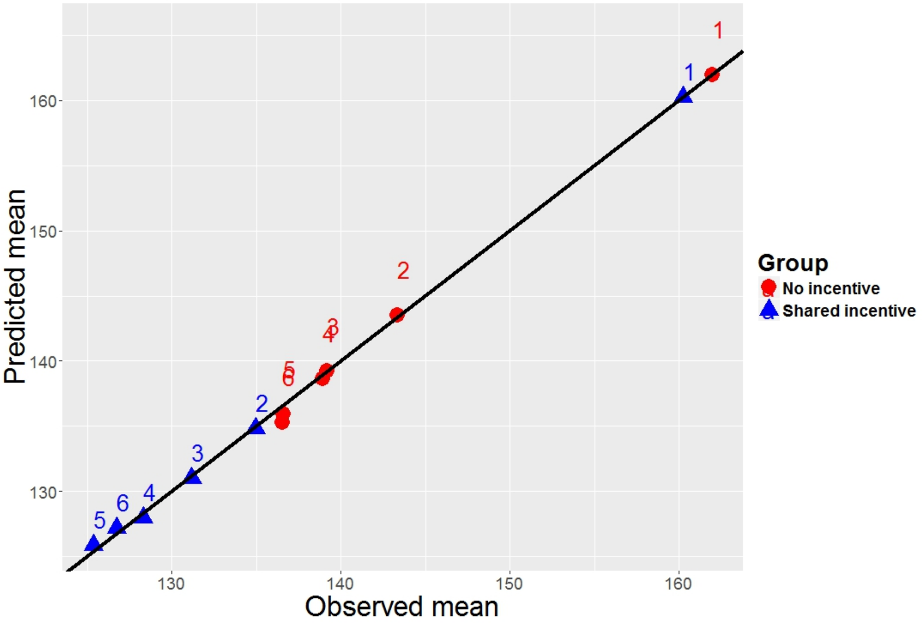
Figure 3.1: Mean LDL values per dropout pattern



experience a decrease of the mean LDL before they decide not to appear at the next visit. This pattern is exactly opposite in the control group. There, subjects who are not observed at the next time point have a noticeable increase in their mean LDL. This directional dissonance w.r.t MAR dropout and the last observed change in LDL is especially apparent for times $t = 3, 4$ and 5.

Figure 3.2 shows on the mean level the observed data fit at each time t for both treatment groups. The fit of linear autoregressive models w.r.t. to the mean seems to be quite well aligned along the main diagonal of the graph. Table 3.2 shows treatment effect estimated using a) observed means (OLS), b) inverse probability of observing-weighted estimator (IPW) with last observed outcome only, in the logit model used to estimate probability of dropout and c) LI and SWEEP estimators. b) and c) estimators yield unbiased estimates of the treatment effect in case of SEQ-MAR dropout or if the version of (3.9) for $T = 6$ holds, respectively. We see by comparing OLS and other two/three that, at least with respect to models used for IPW and LI/SWEEP, we can exclude MCAR as a dropout mode in this data. This is in accordance with the plots in Figure (3.1). To further diverge from dropout explainable by observed data to non-ignorable dropout we will utilize the non-ignorable imputation algorithm for increments as governed by different choices of $\delta_{2,1}(\boldsymbol{\alpha})$ and ρ.

Figure 3.2: Observed $\frac{1}{n_t}\sum_{i=1}^{n_t} Y_{it}$ vs. predicted $\frac{1}{n_t}\sum_{j=1}^{t}\sum_{i=1}^{n_t} \hat{E}(\Delta Y_{i,j}|R_{i,j} = 1, \overline{Y}_{i,t-1})$, where $n_t = \sum_{i=1}^{n} \mathbb{I}_{\{R_{i,t}=1\}}$



estimator	Shared inc.-effect
OLS	-6.39
SWEEP	-8.06
IPW	-8.92
LI	-8.06

Table 3.2: Difference in differences (negative) at $T = 6$ between Shared and No Incentive (SWEEP and LI resulting in identical estimates)

3.3.2. Sensitivity analysis for shared incentive effect on LDL

Moving away from ignorable dropout

The purpose of the sensitivity analysis is to subject the effects from the rows 2 and 3 of the Table 3.2 calculated under the assumption $\delta_{2,1}(\alpha; \text{LDL}_1) = 0$ to a test of robustness of its significance and direction to different levels of bias due to non-ignorable dropout. An analogous way to subject IPW estimator from Table 3.2 to such a test could be to use a non-ignorable version of the logit model in which coefficient next to the outcome contemporaneous with the time dropout is modeled for, serves as a sensitivity parameter. This is a similar, though oversimplifying route compared to the one from Scharfstein et al., 2014.

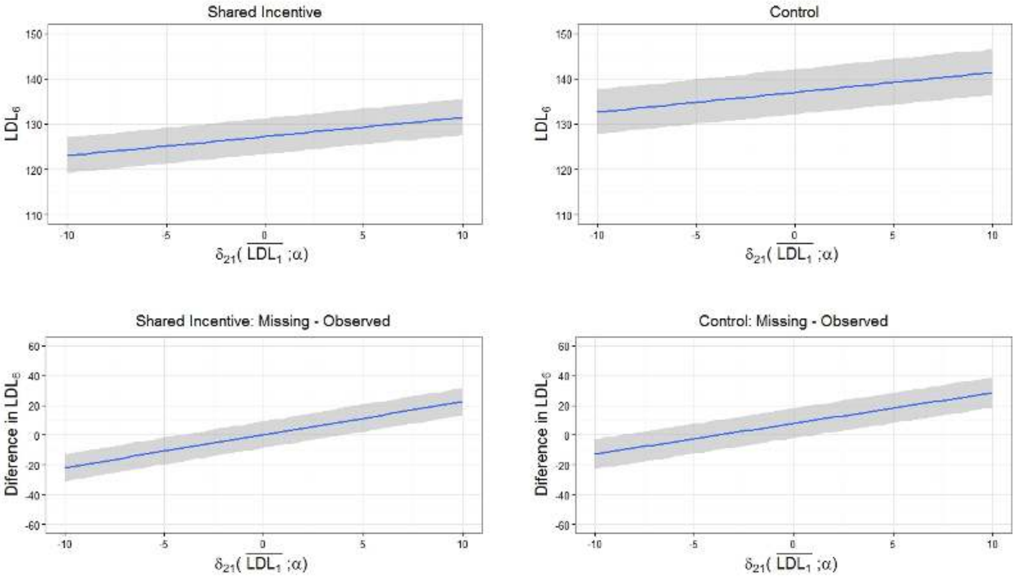
Our sensitivity analysis is centered around a pair of values for $\delta_{2,1}$ in the shared incentive and control groups. For each of these pairs we can calculate a difference in implied means between groups. Additionally, we calculated 95% non-parametric bootstrap CI (n=10 000) for the effect estimate at each of the pairs. We can then evaluate the robustness of the effect corresponding to any choice of the pair $\delta_{2,1} = 0$ for both groups. The significance of the effect is established by checking if its 95% bootstrap confidence interval includes 0. Although we can use the approach described for sensitivity analysis of shared incentive effect estimated by LI at any time t we will present results and conclusions for the last time point $T = 6$. The most conservative version of our approach is to allow no attenuation over time with $\rho = 1$. We show plots for α between -10 and 10 (with step of 0.5) since this area proves to be enough to get a complete picture of the robustness of the analysis to the version of the ignorability assumption (3.9) for $T = 6$.

Results and implications

Figure 3.3 shows treatment specific means and difference in means at $T = 6$ between missing and observed. We can see that the conditional shift $\delta_{t,t-1}(\boldsymbol{\alpha}; \overline{\text{LDL}}_{t-1}) = \alpha$ for $t \in \{2, \dots, 6\}$ between -10 mg/dL and 10 mg/dL yields a marginal difference in the mean between missing and observed at time 6 that ranges between -20 mg/dL and 20 mg/dL. The slope of the linear approximations of the curves in the first row of plots is similar in both groups and the rate of increase seems to be 1 mg/dL for each 4 mg/dL change in conditional shift α. This constant, but attenuated, change in the marginal mean in each group w.r.t. change in α is due to a) modest dropout rates ($< 20\%$ at time 6) and b) the fact that, within the class of models we assume, the marginal shift is mathematically a weighted sum of α shifts (see Appendix B.2 for $T = 3$) where weights are observed (non-cumulative) dropout rates.

By viewing these plots experts can decide on the plausible range for α; since the direct numerical correspondence of the shifts and the difference in marginal means is not possible to establish for $T > 3$ the plots can serve as a control check. The advantage of this approach is that the shifts and

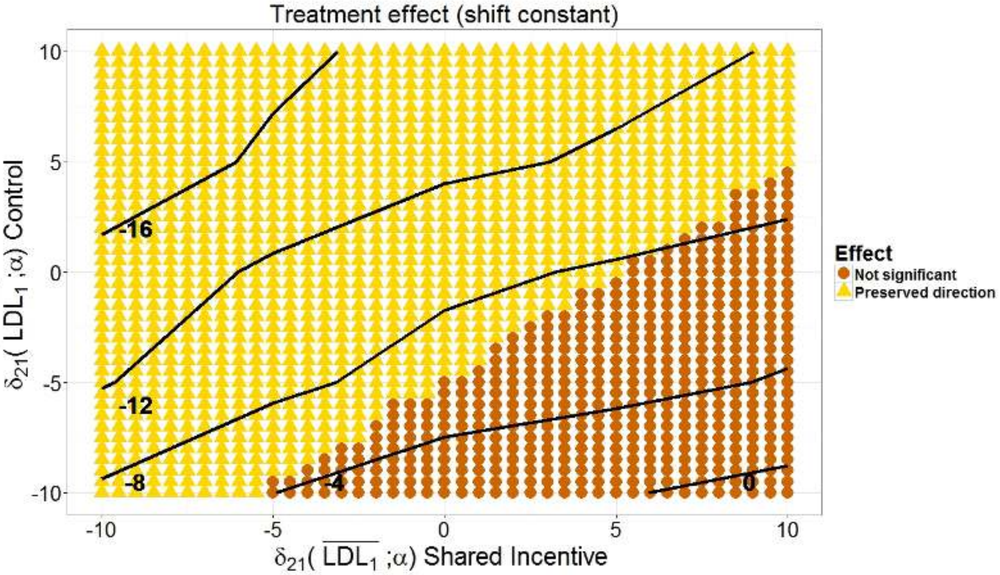
Figure 3.3: Treatment specific mean LDL at $T = 6$ as a function of α for α constant over time and Ω_{Y_j}



differences in means are both observed on the same outcome scale, which makes it easier to think about appropriate and realistic correspondence of the shifts up to time t and differences in marginal means at time t.

Figure 3.4 shows a contour plot of treatment effects for each pair of $(\alpha(\text{sh. incentive}), \alpha(\text{control}))$. For (0,0) we will see effect as shown in Table 3.2. The dropout pattern-specific mean profiles in Figure 3.1 might be interpreted as to suggest that non-ignorable dropout from control and shared incentive is more plausible to be situated in the upper left part of the plot as coded by values $\alpha(\text{shared incentive}) < 0$ and $\alpha(\text{control}) > 0$. "Movement" in this direction on the plot leaves the treatment effect identical in direction and naturally higher in magnitude. Significance characteristic of the treatment effect does not change in this part of the plot either. On the other hand, staying strictly on the main diagonal can be interpreted as assuming non-ignorable but identical dropout - outcome behavior in both treatment groups as reflected by the conditional shifts in increments. Moving the main diagonal orthogonally by 5 mg/dL to the lower-right side of the plot we encounter the area for which treatment effects becomes not significant. In particular, a mean shift of $5(-5)$ mg/dL for each increment at $t = \{2, 3, 4, 5, 6\}$ between dropouts and adherers in shared incentive (control) group while we keep (3.9) true in the other group will lead to treatment effect becoming not significant. This might seem too sensitive as described by $\alpha = 5$ mg/dL but we should keep in mind “recursive shifting” that is going on behind the scenes as described in non-ignorable imputation algorithm. There are 5 different occasions for dropout until $T = 6$ and a person that drops out at time 2 will be shifted 5 times according to chosen α. If we hold the control group at $\alpha = 0$ while we let shared incentive have $\alpha = 5$ we can identify in Figure 3.3 (lower, left plot) that this shift corresponds to 20 mg/dL difference between the means of dropouts and adherers in shared incentive group. To put this in perspective, the mean LDL values in this group dropped from around 160 mg/dL to 130 mg/dL over 15 months. This means that $\alpha = 5$ implies a shift in marginal mean between dropouts and adherers that is $\approx 66\%$ of the mean decrease in LDL under ignorable dropout over the whole study period. Such relative magnitude of the difference between adherers and dropouts at $T = 6$ can be interpreted as generous. With this in mind we can say that Figure 3.4 offers a depiction of a generous buffer for treatment effects in Table 3.2 w.r.t. to non-ignorable dropout.

Figure 3.4: Sensitivity plot for the treatment effect as a function of α in both groups for α constant over time



3.4. Discussion

We described the possible sensitivity analysis approach for linear increments estimator first introduced by Diggle, Farewell, and Henderson, 2007. In the lengthy discussion following this work questions were raised what is a general condition for ignorable dropout w.r.t. this incremental strategy when it comes to estimating unconditional treatment effect. As pointed out by the authors, such transplantation of a technique from coming from stochastic processes setting, where time is random to a setting where observation times are non-random and predetermined can be quite hard to incorporate into the existent ignorability categories as MAR or SEQ-MAR. It is nevertheless crucial for any responsible use of estimators that adjust for dropout to try and conceive the most reasonable and internally most consistent way to "stress-test" our conclusions gathered by such a strategy. To keep this internal coherence we had to restrict the class of models for expected increments to the class for which devising such a cogent sensitivity analysis is possible.

The presented sensitivity analysis for LI within the class of linear autoregressive models offers a sensitivity analysis option for extended SWEEP estimator (Robins et al. 1995.) as well. It allows experts to think only about contemporaneous association between the outcome and dropout process. Their decisions about this association will then imply the magnitude of those time shifted associations needed for identification of the marginal mean. These implied values are congenial (w.r.t. expectation) with the assumption of *future independence given present* on F_C.

Our approach is restricted to the fixed $\delta_{2,1}(\alpha; Y_1)$. Extending it to the case where $\delta_{2,1}(\alpha; Y_1)$ is piecewise constant within Ω_{Y_t} is technically possible, but the interpretation (and/or identification) of the estimate as the marginal (unconditional) mean would be even heuristically harder to claim without further assumptions about simplifying the probabilistic structure of F_C.

Nevertheless, our sensitivity approach relies on domain experts knowledge to the extent to which expert alone would be comfortable about speculating. It offers a natural time respecting partition of the expected shift in outcome to shifts in increments. The increments are then shifted only w.r.t. contemporaneous dropout occurrence, while the rest of the shift is implicitly assumed to be zero, which aligns with values implied for linear shifts when future independence given present holds. This all is done while using marginal, observed dropout rates so any speculation about the parametric form of dropout is not necessary and cannot be a source of variability of the conclusions.

Nevertheless, there are drawbacks coming with this parameter aggregation in the form of implied shifts. These are not informed by the domain expert and might not be true. Bias that comes from misspecification of these shifts is attenuated by the use of observed marginal dropout proportions and in our future work we plan to show that it's influence on the decision about robustness of the findings to dropout does not invalidate our approach.

Comparing in some standardized way our, and the approach by Scharfstein et al. (2014) might offer some insight in how sensitive is the sensitivity analysis w.r.t. modeling the probability of dropout. Relating the value of the coefficient in such a non-ignorable version of a logit model to a value of the shift $\delta_{2,1} \neq 0$ for which $E(\overline{Y}_6 \mid X = 1) - E(\overline{Y}_6 \mid X = 0)$ yields a comparable estimate is, without assuming the full joint distribution for the complete data, not possible. Additionally, this discrepancy extends to the individual level data as well. Any model used for estimating probability of dropout in IPW estimator will not lead to extrapolating outside of observed data. On the other side, even for $\delta_{2,1} = 0$ the assumed mean model for observed ΔY_t can imply prediction for Y_t based on some value of Y_{t-1} among dropouts that is outside of the range of observed Y_t's. What we can do w.r.t. correspondence between the shift in the mean at time t and the influence of the contemporaneous outcome Y_t on the log-odds scale on dropout, is a post-hoc estimation of a non-ignorable logit model, after we impute the data according to the assumed shift $\delta_{2,1}$ as described for the non-ignorable imputation algorithm.

On the other side, the "luxury" of not modeling dropout is payed by the **restricted class of models within which we define our parameter of interest**. Often, justifiability of such simple parametric assumptions w.r.t. observed data on the level of an individual study participant will be disputable. Nevertheless, these or some other, less or more restrictive structural assumptions are indispensable, if one wants to preserve a balance between goals under a) b) and c) (see Section 3.1) in a longitudinal clinical trial where influence of the dropout extends over several time points.

It will be useful for a better understanding of the sensitivity analysis for linear increments (LISA from now on) as well as sensitivity analysis for dropout in general to describe where LISA finds its place among the existent and established sensitivity analysis tools. A very natural choice for positioning LISA is a pattern mixture approach for sensitivity analysis for dropout described in Daniels and Hogan, 2008. In the next chapter we will see how LISA can be perceived as an extension of this approach to longer sequences of observations without increasing the number of sensitivity parameters dramatically (characteristic to which approach from Daniels and Hogan, 2008 is prone to. Under future independence the LISA parameter α maintains intuitive interpretation within Daniels and Hogan's pattern mixture set up as well.

CHAPTER 4

LISA AS AN EXTENSION OF A PATTERN MIXTURE APROACH TO SENSITIVITY ANALYSIS UNDER FUTURE INDEPENDENCE

4.1. Introduction

Daniels and Hogan, 2008 presented a general pattern mixture approach to sensitivity analysis of the marginal treatment effect w.r.t. non-ignorable dropout in longitudinal trials. This approach, in which one specifies fully parametric distribution for each dropout pattern separately, results in clear partitioning of the parameters into those identifiable from observed data and those that are not. This is a very useful feature of pattern mixture approach when one is faced with specifying a sensitivity analysis to non-ignorable dropout. Analog mean and variance structures specified for each dropout pattern (at each scheduled time of observation) yield natural pairs of non-identifiable and identifiable parameters whose differences offer themselves readily to serve as sensitivity parameters to non-ignorable dropout. One drawback of this approach is that the number of sensitivity parameters quickly becomes unwieldy when applied to longer sequences of observations. Reaching an overall conclusion about the robustness of the results to non-ignorable dropout as well as conveying it in a coherent and understandable way is not easily accomplished when there is a large number of “levers” on which the final conclusion depends. We will show that LISA can be perceived as an extension of Daniels and Hogan’s pattern mixture approach (DH from now on) to longer sequences of observations, while keeping interpretability of the small number of sensitivity parameters. For better understanding we describe the relationship between these two methods under general conditions and under an assumption of future independence (see Scharfstein et al., 2014). We specify the relationship between sensitivity parameters in one and the other approach.

4.2. Daniels and Hogan’s approach for sensitivity analysis for $E(Y_3 \mid X = 1) - E(Y_3 \mid X = 0)$

DH take a pattern mixture approach to sensitivity analysis within a normal (distributional) structure for 3 time points. This approach has its benefits in that it allows clear and unambiguous specification and enumeration of unidentified parameters. We illustrate the DH approach first. Assume everybody is observed at $t = 1$ and drop out is possible at times 2 and 3. We will code the dropout patterns as $(R_2 = 0)$, $(R_2 = 1, R_3 = 0)$ and $(R_3 = 1)$. Let

$$\begin{aligned}
Y_2|Y_1, R_2 = 1 &\sim N\Big(\alpha_0^{(\geq 2)} + \alpha_1^{(\geq 2)}Y_1, \quad \tau_2^{(\geq 2)})\Big) \\
Y_2|Y_1, R_2 = 0 &\sim N\Big(\alpha_0^{(1)} + \alpha_1^{(1)}Y_1, \quad \tau_2^{(1)}\Big) \\
Y_3|Y_2, Y_1, R_3 = 1 &\sim N\Big(\beta_0^{(3)} + \beta_2^{(3)}Y_2 + \beta_1^{(3)}Y_1, \quad \tau_3^{(3)}\Big) \\
Y_3|Y_2, Y_1, R_2 = 1, R_3 = 0 &\sim N\Big(\beta_0^{(2)} + \beta_2^{(2)}Y_2 + \beta_1^{(2)}Y_1, \quad \tau_3^{(2)}\Big) \\
Y_3|Y_2, Y_1, R_2 = 0 &\sim N\Big(\beta_0^{(1)} + \beta_2^{(1)}Y_2 + \beta_1^{(1)}Y_1, \quad \tau_3^{(1)}\Big) \qquad (4.1)
\end{aligned}$$

Note that we do not differentiate the parameters at $t = 1$ (since no drop out can occur then, so no need for distinguishing patterns as we will see). This normal theory set up hinges on 20 parameters (2 for $t = 1$, 5 variance parameters with the rest of 13 describing the mean structure within each drop out pattern.

4.3. Linear increments sensitivity analysis (LISA) for $E(Y_3 \mid X = 1) - E(Y_3 \mid X = 0)$

Since increment (change) and not outcome is a central modeling object in the linear increments estimator we define LISA within the class of linear, autoregressive models. Within this class identificational potential w.r.t. identifying $E(Y_t \mid X = 1)$ of a model specification is equivalent for increments and outcomes. There is also a one to one relationship between coefficients of the model ω_t for $E\left[\Delta Y_t \mid \overline{\mathbf{Y}}_{t-1}, R_t = 1,\right] = \omega_t(\overline{\mathbf{Y}}_{t-1}\,; \mathbf{b}_{\Delta Y_t})$ and the model $\eta_{t,t}$ for $E\left[Y_t \mid \overline{\mathbf{Y}}_{t-1}, R_t = 1,\right] = \eta_{t,t}(\overline{\mathbf{Y}}_{t-1}\,; \theta_{t-1,t})$. Using iterative expectation within this class of models we can write $E(Y_3)$ as

Remember we defined in previous chapter that

$$\begin{aligned}
E[Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1] &= E[Y_3 \mid R_3 = 1, Y_2, Y_1] + \varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1) \\
E[Y_3 \mid R_2 = 0, R_1 = 1, Y_1] &= E[Y_3 \mid R_2 = 1, R_1 = 1, Y_1] + \varphi_{31}(\boldsymbol{\gamma}; Y_1) \qquad (4.2)
\end{aligned}$$

Within the class of linear autoregressive models we can make a correspondence between these functions describing the shift in the mean outcome and analogous functions for increments.

$$\underbrace{E\Big(\Delta Y_3|R_3=0,R_2=1,\overline{Y}_2\Big)-E\Big(\Delta Y_3|R_3=1,R_2=1,\overline{Y}_2\Big)}_{\delta_{32}(\boldsymbol{\alpha};Y_1)}=\varphi_{32}(\boldsymbol{\gamma};Y_2,Y_1)$$

$$\underbrace{E\Big(\Delta Y_3|R_2=0,R_1=1,\overline{Y}_1\Big)-E\Big(\Delta Y_3|R_2=1,R_1=1,\overline{Y}_1\Big)}_{\delta_{31}(\boldsymbol{\alpha};Y_1)}+$$

$$\underbrace{E\Big(Y_2|R_2=0,R_1=1,\overline{Y}_1\Big)-E\Big(Y_2|R_2=1,R_1=1,\overline{Y}_1\Big)}_{\delta_{21}(\boldsymbol{\alpha};Y_1)}=\varphi_{31}(\boldsymbol{\gamma};Y_2,Y_1)$$

$$\underbrace{E\Big(\Delta Y_2|R_2=0,R_1=1,\overline{Y}_1\Big)-E\Big(\Delta Y_2|R_2=1,R_1=1,\overline{Y}_1\Big)}_{\delta_{21}(\boldsymbol{\alpha};Y_1)}=\varphi_{21}(\boldsymbol{\gamma};Y_1)$$

LISA is based on prediction. For the expected observed increment ΔY_t at each time $t \in \{2,3\}$ we fit a linear autoregressive model with the full history of observed outcomes. For $t = 1$ $\Delta Y_1 = Y_1$ per definition and the model for the expectation includes only the intercept.

$$E[\Delta Y_2 \mid R_2 = 1, Y_1] = b^0_{\Delta Y_2} + b^1_{\Delta Y_2} Y_1 = \omega_2$$

$$E[\Delta Y_3 \mid R_2 = 1, Y_2, Y_1] = b^0_{\Delta Y_3} + b^2_{\Delta Y_3} Y_2 + b^1_{\Delta Y_3} Y_1 = \omega_3$$

Estimates of the coefficients from ω_2 and ω_3 are then used to impute the missing increment at time t for individual i according to the recursive imputation algorithm described in Chapter 3

Notice that within LISA there is no explicit assumption about the value of $E[Y_3 \mid R_2 = 0, R_1 = 1, Y_2, Y_1] - E[Y_3 \mid R_3 = 1, Y_2, Y_1]$ which is one of the sensitivity parameters for DH approach and corresponds to the relationship of conditional means for last and 3rd last row of (4.1).

4.4. LISA as extension of DH to longer sequence of observations

We write out $E(Y_3)$ using the set up in (4.1)

$$E(Y_3) = E\Big(E[Y_3 \mid Y_1]\Big)$$

$$= E\Big(E[Y_3 \mid R_2 = 1, Y_1]\; P(R_2 = 1 \mid Y_1) + E[Y_3 \mid R_2 = 0, Y_1]\; P(R_2 = 0 \mid Y_1)\Big)$$

$$\begin{aligned}
&= E\Big(E\Big[E(Y_3 \mid R_2 = 1, Y_2, Y_1) \Big| R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1) + \\
&\quad E\Big[E(Y_3 \mid R_2 = 0, Y_2, Y_1) \Big| R_2 = 0, Y_1\Big]\; P(R_2 = 0 \mid Y_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&= E\Big(E\Big[E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1)\; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) + \\
&\qquad E(Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1)\; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) \mid R_2 = 1, Y_1\Big] \times \\
&\quad P(R_2 = 1 \mid Y_1) + \\
&\quad E\Big[E(Y_3 \mid R_2 = 0, Y_2, Y_1) \mid R_2 = 0, Y_1\Big]\; P(R_2 = 0 \mid Y_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&= E\Big(E\Big[\Big(\beta_0^{(3)} + \beta_2^{(3)}Y_2 + \beta_1^{(3)}Y_1\Big)\; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) + \\
&\qquad \Big(\beta_0^{(2)} + \beta_2^{(2)}Y_2 + \beta_1^{(2)}Y_1\Big)\; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) \Big| R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1) + \\
&\qquad E\Big[\Big(\beta_0^{(1)} + \beta_2^{(1)}Y_2 + \beta_1^{(1)}Y_1\Big) \Big| R_2 = 0, Y_1\Big]\; P(R_2 = 0 \mid Y_1)\Big)
\end{aligned}$$

$$\begin{aligned}
&= \beta_0^{(3)} P(R_3 = 1, R_2 = 1) + \\
&\quad \beta_2^{(3)} E\Big(E\Big[Y_2\; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\Big| R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_1^{(3)} E\Big(Y_1\; P(R_3 = 1, R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_0^{(2)} P(R_3 = 0, R_2 = 1) + \\
&\quad \beta_2^{(2)} E\Big(E\Big[Y_2\; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)\Big| R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_1^{(2)} E\Big(Y_1\; P(R_3 = 0, R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_0^{(1)} P(R_2 = 0) + E\Big(\Big[\beta_2^{(1)}\Big(\alpha_0^{(1)} + \alpha_1^{(1)}Y_1\Big) + \beta_1^{(1)}Y_1\Big]\; P(R_2 = 0 \mid Y_1)\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \beta_0^{(3)} P(R_3 = 1, R_2 = 1) + \\
&\quad \beta_2^{(3)} E\Big(E\Big[Y_2\, P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\Big|\, R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_1^{(3)} E\Big(Y_1\, P(R_3 = 1, R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_0^{(2)} P(R_3 = 0, R_2 = 1) + \\
&\quad \beta_2^{(2)} E\Big(E\Big[Y_2\, P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)\Big|\, R_2 = 1, Y_1\Big]\; P(R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_1^{(2)} E\Big(Y_1\, P(R_3 = 0, R_2 = 1 \mid Y_1)\Big) + \\
&\quad \beta_0^{(1)} P(R_2 = 0)\ + \beta_2^{(1)} \alpha_0^{(1)} P(R_3 = 0) + \\
&\quad \beta_2^{(1)} \alpha_1^{(1)} E\Big(Y_1\, P(R_2 = 0 \mid Y_1)\Big) + \\
&\quad \beta_1^{(1)} E\Big(Y_1\, P(R_2 = 0 \mid Y_1)\Big)
\end{aligned}
$$

DH make some assumptions to limit the number of parameters in (4.1). They assume that the variances τ_2 are the same between patterns $(R_2 = 1)$ and $(R_2 = 0)$, and τ_3 among $(R_2 = 0)$, $(R_2 = 1, R_3 = 0)$, and $(R_2 = 1, R_3 = 1)$. Further, and more important for simplifying the expression above, they assume the slopes to be equal as well. We write this constraint as

$$
\begin{aligned}
\tau_1 &= \tau_2 = \tau_3 \\
\tau_2^{(1)} &= \tau_2^{(\geq 2})\\
\tau_3^{(1)} &= \tau_3^{(2)} = \tau_3^{(3)} \\
\alpha_1^{(1)} &= \alpha_1^{(\geq 2)} \\
\beta_2^{(1)} &= \beta_2^{(2)} = \beta_2^{(3)} \\
\beta_1^{(1)} &= \beta_1^{(2)} = \beta_1^{(3)}
\end{aligned}
\tag{4.3}
$$

This simplifies the final expression for $E(Y_3)$ further, so

$$E(Y_3) = \beta_0^{(3)}\ P(R_3 = 1, R_2 = 1) + \beta_0^{(2)}\ P(R_3 = 0, R_2 = 1) + \beta_0^{(1)}\ P(R_2 = 0) +$$

$$\beta_2^{(3)}\alpha_0^{(\geq 2)} + \beta_2^{(3)}\Big(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\Big)\ P(R_2 = 0) +$$

$$\beta_2^{(3)}\alpha_1^{(\geq 2)}E\Big(Y_1\Big) + \beta_1^{(3)}E\Big(Y_1\Big)$$

If we rewrite this using the fact that $P(R_3 = 1, R_2 = 1) + P(R_3 = 0, R_2 = 1) + P(R_2 = 0) = 1$ we have

$$E(Y_3) = \beta_0^{(3)}\ + \Big(\beta_0^{(2)} - \beta_0^{(3)}\Big)\ P(R_3 = 0, R_2 = 1) + \Big(\beta_0^{(1)} - \beta_0^{(3)}\Big)\ P(R_2 = 0) +$$

$$\beta_2^{(3)}\alpha_0^{(\geq 2)} + \beta_2^{(3)}\Big(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\Big)\ P(R_2 = 0) +$$

$$\beta_2^{(3)}\alpha_1^{(\geq 2)}E\Big(Y_1\Big) + \beta_1^{(3)}E\Big(Y_1\Big) \tag{4.4}$$

At this point we can relate some of the terms in (4.4) to some of the sensitivity parameters used for LISA. $\beta_0^{(2)} - \beta_0^{(3)}$ is equal to $\varphi_{32}(\boldsymbol{\gamma}; Y_2, Y_1) = \rho\alpha$ while $\alpha_0^{(1)} - \alpha_0^{(\geq 2)}$ corresponds to $\varphi_{21}(\boldsymbol{\gamma}; Y_1) = \alpha$. These two are also parameters that non-ignorable imputation algorithm uses to impute the values of Y_2 and Y_3 for individuals missing at those times. There is no analogous one-to-one relationship between a parameter in LISA and $\beta_0^{(1)} - \beta_0^{(3)}$, but there is an implied constraint between $\beta_0^{(1)} - \beta_0^{(3)}$ and parameters in LISA. We can see this after expressing $\varphi_{31}(\boldsymbol{\gamma}; Y_1) = E(Y_3\, |R_2 = 0, Y_1) - E(Y_3 \mid R_2 = 1, Y_1)$ as a function of $\beta_0^{(1)} - \beta_0^{(3)}$.

$$\varphi_{31}(\boldsymbol{\gamma};Y_1) = E(Y_3\,|R_2=0,Y_1) - E(Y_3 \mid R_2=1,Y_1)$$

$$\begin{aligned}
&= \Big(\beta_0^{(1)} + \beta_2^{(1)}Y_2 + \beta_1^{(1)}Y_1\Big) \quad - \\
&\qquad \int_{Y_2} \Big(\beta_0^{(3)} + \beta_2^{(3)}Y_2 + \beta_1^{(3)}Y_1\Big) P(R_3=1 \mid R_2=1,Y_2,Y_1) + \\
&\qquad\quad \Big(\beta_0^{(2)} + \beta_2^{(2)}Y_2 + \beta_1^{(2)}Y_1\Big) P(R_3=0 \mid R_2=1,Y_2,Y_1) \qquad \mathrm{d}F(Y_2|R_2=1,Y_1)
\end{aligned}$$

Using (4.3) this becomes

$$\begin{aligned}
\varphi_{31}(\boldsymbol{\gamma};Y_1) &= \beta_0^{(1)} + \beta_2^{(1)}E\Big(Y_2 \mid R_2=0,Y_1\Big) + \beta_1^{(1)}Y_1 - \beta_0^{(3)}P(R_3=1 \mid R_2=1,Y_1) - \\
&\quad \beta_0^{(2)}P(R_3=0 \mid R_2=1,Y_1) - \beta_2^{(3)}E\Big(Y_2 \mid R_2=1,Y_1\Big) - \beta_1^{(3)}Y_1 \\
&= \beta_0^{(1)} + \beta_2^{(3)}\Big[E\Big(Y_2 \mid R_2=0,Y_1\Big) - E\Big(Y_2 \mid R_2=1,Y_1\Big)\Big] - \\
&\quad \beta_0^{(3)}\Big(1 - P(R_3=0 \mid R_2=1,Y_1)\Big) - \beta_0^{(2)}P(R_3=0 \mid R_2=1,Y_1) \\
&= (\beta_0^{(1)} - \beta_0^{(3)}) + \beta_2^{(3)}\Big[E\Big(Y_2 \mid R_2=0,Y_1\Big) - E\Big(Y_2 \mid R_2=1,Y_1\Big)\Big] - \\
&\quad (\beta_0^{(2)} - \beta_0^{(3)})P(R_3=0 \mid R_2=1,Y_1)
\end{aligned}$$

Remember that within the linear autoregressive class of models (to which DH and LISA both belong to) it holds that $b_{\Delta Y_3}^{(2)} + 1 = \beta_2^{(3)}$ where $b_{\Delta Y_3}^{(2)}$ is the coefficient from an autoregressive linear model $\omega_3(\overline{\mathbf{Y}}_2, R_3 = 1\,;\mathbf{b}_{\Delta Y_3})$ for the expected increment $\Delta Y_3 = Y_3 - Y_2$ among individuals present at time 3. Further we also know from Chapter 3 that for linear autoregressive models

$$\varphi_{31}(\boldsymbol{\gamma};Y_1) = \underbrace{E\Big(\Delta Y_3|R_2=0,Y_1\Big) - E\Big(\Delta Y_3|R_2=1,Y_1\Big)}_{\delta_{31}(\boldsymbol{\alpha};Y_1)} + \underbrace{E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)}_{\delta_{21}(\boldsymbol{\alpha};Y_1)}$$

Using this we can write

$$\varphi_{31}(\boldsymbol{\gamma};Y_1) = (\beta_0^{(1)} - \beta_0^{(3)}) + b^{(2)}_{\Delta Y_3}\Big[E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)\Big] - \underbrace{-(\beta_0^{(2)} - \beta_0^{(3)})P(R_3=0 \mid R_2=1,Y_1)}_{\delta_{31}(\boldsymbol{\alpha};Y_1)} + \underbrace{\Big[E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)\Big]}_{\delta_{21}(\boldsymbol{\alpha};Y_1)=\alpha_0^{(1)}-\alpha_0^{(\geq 2)}} \tag{4.5}$$

We can see that assuming constant, history-independent shifts for $E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big) = \alpha_0^{(1)} - \alpha_0^{(\geq 2)}$, $E(Y_3 \mid R_3=0,R_2=1,Y_2,Y_1) - E(Y_3 \mid R_3=1,R_2=1,Y_2,Y_1) = \beta_0^{(2)} - \beta_0^{(3)}$ and $E(Y_3 \mid R_2=0,Y_2,Y_1) - E(Y_3 \mid R_3=1,R_2=1,Y_2,Y_1) = \beta_0^{(1)} - \beta_0^{(3)}$ does not imply a constant shift $\delta_{31}(\boldsymbol{\alpha};Y_1)$ and/or $\varphi_{31}(\boldsymbol{\gamma};Y_1)$ since both are functions of $P(R_3=0 \mid R_2=1,Y_1)$. This probability is estimable from the observed data, but neither DH approach, nor LISA specify models for the dropout probability explicitly; therefore, we should compare LISA and DH parameters only for the unknown true value of $P(R_3=0 \mid R_2=1,Y_1)$. Nevertheless, (4.4) and/or (4.5) depend on $(\beta_0^{(1)} - \beta_0^{(3)})$ and we will see how LISA sets this parameter implicitly and how this choice can, later, in the light of *future independence* assumption, be interpreted. We can check from (4.4) and simulated data what LISA picks for $(\beta_0^{(1)} - \beta_0^{(3)})$. Notice also that this is not possible using (4.5), since in this case we would have to specify a model for $P(R_3=0 \mid R_2=1,Y_1)$ which would mean a) specifying part of the distribution of $(\overline{\mathbf{Y_3}},\overline{\mathbf{R_3}})$ that DH and LISA do not specify (this could be an issue in simulations; if (4.1) is used to simulate full data we would be agnostic to the correct model) and b) specifying a "gap" model for dropout, where the immediate previous outcome is not included, but the one before it, is. This is why we will use (4.4) in the next section to explain how to interpret the implicit choice for $(\beta_0^{(1)} - \beta_0^{(3)})$ in LISA.

4.5. $(\beta_0^{(1)} - \beta_0^{(3)})$ in LISA

We rewrite (4.4) as

$$E(Y_3) = \beta_0^{(3)} \ + \left(\beta_0^{(2)} - \beta_0^{(3)}\right) P(R_3 = 0) +$$

$$\beta_2^{(3)} \alpha_0^{(\geq 2)} + \beta_2^{(3)} \left(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\right) P(R_2 = 0) +$$

$$\beta_2^{(3)} \alpha_1^{(\geq 2)} E\Big(Y_1\Big) + \beta_1^{(3)} E\Big(Y_1\Big) + \left[\left(\beta_0^{(1)} - \beta_0^{(3)}\right) - \left(\beta_0^{(2)} - \beta_0^{(3)}\right)\right] P(R_2 = 0) \tag{4.6}$$

where we only use $P(R_3 = 0) = P(R_3 = 0, R_2 = 1) + P(R_2 = 0)$. The non-ignorable imputation algorithm used for LISA sets the last term in the above expression to 0 (or more accurately it assumes that $(\beta_0^{(1)} - \beta_0^{(3)}) = (\beta_0^{(2)} - \beta_0^{(3)})$). This means that in general choosing only $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ and $\delta_{32}(\boldsymbol{\alpha}; Y_2, Y_1)$ is not enough, except of course in the special case when $(\beta_0^{(1)} - \beta_0^{(3)}) = (\beta_0^{(2)} - \beta_0^{(3)})$. Bias of LISA in this case would be equal to the negative last term of (4.6). We can see that it is a linear function of the difference $\left(\beta_0^{(1)} - \beta_0^{(3)}\right) - \left(\beta_0^{(2)} - \beta_0^{(3)}\right)$ with the slope $P(R_2 = 0)$. We will see why, in the case of "future independence", we don't have to worry about this bias and what would be a reasonable approach to evaluate the robustness to this bias when it comes to conclusions about the dropout recovered from LISA. If LISA preserves interpretation and conclusions remain robust, it offers a useful extension of the DH approach to longer sequences of observations with only a few sensitivity parameters. We will check this robustness in simulations for T=3, as well as in data from already described cluster randomized multi-site trial evaluating the effect of financial incentives on patients ability to lower the level of low-density lipoprotein (LDL).

4.6. Future independence

For $T = 3$ (3.12) implies that $f(Y_3|R_2 = 0, Y_2, Y_1) = f(Y_3|R_2 = 1, Y_2, Y_1)$. This again has some implications within DH pattern mixture set up (4.1). If (3.12) holds then it follows that

$$E(Y_3 \mid R_2 = 1, Y_2, Y_1) = E(Y_3 \mid R_2 = 0, Y_2, Y_1)$$

This means that $E(Y_3 \mid R_2 = 0, Y_2, Y_1) = E(Y_3 \mid R_2 = 1, Y_2, Y_1) = \beta_0^{(1)} + \beta_2^{(1)} Y_2 + \beta_1^{(1)} Y_1$. In general it also holds that

$$
\begin{aligned}
E(Y_3 \mid R_2 = 1, Y_2, Y_1) = {} & E(Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1)\; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \\
& E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1)\; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \\
= {} & \left(\beta_0^{(2)} + \beta_2^{(2)} Y_2 + \beta_1^{(2)} Y_1\right) P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \\
& \left(\beta_0^{(3)} + \beta_2^{(3)} Y_2 + \beta_1^{(3)} Y_1\right) P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \\
= {} & \beta_0^{(2)} P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_0^{(3)} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) + \\
& \left(\beta_2^{(2)} P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_2^{(3)} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\right) Y_2 + \\
& \left(\beta_1^{(2)}\; P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_1^{(3)}\; P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\right) Y_1
\end{aligned}
$$

Thus, with future independence and DH pattern mixture parametrization (4.1) and (4.3) we have

$$
\beta_0^{(1)} = \beta_0^{(2)} P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_0^{(3)} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \tag{4.7}
$$

Note how the combination of DH parametrization (4.1) and (4.3) and future independence puts a constraint on $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$. The only values for which $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$ can remain unconstrained are $\beta_0^{(1)} = \beta_0^{(2)} = \beta_0^{(3)}$, otherwise (4.7) can hold only if $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$ is a constant function of Y_1 and Y_2. We will continue using (4.7) but this will be useful to keep in mind when we evaluate results from simulations.

Then if we write out $E(Y_3)$ again we have using the above

$$E(Y_3) = E\Big(E[Y_3 \mid Y_1]\Big)$$

$$= E\Big(E[Y_3 \mid R_2 = 1, Y_1]\, P(R_2 = 1 \mid Y_1) + E[Y_3 \mid R_2 = 0, Y_1]\, P(R_2 = 0 \mid Y_1)\Big)$$

$$= E\Big(E\Big[E(Y_3 \mid R_2 = 1, Y_2, Y_1) \mid R_2 = 1, Y_1\Big]\ P(R_2 = 1 \mid Y_1) +$$
$$E\Big[E(Y_3 \mid R_2 = 0, Y_2, Y_1) \mid R_2 = 0, Y_1\Big]\ P(R_2 = 0 \mid Y_1)\Big)$$

$$= E\Big(E\Big[(\beta_0^{(1)} + \beta_2^{(1)} Y_2 + \beta_1^{(1)} Y_1)\ \Big|\ R_2 = 1, Y_1\Big]\ P(R_2 = 1 \mid Y_1) +$$
$$E\Big[(\beta_0^{(1)} + \beta_2^{(1)} Y_2 + \beta_1^{(1)} Y_1)\ \Big|\ R_2 = 0, Y_1\Big]\ P(R_2 = 0 \mid Y_1)\Big)$$

use (4.3)

$$= E\Big(\Big(\beta_0^{(1)} + \beta_2^{(3)}\Big(\alpha_0^{(\geq 2)} + \alpha_1^{(\geq 2)} Y_1\Big) + \beta_1^{(3)} Y_1\Big)\ P(R_2 = 1 \mid Y_1) +$$
$$\Big(\beta_0^{(1)} + \beta_2^{(3)}\Big(\alpha_0^{(1)} + \alpha_1^{(\geq 2)} Y_1\Big) + \beta_1^{(3)} Y_1\Big)\ P(R_2 = 0 \mid Y_1)\Big)$$

use (4.7)

$$= E\Big(\Big[\beta_0^{(2)} P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_0^{(3)} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\Big] P(R_2 = 1 \mid Y_1)\Big) +$$

$$\beta_2^3 \alpha_0^{(\geq 2)} P(R_2 = 1) + \beta_2^3 \alpha_1^{(\geq 2)}\ E\Big(Y_1\ P(R_2 = 1 \mid Y_1)\Big) + \beta_1^3 E\Big(Y_1\ P(R_2 = 1 \mid Y_1)\Big) +$$

$$E\Big(\underbrace{\Big[\beta_0^{(2)} P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \beta_0^{(3)} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1)\Big] P(R_2 = 0 \mid Y_1)}_{=0 \text{ since } R_2 = 0 \text{ AND } R_2 = 1 \text{ can't be true}}\Big) +$$
$$\beta_2^3 \alpha_0^{(1)} P(R_2 = 0) + \beta_2^3 \alpha_1^{(\geq 2)}\ E\Big(Y_1\ P(R_2 = 0 \mid Y_1)\Big) + \beta_1^3 E\Big(Y_1\ P(R_2 = 0 \mid Y_1)\Big)$$

$$= \beta_0^{(2)} P(R_3 = 0, R_2 = 1) + \beta_0^{(3)} P(R_3 = 1, R_2 = 1) + \beta_2^3 \alpha_0^{(\geq 2)} \Big(1 - P(R_2 = 0)\Big) +$$

$$\beta_2^3 \alpha_1^{(\geq 2)}\, E\Big(Y_1\Big) + \beta_1^3 E\Big(Y_1\Big) + \beta_2^3 \alpha_0^{(1)} P(R_2 = 0)$$

$$= \beta_0^{(2)} P(R_3 = 0, R_2 = 1) + \beta_0^{(3)} P(R_3 = 1, R_2 = 1) +$$

$$\beta_2^3 \Big(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\Big) P(R_2 = 0) + \beta_2^3 \alpha_0^{(\geq 2)} +$$

$$\beta_2^3 \alpha_1^{(\geq 2)}\, E\Big(Y_1\Big) + \beta_1^{(3)} E\Big(Y_1\Big)$$

$$E(Y_3) = \beta_0^{(2)} P(R_3 = 0 \mid R_2 = 1) P(R_2 = 1) + \beta_0^{(3)} P(R_3 = 1 \mid R_2 = 1) P(R_2 = 1) +$$

$$\beta_2^{(3)} \Big(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\Big) P(R_2 = 0) + \beta_2^3 \alpha_0^{(\geq 2)} +$$

$$\beta_2^{(3)} \alpha_1^{(\geq 2)}\, E\Big(Y_1\Big) + \beta_1^{(3)} E\Big(Y_1\Big)$$

Constructing such correspondence is possible only if we remain within the confines of the linear autoregressive models for the conditional expectations. Using the fact that $P(R_3 = 0, R_2 = 1)$ in the expression for $E(Y_3)$

$$E(Y_3) = \Big[\beta_0^{(3)} + (\beta_0^{(2)} - \beta_0^{(3)})P(R_3 = 1 \mid R_2 = 1)\Big]P(R_2 = 1) +$$

$$\beta_2^{(3)}\Big(\alpha_0^{(1)} - \alpha_0^{(\geq 2)}\Big)P(R_2 = 0) + \beta_2^3\alpha_0^{(\geq 2)} +$$

$$\beta_2^{(3)}\alpha_1^{(\geq 2)}\, E\Big(Y_1\Big) + \beta_1^{(3)}E\Big(Y_1\Big) \tag{4.8}$$

You can see that compared to (4.4) the expression (4.8) has no part that depends on $\beta_0^{(1)} - \beta_0^{(3)}$. So additional assumption of "future independence" given the present makes this term obsolete for identification of $E(Y_3)$. Thus, in such a case for $T = 3$, the non-ignorable imputation algorithm presented in chapter 3 identifies $E(Y_3)$.

4.7. Simulations

Generating full data for which (3.12) holds is not trivial. Heuristically speaking, we need to allow conditional independence and/or lack thereof between Y_t and R_j, depending on the time lag between them. Because the collection and chronological order of structural equations for such structure are complex, we will instead generate full data using pattern mixture setup from (4.1). For simplicity we choose intercepts μ_1, $\alpha_0^{(\geq 2)}$ and $\beta_0^{(3)}$ to be zero; this way the sensitivity parameters $\beta_0^{(2)} - \beta_0^{(3)} = \delta_{32}(\boldsymbol{\alpha}; Y_2, Y_1)$, $\alpha_0^{(1)} - \alpha_0^{(\geq 2)} = \delta_{21}(\boldsymbol{\alpha}; Y_1)$ and $\beta_0^{(1)} - \beta_0^{(3)}$ depend only on $\beta_0^{(2)}$, $\alpha_0^{(1)}$ and $\beta_0^{(1)}$. These are the parameters we vary (by 1) between -5 and 5 each. Additionally, we generate data with each of 3 combinations of $P(R_2 = 0)$ and $P(R_3 = 0, R_2 = 1)$ $\{(0.1, 0.2), (0.2, 0.25), (0.25, 0.2)\}$. Slopes $\alpha_1^{(\geq 2)}$, $\beta_2^{(3)}$ and $\beta_1^{(3)}$ are chosen to be 1, 1, and 0.2 and variances σ_1, $\tau_2^{(\geq 2)}$, and $\tau_3^{(3)}$ are set to 1, 0.25, and 0.25. We generate 100 data sets with n=1000 observations for each of $11 \times 11 \times 11 \times 3 = 3993$ combinations of $\beta_0^{(2)}$, $\alpha_0^{(1)}$, and $\beta_0^{(1)}$ and probabilities of dropout. We picked variances so that a bias averaged over 100 data sets has variability small enough to facilitate illustration.

We will use expression (4.6) as a way to illustrate the bias of LISA which comes from setting $\beta_0^{(1)} - \beta_0^{(3)} = \beta_0^{(2)} - \beta_0^{(3)}$. Our hope is that this bias is acceptable w.r.t. the benefit of being able to extend DH pattern mixture sensitivity analysis using LISA to 3 or more observations. Keep also in mind that data generated with $\beta_0^{(1)} - \beta_0^{(3)} = \beta_0^{(2)} - \beta_0^{(3)} = 0$ and any value of $\alpha_0^{(1)} - \alpha_0^{(\geq 2)}$ and/or $P(R_2 = 0)$ and $P(R_3 = 0, R_2 = 1)$ will yield $E(Y_3 \mid R_2 = 1, Y_2, Y_1) = E(Y_3 \mid R_2 = 0, Y_2, Y_1)$ which is the only manifestation of future independence in the form of expectation constraint within data generated in this way.

Figure 4.1: Relationship between the bias in $\beta_0^{(1)} - \beta_0^{(3)}$ and $\beta_{\text{LI}}^{(3)}$

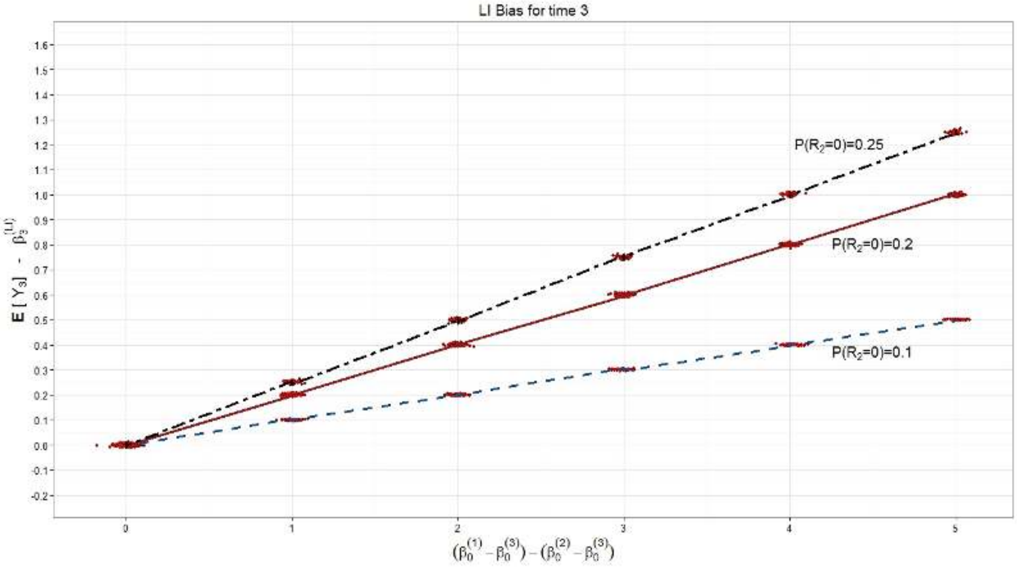


Figure 4.1 shows the dependency of the LI bias on the bias in $\beta_0^{(1)} - \beta_0^{(3)}$ for different values of $P(R_2 = 0)$. We plot this relationship for positive values of $\left(\beta_0^{(1)} - \beta_0^{(3)}\right) - \left(\beta_0^{(2)} - \beta_0^{(3)}\right)$ only. We can see that for $T = 3$ the slope of this linear relationship is captured by $P(R_2 = 0)$. Such a clear and unambiguous relationship is possible only for $T = 3$. For $T = 4$ we need to take into consideration that in this case we are dealing with a synergistic bias (see Appendix B.3). In general, it is possible to perceive the case in which a bias for $E(Y_3)$ (illustrated in Figure 4.1) is equal in size but of an opposite direction of the bias coming from setting $\gamma_0^{(1)} - \gamma_0^{(4)} = \gamma_0^{(2)} - \gamma_0^{(4)} = \gamma_0^{(3)} - \gamma_0^{(4)}$ (assume γ's are analog of β's for time 4, see Appendix B.3). We would then have an unbiased estimate of $E(Y_4)$. Even if we restrict ourselves only to those combinations for which $E(Y_3)$ is estimated unbiasedly, it would not be possible to illustrate relationship between LI bias for $T = 4$ in a similar

graph as in Figure 4.1 because we now have two sources of bias $\gamma_0^{(1)} - \gamma_0^{(4)}$ and $\gamma_0^{(2)} - \gamma_0^{(4)}$ instead of one.

Figure 1 shows that even for bias $\hat{\beta_0}^{(1)} - \hat{\beta}_0^{(3)} - \left(\beta_0^{(1)} - \beta_0^{(3)}\right)$ as big as 5 mg/dL and a marginal dropout proportion at time 2 of 0.25 the resulting shift in the marginal mean won't be bigger than 1.25. This is one advantage of using observed dropout rates, and can be interpreted as a non-parametric bound w.r.t. dropout probability, while only conditional expected increments remain prone to missspecification.

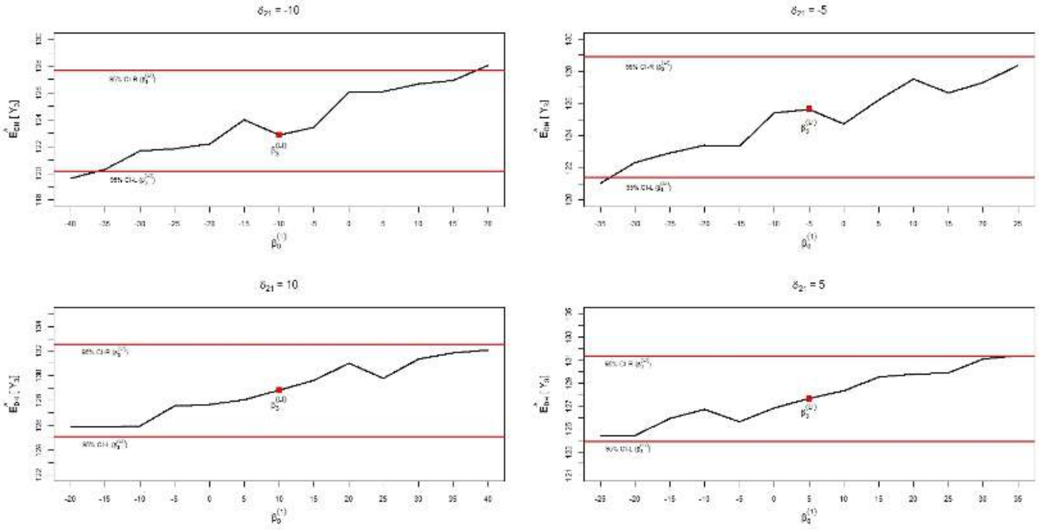
We should mention of course that in the case of the parameter combination that is implied by "future independence", LISA estimates unbiasedly $E(Y_3)$.

4.8. Real data

We saw how LISA in general "pays" for its reduction of sensitivity parameters for $T = 3$. In case of real data we are interested if this bias can considerably "sway" our LISA assessment about the gravity of the influence of non-ignorable dropout on inference. Our rationale for assessing the robustness of this decision is the following: find values of $\left(\beta_0^{(1)} - \beta_0^{(3)}\right) - \left(\beta_0^{(2)} - \beta_0^{(3)}\right)$ that take our LISA estimate of $E(Y_3)$ out of its 95% (bootstrap) CI. We can then evaluate what is the size of the bias in $\beta_0^{(1)} - \beta_0^{(3)}$ that takes our LISA estimated mean outside of the interval that we use to base our decision about the significance of the treatment effect (remember, we decide on significance of the treatment effect based on no overlap between 95% CI's for the mean in the treatment and control group). Thus, remaining within a 95% CI is an acceptable level of missestimation w.r.t. the decision making process about the sensitivity of the findings to non-ignorable dropout. This would mean that LISA preserves its interpretation and utility even when *future independence* is not true.

A modified dataset coming from a multi-center cluster-randomized controlled trial comparing three alternative economic interventions for reducing LDL cholesterol among patients with high cardiovascular risk will be used to asses the robustness of LISA conclusion. We use the first, third, and fifth observations and further modify any intermittent missingness into drop-out by leaving out any outcome recorded after the first occasion a subject has missed. We concentrate on the group with shared physician-patient incentives. This leaves us with observed $P(R_2 = 0) = 0.14$ and $P(R_3 = 0, R_2 = 1) = 0.06$. This means that the slope for the LI bias for $E(Y_3)$ is 0.14. Average width of the 95% CI interval for $E(Y_3)$ when $\beta_0^{(2)} - \beta_0^{(3)} = \alpha_0^{(1)} - \alpha_0^{(\geq 2)} \in [-10, 10]$ is $\approx$ 7.5. This means that for $\beta_0^{(2)} - \beta_0^{(3)} = c$ the difference $\left(\beta_0^{(1)} - \beta_0^{(3)}\right) - \left(\beta_0^{(2)} - \beta_0^{(3)}\right)$ necessary to take the $\hat{E}_{DH}(Y_3)$ outside of 95% CI calculated for β_3^{LI} is $3.75/P(R_2 = 0) = 3.75/0.14 \approx 27$. Figure 4.2 shows the change of $\hat{E}_{DH}(Y_3)$ w.r.t. $\beta_0^{(1)}$ for 4 different values of $\alpha_0^{(1)} - \alpha_0^{(\geq 2)} = \delta_{21}$.

Figure 4.2: Robustness of LISA w.r.t. DH approach in real data



As predicted the observed slope of the change is approximately equal to $P(R_2 = 0) = 0.14$. This means that the difference between δ_{32} and $\beta_0^{(1)} - \beta_0^{(3)}$ of 27 mg/dL is necessary to remove $\hat{E}_{DH}(Y_3)$ from the 95% CI for β_3^{LI}. This is 2.7 times larger than the absolute value of the largest shift considered in LISA of 10 mg/dL. Having this in mind we can say that for $T = 3$, LISA and DH sensitivity analysis would report same conclusion w.r.t. to the sensitivity of $E(Y_3)$ to non-ignorable dropout, if the difference between δ_{32} and $\beta_0^{(1)} - \beta_0^{(3)}$ is held within a realistic interval.

4.9. Discussion

In this paper we showed how LISA can be presented as an extension of DH pattern mixture approach to sensitivity analysis in a longitudinal clinical trial. In a general case for $T = 3$ we saw how the bias coming from parameter reduction in LISA w.r.t. DH can be evaluated and its influence judged w.r.t. to decision about the robustness of inference w.r.t. non-ignorable dropout. With $T = 3$ it is possible to write out a closed form linear relationship between LI bias and bias w.r.t. $\beta_0^{(1)} - \beta_0^{(3)}$ when data complies with DH pattern mixture distributional normality constraints. We concluded that the advantage that LISA brings in the form of parameter reduction and clarity of representation and interpretability of the results cannot be seriously jeopardized by the bias coming from implicit assumptions made by non-ignorable imputation algorithm. It is realistic to expect that the decision about robustness of the effect of economic interventions for reducing LDL cholesterol is not sensitive to this type of bias.

In the case of $T = 4$ it is harder to define such a bias. Incremental nature of the LI estimator allows for synergistic effects in this case when it comes to evaluating bias of LISA w.r.t. DH. It is possible to imagine the situation in which $\beta_0^{(1)} - \beta_0^{(3)}$ is not equal to $\beta_0^{(2)} - \beta_0^{(3)}$ (which would make β_3^{LI} biased for $E(Y_3)$) but which still help, together with φ_{32} and φ_{21}, implicitly set φ_{41} and φ_{42} (analogs of (4.5) for $T = 4$, see Appendix B.3) to correct values. One way to get around this ambiguity is to assume $E(Y_2)$ and $E(Y_3)$ are unbiasedly estimated by LISA and that the only bias for $E(Y_4)$ can come from setting $\gamma_0^{(1)} - \gamma_0^{(4)} = \gamma_0^{(2)} - \gamma_0^{(4)} = \gamma_0^{(3)} - \gamma_0^{(4)}$where γ_0's are analog intercepts within extended DH parametrization for $T = 4$.

In the special case of future independence we saw that the closed form for $E(Y_3)$ expressed as a function of DH parameters does not depend on $\beta_0^{(1)} - \beta_0^{(3)}$. This allows us to argue that LISA under such assumptions cannot misspecify $\beta_0^{(1)} - \beta_0^{(3)}$ and that accurate assumption about $\varphi_{32}(\delta_{32})$ and $\varphi_{21}(\delta_{21})$ is enough to accurately make a decision about the sensitivity of the treatment effect to non-ignorable drop out. In this case LISA solves the problem of informing parameters that reflect asynchronous association of Y_t and R_j as captured in φ_{tj} where $j < t - 1$.

LISA is a valuable addition to existent tools for sensitivity analysis w.r.t. to non-ignorable dropout in longitudinal clinical trials. Since we are using only marginal drop out rates and are not modeling probability of dropout, LISA could be a useful reference point to check the influence of different models for estimating the probability of non-ignorable dropout. Above and beyond that, LISA is a natural extension of DH framework to longer sequences of observations, while maintaining results of the sensitivity analysis interpretable and presentable in a compact and clear format.

CHAPTER 5

DISCUSSION

5.1. Recap

Our aim with this work was to warn against intricacies (of measure-theoretic nature) that crop up when a technique relying on random notion of time is applied in a setting like longitudinal clinical trial where randomness of observation times is hard to assume. This is more of a probabilistic issue, than statistical one and it has longer reaching consequences in situations where not all the data are observed. IIAs for inference about local characteristics of a “censored” stochastic process are given in Commenges et al., 2008. Authors describe IIA using the concept of the Radon-Nikodym derivative (in short this is a generalization of the concept of derivation for functions to an analog concept for measures, it can be understood as generalized likelihood ratio). In the causal setting, similar considerations though more rigorous from measure-theoretic aspect are given in Rø ysland, 2011 and Rø ysland, 2012 for continuous time marginal structural models (MSM). There, the author relates this IIA to Girsanov’s change of measure. This change of measure occurs between martingale measure of a hypothetical randomized trial (in the case of longitudinal data with dropout this would be counterfactual complete data generating distribution F_c) and an observational martingale measure (in our case this is replaced by “by dropout adulterated observed data distribution” F_o). In settings in which randomized trial measure is absolutely continuous with respect to the observational measure one is able to define a Radon-Nikodym derivative between these two measures. Absolute continuity between two measures is equivalent to solution of the stochastic differential equation, defined by this Radon-Nikodym derivative, being uniformly integrable. This is something we can conclude by comparing local characteristics of the observed (factual) and unobserved (counterfactual) outcome process. This comparison is of course in itself a hypothetical one, since by its definition underlying hypothetical randomized trial is unobservable, and it should be perceived only as a recipe. Such inability to devise formal tests for such a comparison of local characteristics using only observed data is a dynamic analog of the inability to test missing at random (MAR) vs. missing not at random (MNAR). Røysland points out that such generalized likelihood ratio process can be interpreted as a continuous time version of weights used for inverse probability weighted (IPW from now on) estimators in discrete time. This interpretation makes SEQ-MAR a natural analogy to the existence of Radon-Nikodym derivative between F_O and F_C. The inverse probability of observing weights serve as a type of an identification-providing sampling "bridge" between unobserved complete data distribution F_C and observed data distribution F_O. Heuristically, we can emulate sampling from a counterfactual distribution by weighting the sampling process from the observed distribution correctly (in the sense of unbiased estimation of the Radon-Nikodym derivative of the counterfactual w.r.t. observed measure). We should keep in mind that for this "bridge" to be sensical we need a) two clearly defined "shores", that is valid distributions (probability measures) and b) corresponding measures need to be related by absolute continuity, or, in terms of weights, probability of dropout needs to be estimable from the observed data. We could conclude that for random time (or stochastic process framework), exchangeability of local characteristics (instantaneous change) between adherers and dropouts is enough to define when dropout is ignorable.

If we are to use a method transplanted from such a setting as an identificational assistance w.r.t. marginal parameters (marginal mean) in a setting where observation times are not random but prespecified and in general known at the beginning of the study (as is the case for longitudinal clinical trial) we need to take caution. We showed that IIA restricted only to a local change is in general not enough to estimate/identify marginal (unconditional) parameters. In particular, demanding that mean increments are only exchangeable w.r.t. immediate history between dropouts and adherers would not offer us a general IIA as far reaching as necessary for identifying the marginal mean. Instead, as we saw on the example of E-SEQ-MAR we have to preserve exchangeability of increments w.r.t. all possible "gaps" between the time of dropout and time at which we are interested in estimating the mean. Therefore, we anchor the ignorable dropout w.r.t. LI at E-SEQ-MAR described by $T(T-1/2)$ constraints for $1 \leq j \leq t$, $2 \leq t \leq T$. We were further able to extend the described relationship between extended SWEEP and LI in the case of ignorable dropout, to non-ignorable dropout in a structured way. We suggested a statistically valid way to partition the shift in mean outcome (captured by functions φ's) into a sum of shifts in increments (δ's) that depend only on a contemporaneous relationship between the outcome and the dropout processes. To preserve the interpretability we confine our approach to linear, autoregressive ω's and a constant $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ sensitivity parameter. Our approach offers one solution to a prominent problem of time-shifted association between continuous outcome and the dropout process encountered inevitably in sensitivity analysis for longitudinal data. Experts input for LISA is confined to informing those parameters for which experts are realistically expected to be able to come up with the plausible values. We define a plausible constraint on F_C in the form of future independence under which aggregation of the parameters as introduced by LISA does not impair the identification of the marginal mean for any Y_t $t < T$. We showed how LISA can be perceived as an extension of a pattern mixture approach by Daniels and Hogan. If our approach is to be used instead of a classical pattern mixture approach for longer sequences of observations we offered in chapter 4 a way to evaluate a possible discrepancy between by LISA and DH approach when it comes to conclusions about the robustness of the findings to non-ignorable dropout. We could conclude that bias of LISA w.r.t. DH is a controlled one and in principle only existent when future independence cannot be assumed. This makes LISA a useful and viable extension of [previously introduced pattern mixture paradigm for sensitivity analysis to non-ignorable dropout in longitudinal data.

5.2. Future directions

Possible extensions and improvements of our approach are possible in few directions. Most immediate and desirable ones are: allowing $\delta_{21}(\boldsymbol{\alpha}; Y_1)$ to depend on the last observed value, closed expression for the variance of $\hat{\beta}_{\mathrm{LI}_t}$, defining a formal consistent way of comparing the conclusions from LISA and method introduced in Scharfstein et al., 2014.

Often there are parts of the domain Ω_{Y_j} of the clinical outcome Y_j, ($j \in \{1, \ldots, T\}$) for which the dropout behavior is more homogenous in general as well as reflected by the introduced incremental shift. This is aligned with the floor and ceiling characteristic of clinical tests. Patients on the extremes of the scale with very protective/harmful values of the outcome under study will show less differential drop out behavior than those who experience same absolute difference in the mean but in the “normal” part of the domain Ω_{Y_j}. One way to reflect this clinical reality is to assume an attenuated shift in the conditional mean within those more inertial areas of the outcome domain. In our approach it is possible to implement a piecewise constant shift within three separate areas of the Ω_{Y_j}: c_1 within the “average” part of Ω_{Y_j} and c_2 in each of the two margin areas, where c_2 is defined as some variable fraction of c_1. Technically this is easy to implement, though in order to claim the unconditional interpretation of the estimated mean we have to impose more constraints on F_C above and beyond future ignorability. Proving this might get technically very complex even on the level of a heuristic argument.

It would be very useful and practical to have a closed form expression for the variance of $\hat{\beta}_{\mathsf{LI}_t}$ for each value of $\delta_{21}(\boldsymbol{\alpha}; Y_1)$. We could save ourselves the trouble and time to program and run bootstrap procedure each time we want to do implement LISA. This is memory as well as time consuming effort that we have to implement each time. In Robins, Rotnitzky, and Zhao, 1995 a recursive expression was given for the variance of the extended SWEEP estimator and the starting point would be to investigate the possibilities to adapt this recursive formula to the case of non-ignorable dropout where $\delta_{21}(\boldsymbol{\alpha}; Y_1) \neq 0$ governs the shifts.

An analogous way to subject IPW estimator from Table 3.2 to a test of sensitivity to non-ignorable dropout could be to use a non-ignorable version of the logit model in which coefficient next to the outcome contemporaneous with the dropout serves as a sensitivity parameter. This is a similar, though oversimplifying route compared to the one from Scharfstein et al., 2014. Relating the value of that coefficient in such a non-ignorable version of a logit model to a value of the shift $\delta_{2,1} \neq 0$ for which $E(\overline{Y}_t \mid X = 1) - E(\overline{Y}_t \mid X = 0)$ yields a comparable estimate is, without assuming the full joint distribution for the complete data, not possible. Additionally, this discrepancy extends to the individual level data as well. Any model used for estimating probability of dropout in IPW estimator will not lead to extrapolating outside of observed data. On the other side, even for $\delta_{2,1} = 0$ the assumed mean model for observed ΔY_t can imply prediction for Y_t based on some value of Y_{t-1} among dropouts that is outside of the range of observed Y_t's. In principle, correspondence between the shift in the mean at time t and the influence of the contemporaneous outcome Y_t on the log-odds scale on dropout can be established if we assume some parametric form for the joint distribution of complete data. What we can do is a post-hoc estimation of a non-ignorable logit model, after we impute the data according to the assumed shift $\delta_{2,1}$ as described for the non-ignorable imputation algorithm.

APPENDIX A

STOCHASTIC PROCESSES AND MODES OF DROPOUT IN LONGITUDINAL DATA

A.1. Some basic concepts from stochastic processes

A very nice historical perspective of application of martingales in survival analysis is given by Aalen et al., 2009 while a comprehensive and more technical description of the rigorous development of this technique can be found in Fleming and Harrington, 2011 monograph. The notation and concepts we will use will be either directly taken from this book or will be an extension thereof.

Definition 1. A (real-valued) stochastic process is a family of random variables $X = \{X(t) : t \in \Gamma\}$ indexed by a set Γ, all defined on the same probability space $(\Omega, \mathcal{F}, P)$

Γ will be either $[0, \infty]$ for a continuous time stochastic process or $\{0, 1, 2, \dots\}$ for a discrete time one. For a stochastic process X, the (random) functions $X(\cdot, \omega) : R^+ \to R, \omega \in \Omega$ are called the sample paths or trajectories of X. A very important notion in our conceptual developments will be the ability of somehow quantifying the difference in information accrual for two processes. Therefore we need to somehow formulate the concept of information accruing over time.

Definition 2.

1. A family of sub-σ-algebras $\{\mathcal{F}_t : t \geq 0\}$ of a σ-algebra $\mathcal{F}$ is called increasing if $s \leq t$ implies $\mathcal{F}_s \subset \mathcal{F}_t$ (i. e. if for $s \leq t$, $A \in \mathcal{F}_s$ implies $A \in \mathcal{F}_t$). An increasing family of sub-σ-algebras is called a *filtration*.

2. When $\{\mathcal{F}_t : t \geq 0\}$ is a filtration, the σ-algebra $\cup_{h>0} \mathcal{F}_{t+h}$ is usually denoted by $\mathcal{F}_{t+}$. The corresponding limit from the left, $\mathcal{F}_{t-}$, is the smallest σ-algebra containing all the sets in $\cup_{h>0} \mathcal{F}_{t-h}$ and is written $\sigma\{\cup_{h>0} \mathcal{F}_{t-h}\}$ or $\bigvee_{h>0} \mathcal{F}_{t-h}$

3. A filtration $\{\mathcal{F}_t : t \geq 0\}$ is right-continuous if, for any t, $\mathcal{F}_{t+} = \mathcal{F}_t$

4. A stochastic basis is a probability space $(\Omega, \mathcal{F}, P)$ equipped with a right-continuous filtration $\{\mathcal{F}_t : t \geq 0\}$ and is denoted by $(\Omega, \mathcal{F}, \{\mathcal{F}_t : t \geq 0\}, P)$.

5. A stochastic basis is called complete if $\mathcal{F}$ contains any subset of a P-null set (so $\mathcal{F}$) is complete and if each $\mathcal{F}_t$ contains all P-null sets of $\mathcal{F}$

The most natural filtrations are *histories* of stochastic processes, or families with $\mathcal{F}_t = \sigma\{X(s) : 0, s \leq t\}$, the smallest σ-algebra with respect to which each of the variables $X(s),\ 0 \leq s \leq t$ is measurable. In this case, $\mathcal{F}_t$ "contains the information" generated by the process X on $[0, t]$.

Definition 3. A stochastic process $\{X(t) : t \geq 0\}$ is adapted to a filtration if, for every $t \geq 0$, $X(t)$ is $\mathcal{F}_t$-measurable.

Any process is adapted to its history.

Conditioning on the path up to time t of a process X is
conditioning on the σ-algebra $\sigma\{X(u) : 0 \leq u \leq t\}$. We proceed to define conditional expectation with respect to an arbitrary σ-algebra.

Definition 4. Suppose Y is a random variable on a probability space $(\Omega, \mathcal{F}, P)$ and let $\mathcal{G}$ be a sub-σ-algebra of $\mathcal{F}$. Let X be a random variable satisfying

1. X is $\mathcal{G}$-measurable; and

2. $\int_B Y \, \mathrm{d}P = \int_B X \, \mathrm{d}P$ for all subsets $B \in \mathcal{G}$

The variable X is called the *conditional expectation* of Y given $\mathcal{G}$, and is denoted by $E(Y|\mathcal{G})$.

It is a standard result that $E(Y|\mathcal{G})$ exists when $E|Y| < \infty$. Also, if W and X are two variables satisfying (1) and (2) above, W and X are equivalent. If $A \in \mathcal{F}$ is an event, then by $P(A|\mathcal{G})$ we mean $E(I_A|\mathcal{G})$.

We will use following properties of conditional expectation as needed. Let $(\Omega, \mathcal{F}, P)$ be an arbitrary probability space, X and Y random variables on this space, and $\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$ sub-σ-algebras

1. if $\mathcal{F}_t = \{\emptyset, \Omega\}$, $E(X|\mathcal{F}_t) = E(X)$ a.s.

2. $E\Big[E(X|\mathcal{F}_t)\Big] = E(X)$.

3. If $\mathcal{F}_s \subset \mathcal{F}_t$, then $E\Big[E(X|\mathcal{F}_s)|\mathcal{F}_t\Big] = E\Big[E(X|\mathcal{F}_t)|\mathcal{F}_s\Big] = E(X|\mathcal{F}_s)$ a.s.

4. if $\sigma\{Y\} \subset \mathcal{F}_t$ (i.e. if Y is $\mathcal{F}_t$-measurable) then $E(XY|\mathcal{F}_t) = YE(X|\mathcal{F}_t)$ a.s. This immediately implies $E(Y|\mathcal{F}_t) = E(Y)$ a.s.

5 Let X and Y be independent random variables, and let $\mathcal{G} = \sigma(X)$. Then $E(Y|\mathcal{G}) = E(Y)$ a.s.

If $\mathcal{A}$ is any collection of random variables, $E(Y|\mathcal{A})$ will denote $E(Y|\sigma(\mathcal{A}))$.

We will conclude this section with definitions of a counting process and a martingale process.

Definition 5. A *counting process* is a stochastic process $\{N(t) : t \geq 0\}$ adapted to a filtration $\{\mathcal{F}_t : t \geq 0\}$ with $N(0) = 0$ and $N(t) < \infty$ a.s., and whose paths are with probability one right-continuous, piecewise constant, and have only jump discontinuities, with jumps of size +1.

Definition 6. Let $\{X(t) : t \geq 0\}$ be a right continuous stochastic process with left hand limits and $\{\mathcal{F}_t : t \geq 0\}$ a filtration, defined on a common probability space. X is called a *martingale* with respect to $\mathcal{F}_t$ if

1. X is adapted to $\{\mathcal{F}_t : t \geq 0\}$
2. $E|X(t)| < \infty$ for all $t < \infty$
3. $E\{X(t+s)|\mathcal{F}_t\} = X(t)$ a.s. for all $s \geq 0$, $t \geq 0$

If we substitute $=$ by $\geq$, $\leq$ in (3) we get the definition of a sub-, super-martingale respectively.

A very common technique in dealing with general stochastic processes is to break them down into separate martingale and drift terms. This is a powerful technique underlying a lot of martingale methods in survival analysis. It is simply described in the case in which $\{X_t\}_{t=0,1,\ldots}$ is a stochastic process adapted to the discrete-time filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t=0,1,\ldots}, P)$. If X is integrable, then it is possible to decompose it into the sum of a martingale M and another process A. The process A which starts from zero is such that A_t is $\mathcal{F}_{t-1}$-measurable (predictable) for each $t \geq 1$. Due to M being a martingale we have the identity

$$A_t - A_{t-1} = E\Big[A_t - A_{t-1}|\mathcal{F}_{t-1}\Big] = E\Big[X_t - X_{t-1}|\mathcal{F}_{t-1}\Big] \tag{A.1}$$

The first equality follows from the fact that A_t is $\mathcal{F}_{t-1}$-measurable, and the second by the martingale characteristic of the process $\{M_t\}_{t=0,1,\ldots}$.

So, A is uniquely defined by

$$A_t = \sum_{k=1}^{t} E\Big[X_k - X_{k-1} | \mathcal{F}_{k-1}\Big] \tag{A.2}$$

and is referred to as the compensator of X. This decomposition was in fact first proven for discrete time stochastic processes and is named *Doob decomposition* after Joseph L. Doob.

A.2. Positivity and traditional dropout modes: independent censoring, MAR and SEQ-MAR

For a detailed treatment of (observed) positivity assumption we refer to Laan and Rose, 2011. Here we will show a notational definition and offer one interpretation of it. Positivity is a condition coded as

$$P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1}) \geq \sigma > 0$$

for $t = 2, \ldots, T$. This is also called strong or theoretical positivity. In a finite sample one refers to observed positivity if

$$\hat{P}(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1}) \geq \sigma > 0$$

For IPW GEE estimators lack of observed positivity is one factor that can introduce considerable bias even when the model for $P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1})$ is correctly specified. Heuristically, positivity ensures that every subject i or more accurately, every realization of a covariate history $\overline{\mathbf{W}}_{t-1}$ that is possible to perceive under F_C is possible to observe under F_O. In other words any covariate pattern has a positive probability of being observed during the whole course of a study.

The following is a formulation of the MAR assumption for longitudinal studies in which missingness is monotone.

$$P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_T) = P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1})$$

Conditioning on the left hand side entails $\overline{W}_T$, which constitutes the observed and the counterfactual part after the possible drop-out. Conditioning on the right hand side is done only on observables. This is what we refer to as *conditioning on observables* within the identifiability assumption. The other characteristic is the non-parametric nature of the assumption. At this level the conditional probability is just a conditional expectation of the indicator $I_{\{R_t=1\}}$ given appropriate sigma algebras generated by different histories of the processes $\mathbf{W}_t$ and R_t. These are, per definition, completely unconstrained, (square) integrable functional forms.

The SEQ-MAR assumption is formalized as

$$P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1}, \overline{Y}_T) = P(R_t = 1 \mid R_{t-1} = 1, \overline{\mathbf{W}}_{t-1})$$

(notice above, that we double counted/denoted $\{Y_1, Y_2, \ldots, Y_{t-1}\}$, once in $\overline{\mathbf{W}}_{t-1}$ and a second time in $\overline{Y}_T$). We can again see both characteristics of an identifiability assumption. The right hand side conditions only on observables and at the same time the formulation leaves the functional form of the conditional expectation (conditional probability) completely unspecified. It is perhaps useful to point out that in a special case with explicitly monotone drop out where at each measurement time t only the response Y is recorded without vector $\mathbf{V}$, SEQ-MAR and MAR coincide.

In survival analysis the independent censoring assumption is an identifiability assumption that allows valid inference on the population parameters (i.e., the intensity of an event process) using a right censored sample. Right censoring in this case incurs a type of a selection bias in the sample. Assuming independent censoring renders data from uncensored survivors beyond time point t adequate for a consistent estimation of the intensity of the underlying event process for each t.

The definition of independent censoring in survival analysis is closely related to the Doob-Meyer decomposition for submartingales in continuous time. We take the formulation of the condition for absolutely continuous failure time from Fleming and Harrington, 2011. Let T be an absolutely continuous failure time random variable and U the censoring variable. Define $X = min(T, U)$, $\delta = I_{\{T \leq U\}}$, and let λ denote the hazard function for T. Define

$$\begin{aligned} N(t) &= I_{\{X \leq T, \delta = 1\}} \\ N^U(t) &= I_{\{X \leq T, \delta = 0\}} \\ \mathcal{F}_t &= \sigma\{N(u), N^U(u) : 0 \leq u \leq t\}. \end{aligned}$$

Then the process M given by

$$M(t) = N(t) - \int_0^t I_{\{X \geq u\}} \lambda(u) \, \mathrm{d}u$$

is an $\mathcal{F}_t$-martingale if and only if

$$\lambda(t) = \frac{\frac{-\partial}{\partial u} P(T \geq u, U \geq T)|_{u=t}}{P(T \geq t, U \geq T)} \text{ whenever } P(X > t) > 0. \tag{A.3}$$

If we denote the right-hand side of $(A.3)$ by $\lambda^{\#}(t)$ then $\lambda(t)$ is the underlying intensity (hazard) of the failure time variable T while $\lambda^{\#}(t)$ can be written as

$$\begin{aligned} \frac{\frac{-\partial}{\partial u} P(T \geq u, U \geq T)|_{u=t}}{P(T \geq t, U \geq T)} &= \lim_{h \to 0} \frac{1}{h} \frac{P(t \leq T \leq t + h, U \geq t)}{P(T \geq t, U \geq t)} \\ &= \lim_{h \to 0} \frac{1}{h} P(t \leq T \leq t + h \mid T \geq t, U \geq t). \end{aligned} \tag{A.4}$$

As we can see in (A.4) conditioning is on observables, i.e., those subjects still in the risk set. The above statement is again made on a philosophical level (no functional form is assumed for $\lambda(t)$) and in short claims that *iff* the net hazard λ and the crude hazard $\lambda^{\#}(t)$ are the same we can render the residual process a martingale by projecting the predictable part of the process $N(t)$ only on observable histories. The proof can be found in Fleming and Harrington, 2011.

APPENDIX B

B.1. E-SEQ-MAR and LI in the light of positivity assumption

We illustrate for $T = 3$ the relationship between (2.5) and (2.4) in the light of the positivity assumption (see Laan and Rose, 2011) within the class of linear autoregressive models. Let (2.5) hold for $T = 3$

$$\begin{aligned} E[\Delta Y_2 \mid R_2 = 1, Y_1] &= E[\Delta Y_2 \mid Y_1, R_1 = 1] \\ E[\Delta Y_3 \mid R_3 = 1, Y_2, Y_1] &= E[\Delta Y_3 \mid Y_2, Y_1, R_2 = 1] \end{aligned} \tag{B.1}$$

and let

$$\begin{aligned} E(Y_3 \mid Y_2, Y_1, R_3 = 1) &= b^{01}_{\Delta Y_3} + (b^{21}_{\Delta Y_3} + 1)Y_2 + b^{11}_{\Delta Y_3} Y_1 \\ E(Y_3 | Y_2, Y_1, R_2 = 1, R_3 = 0) &= b^{00}_{\Delta Y_3} + (b^{20}_{\Delta Y_3} + 1)Y_2 + b^{10}_{\Delta Y_3} Y_1 \end{aligned}$$

For equivalence of (2.5) and (2.4) we are missing $E(Y_3 \mid R_2 = 1, Y_1) = E(Y_3 \mid R_1 = 1, Y_1)$. Then,

$$\begin{aligned} E(Y_3 \mid R_2 = 1, Y_1) &= E\Big(E(Y_3 \mid R_2 = 1, Y_2, Y_1) \mid R_2 = 1, Y_1\Big) \\ &= E\Big(b^{01}_{\Delta Y_3} + (b^{21}_{\Delta Y_3} + 1)Y_2 + b^{11}_{\Delta Y_3} Y_1 \mid R_2 = 1, Y_1\Big) \\ &= b^{01}_{\Delta Y_3} + (b^{21}_{\Delta Y_3} + 1)E\Big(Y_2 \mid R_2 = 1, Y_1\Big) + b^{11}_{\Delta Y_3} Y_1 \end{aligned} \tag{B.2}$$

where for second equality we used second row of (B.1).

It also holds that,

$$
\begin{aligned}
E(Y_3 \mid R_1 = 1, Y_1) &= E(Y_3 \mid R_2 = 1, R_1 = 1, Y_1)\; P(R_2 = 1 \mid R_1 = 1, Y_1) + \\
&\quad E(Y_3 \mid R_2 = 0, Y_2, Y_1)\; P(R_2 = 0 \mid R_1 = 1, Y_1) \\
&= E\Big(E(Y_3 \mid R_2 = 1, Y_2, Y_1) \mid R_2 = 1, Y_1\Big)\; P(R_2 = 1 \mid R_1 = 1, Y_1) + \\
&\quad E\Big(E(Y_3 \mid R_2 = 0, Y_2, Y_1) \mid R_2 = 0, Y_1\Big)\; P(R_2 = 0 \mid R_1 = 1, Y_1) \\[2ex]
&= E\Big(b^{01}_{\Delta Y_3} + (b^{21}_{\Delta Y_3} + 1)Y_2 + b^{11}_{\Delta Y_3} Y_1 \Big| R_2 = 1, Y_1\Big) P(R_2 = 1 \mid R_1 = 1, Y_1) + \\
&\quad E\Big(b^{01}_{\Delta Y_3} + (b^{21}_{\Delta Y_3} + 1)Y_2 + b^{11}_{\Delta Y_3} Y_1 \Big| R_2 = 0, Y_1\Big) P(R_2 = 0 \mid R_1 = 1, Y_1) \\[2ex]
&= b^{01}_{\Delta Y_3} P(R_2 = 1 \mid R_1 = 1, Y_1) + b^{00}_{\Delta Y_3} P(R_2 = 0 \mid R_1 = 1, Y_1) + \\[2ex]
&\quad \Big[(b^{21}_{\Delta Y_3} + 1) E\Big(Y_2 \mid R_2 = 1, Y_1\Big) P(R_2 = 1 \mid R_1 = 1, Y_1) + \\
&\quad\; (b^{20}_{\Delta Y_3} + 1) E\Big(Y_2 \mid R_2 = 0, Y_1\Big) P(R_2 = 0 \mid R_1 = 1, Y_1)\; \Big] + \\[2ex]
&\quad \Big[b^{11}_{\Delta Y_3} P(R_2 = 1 \mid R_1 = 1, Y_1) + b^{10}_{\Delta Y_3} P(R_2 = 0 \mid R_1 = 1, Y_1)\Big] Y_1
\end{aligned}
\tag{B.3}
$$

We can discuss equality of (B.2) and (B.3) in 2 cases: when positivity

$$P(R_{i2} = 1 \mid R_{i1} = 1, Y_{i1}) > 0$$

holds for any individual i or when the observed version of this assumption is not reflected in the data.

If we can assume that positivity holds in our data,

$$E[Y_{i,\,2} \mid R_{i,\,1} = 1, Y_{i,\,1}] = E[Y_{i,\,2} \mid R_{i,\,2} = 1, R_{i,\,2} = 1, Y_{i,\,1}]$$

(first row of (B.1)) is equivalent to

$$E[Y_{i,\,2} \mid R_{i,\,2} = 0, R_{i,\,1} = 1, Y_{i,\,1}] = E[Y_{i,\,3} \mid R_{i,\,2} = 1, R_{i,\,1} = 1, Y_{i,\,1}]$$

because

$$\begin{aligned} E[Y_{i,\,2} \mid R_{i,\,1} = 1, Y_{i,\,1}] = {} & E[Y_{i,\,2} \mid R_{i,\,2} = 1, R_{i,\,1} = 1, Y_{i,\,1}]\, P(R_{i,\,2} = 1 \mid R_{i,\,1} = 1, Y_{i,\,1}) + \\ & E[Y_{i,\,2} \mid R_{i,\,2} = 0, R_{i,\,1} = 1, Y_{i,\,1}]\, P(R_{i,\,2} = 0 \mid R_{i,\,1} = 1, Y_{i,\,1}) \end{aligned}$$

Thus in this case ($E(Y_2 \mid R_2 = 0, Y_1) = E(Y_2 \mid R_2 = 1, Y_1)$) we can write (B.3) as

$$\begin{aligned} E(Y_3 \mid R_1 = 1, Y_1) = {} & b^{01}_{\Delta Y_3} P(R_2 = 1 \mid R_1 = 1, Y_1) + b^{00}_{\Delta Y_3} P(R_2 = 0 \mid R_1 = 1, Y_1) + \\ & \Big[(b^{21}_{\Delta Y_3} + 1) P(R_2 = 1 \mid R_1 = 1, Y_1) + \\ & \quad (b^{20}_{\Delta Y_3} + 1) P(R_2 = 0 \mid R_1 = 1, Y_1) \ \Big] E\Big(Y_2 \mid R_2 = 1, Y_1 \Big) + \\ & \Big[b^{11}_{\Delta Y_3} P(R_2 = 1 \mid R_1 = 1, Y_1) + b^{10}_{\Delta Y_3} P(R_2 = 0 \mid R_1 = 1, Y_1) \Big] Y_1 \end{aligned} \tag{B.4}$$

Then, equality of (B.2) and (B.3) is equivalent to the equality of the coefficients

$$\begin{aligned} b^{01}_{\Delta Y_3} &= b^{00}_{\Delta Y_3} \\ b^{21}_{\Delta Y_3} &= b^{20}_{\Delta Y_3} \\ b^{11}_{\Delta Y_3} &= b^{10}_{\Delta Y_3} \end{aligned}$$

regardless of the functional form of $P(R_2 = 1 \mid R_1 = 1, Y_1)$. This equality of the coefficients can be perceived as one form of congeniality of implied models η_{22}, η_{33} and η_{23}. Positivity ensures enough structure for the relationship between $F_C(\Theta_c)$ and $F_O(\Theta_o)$ that we are able to talk about identification as a separate concept from specification. In this regard the positivity assumption suffices for continuous longitudinal (panel) data. Analogous prerequisites, though more rigorous from measure-theoretic aspect, are given in (Rø ysland, 2011) for continuous time marginal structural models (MSM).

If positivity is not present in our data the equality of (B.2) and (B.3) cannot be reduced to such "coefficientwise" equality because we cannot follow that $E(Y_2 \mid R_2 = 0, Y_1) = E(Y_2 \mid R_2 = 1, Y_1)$ for all i. Namely for those individuals without a positive probability to be observed at time 2 ($P(R_{i2} = 1 \mid R_{i1} = 1, y_{i1}) = 0$), we cannot establish the analogous equivalence since $R_{i,\,2} = 1$ is a null set. Individuals with $Y_{i1} = y_{i1}$ have no "representative" in any of the observed data after time 2, so when we extrapolate beyond observed data for them by using $(B.1)$ we cannot be wrong, which is when specifying a model becomes enough for identification. At this point congeniality of models $\eta_{t'\,t}$ for $1 \leq t \leq T$ and $1 \leq t^{'} \leq t+1$ (as discussed in section (2.2.2)) takes the role that positivity has w.r.t. estimators using the inverse probability of observing. Namely, positivity defines the condition under which an unambiguous notion of a correct model for probability of observing an individual at each time t can actually exist. Without it, we don't have a reference model to conceptualize the notion of correctness of any model for probability of dropout we might specify. It is known that in addition to inflated variance IPW EE estimator can exhibit considerable bias even when true inverse probability weights are used, if there is no observed positivity in the data (see Laan and Rose, 2011). Since estimators like LI don't rely on a model for dropout, congeniality of models chosen to make increments exchangeable between adherers and dropouts at each time has a similar role of defining a notion of a correct outcome model. In the case when $\eta_{t't}$ for $1 \leq t \leq T$ and $1 \leq t^{'} \leq t+1$ are congenial, existence of some/any $F_C(\Theta_c)$ is not in question, so it is possible to be correct or wrong w.r.t. this true but unknown distribution. If implied $\eta_{t't}$ for $1 \leq t \leq T$ and $1 \leq t^{'} \leq t+1$ are not congenial in the sense that no $F_C(\Theta_c)$ can accommodate all the constraints simultaneously, then we are left with all time specific constraints individually being correct, but no single individual for which all of them can hold simultaneously. Identification through specification becomes problematic for LI because of the inability to "skip" implicit specification of all $\eta_{t't}$ for $1 \leq t \leq T$ and $1 \leq t^{'} \leq t+1$. As we saw, for identification of $E(Y_T|X)$ it suffices to specify only $T-1$ $\eta_{t'T}(\overline{\mathbf{Y}}_{t'-1}\,; \theta_{t'T})$ for $t^{'} = T+1, \ldots, 1$. When we impose $T(T-1)/2$ constraints we make congeniality, at best, equally plausible than when we deal with only $T-1$ constraints.

B.2. LISA under future independence for T=3

We show that the form of the marginal mean $E(Y_3)$ under assumptions implied by non-ignorable imputation algorithm and future independence has no term depending on $E(Y_3 \mid R_2 = 0, Y_2, Y_1) - E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1)$. Non ignorable imputation algorithm implicitly assumes for $T = 3$ that

$$\begin{aligned}
Y_2|Y_1, R_2 = 1 &\sim N\Big(b^0_{\Delta Y_2} + (b^1_{\Delta Y_2} + 1)Y_1, \quad \sigma_{\Delta Y_2}\Big) \\
Y_2|Y_1, R_2 = 0 &\sim N\Big(\alpha + b^0_{\Delta Y_2} + (b^1_{\Delta Y_2} + 1)Y_1, \quad \sigma_{\Delta Y_2}\Big) \\
Y_3|Y_2, Y_1, R_3 = 1 &\sim N\Big(b^0_{\Delta Y_3} + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3}Y_1, \quad \sigma_{\Delta Y_3}\Big) \\
Y_3|Y_2, Y_1, R_2 = 1, R_3 = 0 &\sim N\Big((\rho\alpha + b^0_{\Delta Y_3}) + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3}Y_1, \quad \sigma_{\Delta Y_3}\Big) \qquad \text{(B.5)}
\end{aligned}$$

Further let

$$Y_t \perp R_j \mid \overline{\mathbf{Y}}_j, \text{ for } 1 \leq j < t, 3 \leq t \leq T \qquad . \qquad \text{(B.6)}$$

For $T = 3$ (B.6) implies that $f(Y_3|R_2 = 0, Y_2, Y_1) = f(Y_3|R_2 = 1, Y_2, Y_1)$ (notice that this is $\neq$

SEQ-MAR (see Robins, Rotnitzky, and Zhao (1995)). This again has a following implication on the conditional expectations

$$E(Y_3 \mid R_2 = 1, Y_2, Y_1) = E(Y_3 \mid R_2 = 0, Y_2, Y_1)$$

Thus, for linear autoregressive class of models, $E(\Delta Y_3 \mid R_2 = 0, Y_2, Y_1) = E(\Delta Y_3 \mid R_2 = 1, Y_2, Y_1)$. In general it always holds that

$$\begin{aligned}
E(Y_3 \mid R_2 = 1, Y_2, Y_1) &= E(Y_3 \mid R_3 = 0, R_2 = 1, Y_2, Y_1)\, P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \\
&\quad E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1)\, P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \\
&= \left(b^0_{\Delta Y_3} + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3} Y_1\right) P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + \\
&\quad \left((\rho\alpha + b^0_{\Delta Y_3}) + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3} Y_1\right) P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \\
&= (\rho\alpha + b^0_{\Delta Y_3}) P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + b^0_{\Delta Y_3} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) + \\
&\quad (b^2_{\Delta Y_3} + 1)\, Y_2 \; + b^1_{\Delta Y_3}\, Y_1
\end{aligned}$$

Thus, with (B.5) and (B.6), $b^{0_2}_{\Delta Y_3}$ the intercept of $E(\Delta Y_3 \mid R_2 = 0, Y_2, Y_1)$ is

$$b^{0_2}_{\Delta Y_3} = (\rho\alpha + b^0_{\Delta Y_3}) P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1) + b^0_{\Delta Y_3} P(R_3 = 1 \mid R_2 = 1, Y_2, Y_1) \tag{B.7}$$

Again the combination of parametrization (B.5) and future independence puts a constraint on $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$. The only values for which $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$ can remain unconstrained is $\alpha = 0$, otherwise (B.7) can hold only if $P(R_3 = 0 \mid R_2 = 1, Y_2, Y_1)$ is a constant function of Y_1 and Y_2. Nevertheless, we can still, use implications of (B.5) and (B.7) as an heuristical argument for redundancy of specification of $E(Y_3 \mid R_2 = 0, Y_2, Y_1) - E(Y_3 \mid R_3 = 1, R_2 = 1, Y_2, Y_1)$ under these assumptions. Then if we write out $E(Y_3)$ we have using the above

$$E(Y_3) = E\Big(E[Y_3 \mid Y_1]\Big)$$

$$= E\Big(E[Y_3 \mid R_2 = 1, Y_1]\, P(R_2 = 1 \mid Y_1) + E[Y_3 \mid R_2 = 0, Y_1]\, P(R_2 = 0 \mid Y_1)\Big)$$

$$\begin{aligned} = E\Big(&E\Big[E(Y_3 \mid R_2 = 1, Y_2, Y_1) \mid R_2 = 1, Y_1\Big]\ P(R_2 = 1 \mid Y_1) + \\ &E\Big[E(Y_3 \mid R_2 = 0, Y_2, Y_1) \mid R_2 = 0, Y_1\Big]\ P(R_2 = 0 \mid Y_1)\Big) \end{aligned}$$

$$\begin{aligned} = E\Big(&E\Big[(b^{0_2}_{\Delta Y_3} + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3} Y_1)\ \Big|\ R_2 = 1, Y_1\Big]\ P(R_2 = 1 \mid Y_1) + \\ &E\Big[(b^{0_2}_{\Delta Y_3} + (b^2_{\Delta Y_3} + 1)Y_2 + b^1_{\Delta Y_3} Y_1)\ \Big|\ R_2 = 0, Y_1\Big]\ P(R_2 = 0 \mid Y_1)\Big) \end{aligned}$$

$$\begin{aligned} = E\Big(&\Big(b^0_{\Delta Y_3} + \rho\alpha\, P(R_3 = 0 \mid R_2 = 1, Y_1) + (b^2_{\Delta Y_3} + 1)\Big(b^0_{\Delta Y_2} + (b^1_{\Delta Y_2} + 1)Y_1\Big) + b^1_{\Delta Y_3} Y_1\Big)\times \\ &P(R_2 = 1 \mid Y_1)\ +\ \Big(b^0_{\Delta Y_3} +\ \underbrace{E\Big[\rho\alpha\, P(R_3 = 0 \mid R_2 = 1, Y_1)\Big|\ R_2 = 0, Y_1\Big]}_{=0 \text{ since } R_2 = 0 \text{ AND } R_2 = 1 \text{ can't be simultaneously true}}\ + \\ &(b^2_{\Delta Y_3} + 1)\Big((\alpha + b^0_{\Delta Y_2}) + (b^1_{\Delta Y_2} + 1)Y_1\Big) + b^1_{\Delta Y_3} Y_1\Big)\ P(R_2 = 0 \mid Y_1)\ \Big) \end{aligned}$$

use (4.7)

$$\begin{aligned} = &b^0_{\Delta Y_3} P(R_2 = 1) + \rho\alpha P(R_3 = 0, R_2 = 1) + (b^2_{\Delta Y_3} + 1) b^0_{\Delta Y_2} P(R_2 = 1) + \\ &(b^2_{\Delta Y_3} + 1)(b^1_{\Delta Y_2} + 1)\ E\Big(Y_1\, P(R_2 = 1 \mid Y_1)\Big) + b^1_{\Delta Y_3} E\Big(Y_1\, P(R_2 = 1 \mid Y_1)\Big) + \\ &b^0_{\Delta Y_3} P(R_2 = 0) + (b^2_{\Delta Y_3} + 1)(\alpha + b^0_{\Delta Y_2}) P(R_2 = 0) + \\ &(b^2_{\Delta Y_3} + 1)(b^1_{\Delta Y_2} + 1)\ E\Big(Y_1\, P(R_2 = 0 \mid Y_1)\Big) + b^1_{\Delta Y_3} E\Big(Y_1\, P(R_2 = 0 \mid Y_1)\Big) \end{aligned}$$

$$\begin{aligned} = &b^0_{\Delta Y_3}\ +\ \rho\alpha P(R_3 = 0, R_2 = 1)\ +\ \Big(b^2_{\Delta Y_3} + 1\Big) b^0_{\Delta Y_2}\ +\ (b^2_{\Delta Y_3} + 1)\alpha\, P(R_2 = 0) \\ &\Big(b^2_{\Delta Y_3} + 1\Big)\Big(b^1_{\Delta Y_2} + 1\Big) E(Y_1)\ +\ b^1_{\Delta Y_3} E(Y_1) \end{aligned}$$

So we see that the marginal mean $E(Y_3)$ (within the previously mentioned heuristical confines of this argument) is not a function of a shift between $b^{0_2}_{\Delta Y_3}$ and $b^{0}_{\Delta Y_3}$ and that it can be expressed exclusively as a function of observed dropout rates and parameters identifiable from observed data and contemporaneous shifts δ_{21} and δ_{32}.

B.3. LISA bias w.r.t. DH for $T = 4$

We write out $E(Y_4)$ as a function of marginal dropout rates and DH parameters for $T = 4$.

$$
\begin{aligned}
E(Y_4) = {} & \gamma_0^{(4)} + (\gamma_0^{(1)} - \gamma_0^{(4)})P(R_2 = 0) + (\gamma_0^{(2)} - \gamma_0^{(4)})P(R_3 = 0, R_2 = 1) + \\
& (\gamma_0^{(3)} - \gamma_0^{(4)})P(R_4 = 0, R_3 = 1) + \gamma_3^{(4)}\Big(\beta_0^{(\geq 3)} + (\beta_0^{(2)} - \beta_0^{(\geq 3)})\, P(R_3 = 0, R_2 = 1) + \\
& (\beta_0^{(1)} - \beta_0^{(\geq 3)})\, P(R_2 = 0) + \beta_2^{(\geq 3)}\alpha_0^{(\geq 2)} + \beta_2^{(\geq 3)}(\alpha_0^{(1)} - \alpha_0^{(\geq 2)})\, P(R_2 = 0) + \\
& \beta_2^{(\geq 3)}\alpha_1^{(\geq 2)}E(Y_1) + \beta_1^{(\geq 3)}E(Y_1)\Big) + \\
& \gamma_3^{(4)}\Big(\alpha_0^{(\geq 2)} + (\alpha_0^{(1)} - \alpha_0^{(\geq 2)})P(R_2 = 0)\alpha_1^{(\geq 2)}E(Y_1)\Big) + \gamma_1^{(4)}E(Y_1)
\end{aligned}
$$

Notice that the term next to $\gamma_3^{(4)}$ and $\gamma_2^{(4)}$ are $E(Y_3)$ and $E(Y_2)$. Given that these 2 are estimated unbiasedly we could discuss bias of β^4_{LI} that comes from setting $\gamma_0^{(1)} - \gamma_0^{(4)} = \gamma_0^{(2)} - \gamma_0^{(4)} = \gamma_0^{(3)} - \gamma_0^{(4)}$. Nevertheless, in contrast to $T = 3$ there are two shifts for $T = 4$ that are to be examined w.r.t. bias of β^4_{LI}.

$$\varphi_{41} = \underbrace{E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)}_{\delta_{21}(\boldsymbol{\alpha};Y_1)} +$$

$$\underbrace{(\beta_0^{(1)} - \beta_0^{(\geq 3)}) + b_{\Delta Y_3}^{(2)}\Big[E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)\Big] - (\beta_0^{(2)} - \beta_0^{(\geq 3)})P(R_3=0 \mid R_2=1,Y_1)}_{\delta_{31}(\boldsymbol{\alpha};Y_1)} +$$

$$(\gamma_0^{(1)} - \gamma_0^{(4)}) +$$

$$b_{\Delta Y_4}^{(3)}\Big[(\beta_0^{(1)} - \beta_0^{(\geq 3)}) + b_{\Delta Y_3}^{(2)}\Big[E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)\Big] - (\beta_0^{(2)} - \beta_0^{(\geq 3)})P(R_3=0 \mid R_2=1,Y_1)\Big] +$$

$$(b_{\Delta Y_4}^{(3)} + \gamma_2^{(4)})\Big[E\Big(Y_2|R_2=0,Y_1\Big) - E\Big(Y_2|R_2=1,Y_1\Big)\Big] -$$

$$(\gamma_0^{(2)} - \gamma_0^{(4)})P(R_3=0 \mid R_2=1,Y_1) - (\gamma_0^{(3)} - \gamma_0^{(4)})P(R_4=0,R_3=1 \mid R_2=1,Y_1)$$

Last 5 rows above correspond to δ_{41}.

$$\varphi_{42} = \underbrace{(\gamma_0^{(2)} - \gamma_0^{(4)}) + b_{\Delta Y_4}^{(3)}(\beta_0^{(2)} - \beta_0^{(\geq 3)}) - (\gamma_0^{(3)} - \gamma_0^{(4)})P(R_4=0 \mid R_3=1,R_2=1,Y_2,Y_1)}_{\delta_{42}} +$$

$$\underbrace{(\beta_0^{(2)} - \beta_0^{(3)})}_{\delta_{32}}$$

APPENDIX C

R-CODE

C.1. Simulations from chapter 2

```
library(mvtnorm)
library(rje)
library(nlme)
library(ipw)
library(survey)
library(glmnet)
library(geepack)
library(Hmisc)
library(mclust)
library(monomvn)
w<-c(0,1,2,4,6,8)
alpha<-c(-8,-6,-6,-6, -4)
alpha_pos3<-c(-5.5,-2,-3.4, 2.5, 17)
gamma<-c(0.2,0.3, 0.3, 0.5, 0.6)
nrPatients<-c(125,250,500,1000)
trueProbMis<-NULL


fu<-function(x) {x[1]+w*x[2]}

createFullData1<-function(l, sigma_0Sq, sigma_1Sq, mu_Y, sigmaEpsilonSq){
  # gen U1, U2
  U<-rmvnorm(n=l,mean=c(0,0), sigma=matrix(c(sigma_0Sq,0,0,sigma_1Sq),2,2))
  S<-t(matrix(apply(U, 1, fu), length(w), l))
  M<-matrix(nrow=l, ncol=1)
  M<-cbind(M, cbind(U[,1],U[,2]))
  M<-M[,-1]
  mu<-matrix(nrow=l, ncol=length(w))
  for(k in seq(1, l)){
    mu[k,]<-mu_Y
  }
  #generate the U3, U4, U5, U6 so that Var(S(t))=Var(M(t))
  U_rest<-rmvnorm(n=l,mean=rep(0,length(w)-2), sigma=matrix(c((w[3]^2-1)*sigma_1Sq,0, 0, 0,
                                                              0,(w[4]^2-w[3]^2)*sigma_1Sq, 0, 0,
                                                              0,0,(w[5]^2-w[4]^2)*sigma_1Sq, 0,
                                                              0,0,0,(w[5]^2-w[4]^2)*sigma_1Sq)
                                                            ,length(w)-2,length(w)-2))
  # gen M1, M2, M3, M4, M5, M6
  M<-t(apply(cbind(M,U_rest), 1, cumsum))
  # gen epsilon errors
  epsilon<-rmvnorm(n=l, mean=rep(0,length(w)), sigma=matrix(c(sigmaEpsilonSq,0,0,0,0,0,
                                                              0,sigmaEpsilonSq,0,0,0,0,
                                                              0,0,sigmaEpsilonSq,0,0,0,
                                                              0,0,0,sigmaEpsilonSq,0,0,
                                                              0,0,0,0,sigmaEpsilonSq,0,
                                                              0,0,0,0,0,sigmaEpsilonSq),length(w),length(w)))
  Y_randInterceptAndSlope<-data.frame(mu+S+epsilon)
  names(Y_randInterceptAndSlope)<-c("Y_S0","Y_S1","Y_S2", "Y_S3", "Y_S4")
  Y_MartingaleEff<-data.frame(mu+M+epsilon)
  names(Y_MartingaleEff)<-c("Y_M0","Y_M1","Y_M2","Y_M3","Y_M4")
  return(cbind(Y_randInterceptAndSlope,Y_MartingaleEff, S, M))
```

```
}

funGetInForm<-function(x){
  ys<-unname(unlist(x[seq(1,length(w))]))
  rs<-unname(unlist(x[seq(length(w)+1,2*length(w))]))
  ms<-unname(unlist(x[seq(3*length(w)+1,4*length(w))]))
  misP<-unname(unlist(c(1,x[seq(2*length(w)+2,3*length(w))])))
  return(matrix(cbind(t(ys),t(rs),t(ms), t(misP)), nrow=length(w), ncol=4))
}

#function deletes values according to logit model from Diggle 2007 paper,
# logit is linear in random effect S(t) or M(t), R=1 means obs is missing

fun_miss1<-function(x){
  R<-NULL
  k<-1
  trueProbMis<-NULL
  while(k <length(w)){
    prob<-as.double(expit(alpha[k]+gamma[k]*x[k]))
    trueProbMis<-c(trueProbMis,prob)
    R_k<-rbinom(1, 1, prob)
    R<-c(R,R_k)
    if(R_k){
      R<-c(R,rep(1,length(w)-1-k))
      k<-length(w)
    } else {k<-k+1}
  }
  trueProbMis<-ifelse(rep(length(trueProbMis),(length(w)-1))!=(length(w)-1),
                      c(trueProbMis,rep(0.99, length(w)-1-length(trueProbMis))),trueProbMis)
  s<-ifelse(min(which(R==1))!=Inf,min(which(R==1))+1, length(w)+1)
  return(c(rep(1, length(R))-R, s, trueProbMis))

fun_miss3<-function(x){
  R<-NULL
  k<-1
  trueProbMis<-NULL
  while(k <length(w)){
    prob<-unlist(0.3*(exp(alpha_pos3[k]+gamma[k]*x[k])/(1+exp(alpha_pos3[k]+gamma[k]*x[k]))))
    trueProbMis<-c(trueProbMis,prob)
    R_k<-rbinom(1, 1, prob)
    R<-c(R,R_k)
    if(R_k){
      R<-c(R,rep(1,length(w)-1-k))
      k<-length(w)
    } else {k<-k+1}
  }
  trueProbMis<-ifelse(rep(length(trueProbMis),(length(w)-1))!=(length(w)-1),
                      c(trueProbMis,rep(0.99, length(w)-1-length(trueProbMis))),trueProbMis)
  s<-ifelse(min(which(R==1))!=Inf,min(which(R==1))+1, length(w)+1)
  return(c(rep(1, length(R))-R, s, trueProbMis))
}

calcStblWeights<-function(x){
  weight<-cumprod(x[1:length(w)-1])/x[seq(length(w),2*(length(w)-1))]
  return (weight^(-1))
}

i<-3
prev_Y_or_M<-matrix(c("M","S", "Y_M", "Y_S"), nrow=2, ncol=2, byrow=FALSE)
nrIterations<-1000
\newpage
```

```
for (b in c(1,2)){
                  for (a in c(1,3)){
                                    matrixResults<-matrix(nrow=1, ncol=12)
                                    set.seed(10000)
                                    LI_M<-NULL
                                    LIALL_S<-NULL
                                    LIALL_M<-NULL
                                    SWEEP_M<-NULL
                                    SWEEP_S<-NULL
                                    SWEEP_M_LIALL<-NULL
                                    SWEEP_M_LILast<-NULL
                                    SWEEP_S_LIALL<-NULL
                                    SWEEP_S_LILast<-NULL
                                    SWEEP_M_last<-NULL
                                    SWEEP_S_last<-NULL
                                    SWEEP_S_constant_LI_Constat<-NULL
                                    SWEEP_S_All_LI_Constat<-NULL
                                    LI_Constant_S<-NULL
                                    SWEEP_M_constant_LI_Constat<-NULL
                                    SWEEP_M_All_LI_Constat<-NULL
                                    LI_Constant_M<-NULL
                                    LI_S<-NULL
                                    LILast_M<-NULL
                                    LILast_S<-NULL
                                    Y_M_IPW<-NULL
                                    Y_M_IPWY<-NULL
                                    Y_S_IPW<-NULL
                                    Y_S_IPWY<-NULL
                                    minProb_M<-NULL
                                    minProb_S<-NULL
                                    medProb_M<-NULL
                                    medProb_S<-NULL
                                    drop_M<-NULL
                                    drop_S<-NULL
                                    system.time(
                                    for(j in seq(1,nrIterations)){
                                      # l=number of patients, sigma_0Sq=200, sigma_1Sq=15, mu_Y=c(0,0,0,0,0,0),
                                      # sigmaEpsilonSq=100
                                      Y<-createFullData1(nrPatients[i], 200, 15, rep(0, length(w)), 100)
                                      Y_M<-Y[,seq(length(w)+1,2*length(w))]
                                      Y_S<-Y[,seq(1,length(w))]
                                      # M and S are random effects (martingale, Laird-Waare resp.)
                                      M<-Y[,seq(3*length(w)+1,4*length(w))]
                                      S<-Y[,seq(2*length(w)+1,3*length(w))]
                                      # generate dropout
                                      h<-expression(paste("R_M_AndProbs<-t(apply(",
                                                          prev_Y_or_M[1,b],
                                                          "[,-length(w)],1,fun_miss",
                                                          a, "))", sep=""))
                                      eval(parse(text=eval(h)))

                                      h<-expression(paste("R_S_AndProbs<-t(apply(",
                                                           prev_Y_or_M[2,b],
                                                           "[,-length(w)],1,fun_miss",
                                                           a, "))", sep=""))
                                      eval(parse(text=eval(h)))

                                      # add column of 1's to the nrPatients[i]x5 R matrix
                                      # of missingness indicators
```

```
dat_M_temp<-cbind(Y_M, cbind(rep(1, nrPatients[i]), R_M_AndProbs), M)
dat_S_temp<-cbind(Y_S, cbind(rep(1, nrPatients[i]), R_S_AndProbs), S)
dat_S<-NULL
dat_M<-NULL
# transform data into a form suitable for glm function
for(h in seq(1,nrPatients[i])){
  tem<-funGetInForm(dat_M_temp[h,])
  dat_M<-rbind(dat_M,
               cbind(rep(h, length(w)),
                     seq(1,length(w)),
                     tem,
                     c(NA, tem[,1][c(-length(w))]),
                     c(NA, tem[,3][c(-length(w))]),
                     rep(dat_M_temp[h,2*length(w)+1]))
  )
}

for(h in seq(1,nrPatients[i])){
  tem<-funGetInForm(dat_S_temp[h,])
  dat_S<-rbind(dat_S,
               cbind(rep(h, length(w)),
                     seq(1,length(w)),
                     tem,
                     c(NA, tem[,1][c(-length(w))]),
                     c(NA, tem[,3][c(-length(w))]),
                     rep(dat_S_temp[h,2*length(w)+1]))
  )
}

dat_M<-data.frame(dat_M)
names(dat_M)<-c("pid", "obs", "y", "R","M","probMis","prevY","prevM","dropOut")
dat_M$mis<-1-dat_M$R
dat_M$obs_startZero<-dat_M$obs-1
dat_M$deltaY<-dat_M$y
dat_M$deltaY[dat_M$obs!=1]<-dat_M$deltaY[dat_M$obs!=1]-dat_M$prevY[dat_M$obs!=1]

tm<-dat_M[,c(1,2,3)]
dat_M$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dat_M$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dat_M$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dat_M$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dat_M$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))


dat_S<-data.frame(dat_S)
names(dat_S)<-c("pid", "obs", "y", "R","S","probMis","prevY","prevS","dropOut")
dat_S$mis<-1-dat_S$R
dat_S$obs_startZero<-dat_S$obs-1
dat_S$deltaY<-dat_S$y
dat_S$deltaY[dat_S$obs!=1]<-dat_S$deltaY[dat_S$obs!=1]-dat_S$prevY[dat_S$obs!=1]

tm<-dat_S[,c(1,2,3)]
dat_S$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dat_S$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dat_S$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dat_S$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dat_S$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))

fitMDat<-dat_M[dat_M$mis==0, c(1,2,12)]
fit_M<-lmList(deltaY˜1|obs, data=fitMDat, pool=F)

LI_M<-rbind(LI_M, cumsum(coef(fit_M)[[1]]))
```

```
    predictedLI_1<-NULL
    predictedLI_2<-NULL
    namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
    for(t in seq(1,6)){
      temp<-dat_M[dat_M$mis==0 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]
      if(t!=1){
        predictors<-"1"
        out<-names(temp)[3]

        h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
        eval(parse(text=eval(h)))
        gh<-matrix(nrow=nrPatients[i], ncol=1)
        gh[,1]<-seq(1,nrPatients[i])
        gh<-data.frame(gh)
        names(gh)[1]<-"pid"
        temp1<-dat_M[dat_M$dropOut>t-1 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]

        gh<-merge(gh, temp1[, c(1,seq(4, 4+t-2))], all.x=TRUE)
        if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1],seq(2, t)]<-predictedLI_1[-temp1[,1],]}
        d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
        names(d)<-namPred[seq(1,t)]
        predTemp<-rep(0, nrPatients[i])
        predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
        predTemp[-temp[,which(names(temp)=="pid")]]<-
                                              unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])

        predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
        predictedLI_1<-t(apply(predictedLI_2, 1, rev))
      } else {
        mod1<-lm(deltaY˜1,data=temp)
        predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
        predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
      }
    }
LI_Constant_M<-rbind(LI_Constant_M, apply(predictedLI_2,2,mean))

    dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
          nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
    names(dTemp)<-c("pid","obs","y","mis",
                    "Y_tminus1", "Y_tminus2","Y_tminus3",
                    "Y_tminus4","Y_tminus5")
    dTemp$pid<-dat_M$pid
    dTemp$obs<-dat_M$obs

    dTemp$y<-rep(0,nrPatients[i]*6)
    dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
    dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
    dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
    dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
    dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
    dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

    dTemp$mis<-rep(0, nrPatients[i]*6)
    tm<-dTemp[,c(1,2,3)]
    dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
    dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
    dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
    dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
    dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))
```

```
  SWM<-diag(length(w)+1)
  for(t in seq(6,1)){
    temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
    if(t!=1){
      vtemp<-names(temp)[seq(4, 4+t-2)]
      predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
      out<-names(temp)[3]
      h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
      eval(parse(text=eval(h)))
      h<-expression(paste("mod",t+1,"<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
      eval(parse(text=eval(h)))
      SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
    } else {
      mod1<-lm(y˜1,data=temp)
      mod2<-lm(y˜1,data=temp)
      SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
    }
  }
  SWEEP_M_All_LI_Constat<-rbind(SWEEP_M_All_LI_Constat, SWM[-1])


SWM<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
  if(t!=1){
    y<-temp[,3]
    x<-temp[,4]
    mod1<-glm(y˜offset(I(1*x)))
    #predict(mod1)
    SWM<-SWM%*%t(cbind(diag(t), c(unname(mod1$coeff), 1, rep(0,t-2))))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
    SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  }
}
  SWEEP_M_constant_LI_Constat<-rbind(SWEEP_M_constant_LI_Constat, SWM[-1])


  predictedLI_1<-NULL
  predictedLI_2<-NULL
  namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
  for(t in seq(1,6)){
    temp<-dat_M[dat_M$mis==0 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]
    if(t!=1){
      vtemp<-names(temp)[seq(4, 4+t-2)]
      predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
      out<-names(temp)[3]

      h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
      eval(parse(text=eval(h)))
      gh<-matrix(nrow=nrPatients[i], ncol=1)
      gh[,1]<-seq(1,nrPatients[i])
      gh<-data.frame(gh)
      names(gh)[1]<-"pid"
      temp1<-dat_M[dat_M$dropOut>t-1 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]

      gh<-merge(gh, temp1[, c(1,seq(4, 4+t-2))], all.x=TRUE)
     if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1],seq(2, t)]<-predictedLI_1[-temp1[,1],]}
      d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
      names(d)<-namPred[seq(1,t)]
      predTemp<-rep(0, nrPatients[i])
    predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
    predTemp[-temp[,which(names(temp)=="pid")]]<-
                                         unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])

    predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
    predictedLI_1<-t(apply(predictedLI_2, 1, rev))
  } else {
    mod1<-lm(deltaY˜1,data=temp)
    predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
    predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
  }
}
LIALL_M<-rbind(LIALL_M, apply(predictedLI_2,2,mean))
dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
                          nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
names(dTemp)<-c("pid","obs","y","mis",
                 "Y_tminus1", "Y_tminus2","Y_tminus3",
                 "Y_tminus4","Y_tminus5")
dTemp$pid<-dat_M$pid
dTemp$obs<-dat_M$obs

dTemp$y<-rep(0,nrPatients[i]*6)
dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

dTemp$mis<-rep(0, nrPatients[i]*6)
tm<-dTemp[,c(1,2,3)]
dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))

SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
  if(t!=1){
    vtemp<-names(temp)[seq(4, 4+t-2)]
    predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    h<-expression(paste("mod",t+1,"<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
    SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  }
}
SWEEP_M_LIALL<-rbind(SWEEP_M_LIALL, SWS[-1])

predictedLI_1<-NULL
predictedLI_2<-NULL
namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
for(t in seq(1,6)){
  temp<-dat_M[dat_M$mis==0 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]
  if(t!=1){
    vtemp<-names(temp)[4]
    predictors<-"Y_tminus1"
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"~",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    gh<-matrix(nrow=nrPatients[i], ncol=1)
    gh[,1]<-seq(1,nrPatients[i])
    gh<-data.frame(gh)
    names(gh)[1]<-"pid"
    temp1<-dat_M[dat_M$dropOut>t-1 & dat_M$obs==t, c(1,2,12,13,14,15,16,17)]
    gh<-merge(gh, temp1[,c(1,4)], all.x=TRUE)
   if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1],2]<-predictedLI_1[-temp1[,1],1]}
    d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
    names(d)<-namPred[c(1,2)]
    predTemp<-rep(0,nrPatients[i])
    predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
    predTemp[-temp[,which(names(temp)=="pid")]]<-
                                      unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])

    predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
    predictedLI_1<-t(apply(predictedLI_2, 1, rev))
  } else {
    mod1<-lm(deltaY~1,data=temp)
    predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
    predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
  }
}
LILast_M<-rbind(LILast_M, apply(predictedLI_2,2,mean))



dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
                         nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
names(dTemp)<-c("pid","obs","y","mis",
                "Y_tminus1", "Y_tminus2","Y_tminus3",
                "Y_tminus4","Y_tminus5")
dTemp$pid<-dat_M$pid
dTemp$obs<-dat_M$obs

dTemp$y<-rep(0,nrPatients[i]*6)
dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

dTemp$mis<-rep(0, nrPatients[i]*6)
tm<-dTemp[,c(1,2,3)]
dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))

SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
  if(t!=1){
    vtemp<-names(temp)[4]
    predictors<-"Y_tminus1"
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    SWS<-SWS%*%t(cbind(diag(t),c(unname(mod1$coeff), rep(0,t-2))))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
    SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  }
}
SWEEP_M_LILast<-rbind(SWEEP_M_LILast, SWS[-1])

fitSDat<-dat_S[dat_S$mis==0, c(1,2,12)]
fit_S<-lmList(deltaY˜1|obs, data=fitSDat, pool=F)
LI_S<-rbind(LI_S, cumsum(coef(fit_S)[[1]]))

predictedLI_1<-NULL
predictedLI_2<-NULL
namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
for(t in seq(1,6)){
  temp<-dat_S[dat_S$mis==0 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]
  if(t!=1){
    predictors<-"1"
    out<-names(temp)[3]

    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    gh<-matrix(nrow=nrPatients[i], ncol=1)
    gh[,1]<-seq(1,nrPatients[i])
    gh<-data.frame(gh)
    names(gh)[1]<-"pid"
    temp1<-dat_S[dat_S$dropOut>t-1 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]

    gh<-merge(gh, temp1[, c(1,seq(4, 4+t-2))], all.x=TRUE)
   if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1],seq(2, t)]<-predictedLI_1[-temp1[,1],]}
    d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
    names(d)<-namPred[seq(1,t)]
    predTemp<-rep(0, nrPatients[i])
    predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
    predTemp[-temp[,which(names(temp)=="pid")]]<-
                                    unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])

    predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
    predictedLI_1<-t(apply(predictedLI_2, 1, rev))
  } else {
    mod1<-lm(deltaY˜1,data=temp)
    predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
    predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
  }
}
LI_Constant_S<-rbind(LI_Constant_S,apply(predictedLI_2,2,mean))

dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
                         nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
names(dTemp)<-c("pid","obs","y","mis",
                "Y_tminus1", "Y_tminus2","Y_tminus3",
                "Y_tminus4","Y_tminus5")
dTemp$pid<-dat_M$pid
dTemp$obs<-dat_M$obs

dTemp$y<-rep(0,nrPatients[i]*6)
dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
```

```
   dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
   dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
   dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
   dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
   dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

   dTemp$mis<-rep(0, nrPatients[i]*6)
   tm<-dTemp[,c(1,2,3)]
   dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
   dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
   dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
   dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
   dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))


   SWS<-diag(length(w)+1)
   for(t in seq(6,1)){
     temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
     if(t!=1){
       vtemp<-names(temp)[seq(4, 4+t-2)]
       predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
       out<-names(temp)[3]
       h<-expression(paste("mod1<-lm(",out,"~",predictors, ", data=temp)", sep=""))
       eval(parse(text=eval(h)))
       h<-expression(paste("mod",t+1,"<-lm(",out,"~",predictors, ", data=temp)", sep=""))
       eval(parse(text=eval(h)))
       SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
     } else {
       mod1<-lm(y~1,data=temp)
       mod2<-lm(y~1,data=temp)
       SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
     }
   }
   SWEEP_S_All_LI_Constat<-rbind(SWEEP_S_All_LI_Constat, SWS[-1])


   SWS<-diag(length(w)+1)
   for(t in seq(6,1)){
     temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
     if(t!=1){
       y<-temp[,3]
       x<-temp[,4]
       mod1<-glm(y~offset(I(1*x)))
       #predict(mod1)
       SWS<-SWS%*%t(cbind(diag(t), c(unname(mod1$coeff), 1, rep(0,t-2))))
     } else {
       mod1<-lm(y~1,data=temp)
       mod2<-lm(y~1,data=temp)
       SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
     }
   }
   SWEEP_S_constant_LI_Constat<-rbind(SWEEP_S_constant_LI_Constat, SWS[-1])


predictedLI_1<-NULL
predictedLI_2<-NULL
namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
for(t in seq(1,6)){
   temp<-dat_S[dat_S$mis==0 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]
   if(t!=1){
            vtemp<-names(temp)[seq(4, 4+t-2)]
            predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
       out<-names(temp)[3]
       h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
       eval(parse(text=eval(h)))
       gh<-matrix(nrow=nrPatients[i], ncol=1)
       gh[,1]<-seq(1,nrPatients[i])
       gh<-data.frame(gh)
       names(gh)[1]<-"pid"
       temp1<-dat_S[dat_S$dropOut>t-1 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]

       gh<-merge(gh, temp1[, c(1,seq(4, 4+t-2))], all.x=TRUE)
       if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1],seq(2, t)]<-predictedLI_1[-temp1[,1],]}
       d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
       names(d)<-namPred[seq(1,t)]
       predTemp<-rep(0,nrPatients[i])
       predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
       predTemp[-temp[,which(names(temp)=="pid")]]<-
                                      unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])
       predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
       predictedLI_1<-t(apply(predictedLI_2, 1, rev))
       } else {
       mod1<-lm(deltaY˜1,data=temp)
       predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
       predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
      }
}
LIALL_S<-rbind(LIALL_S,apply(predictedLI_2,2,mean))

dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
                         nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
names(dTemp)<-c("pid","obs","y","mis",
                "Y_tminus1", "Y_tminus2","Y_tminus3",
                "Y_tminus4","Y_tminus5")
dTemp$pid<-dat_M$pid
dTemp$obs<-dat_M$obs

dTemp$y<-rep(0,nrPatients[i]*6)
dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

dTemp$mis<-rep(0, nrPatients[i]*6)
tm<-dTemp[,c(1,2,3)]
dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))

SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
  if(t!=1){
    vtemp<-names(temp)[seq(4, 4+t-2)]
    predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    h<-expression(paste("mod",t+1,"<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
```

```
      SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
    } else {
      mod1<-lm(y~1,data=temp)
      mod2<-lm(y~1,data=temp)
      SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
    }
  }
  SWEEP_S_LIALL<-rbind(SWEEP_S_LIALL, SWS[-1])


predictedLI_1<-NULL
predictedLI_2<-NULL
namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
for(t in seq(1,6)){
  temp<-dat_S[dat_S$mis==0 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]
  if(t!=1){
    vtemp<-names(temp)[4]
    predictors<-"Y_tminus1"
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"~",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    gh<-matrix(nrow=nrPatients[i], ncol=1)
    gh[,1]<-seq(1,nrPatients[i])
    gh<-data.frame(gh)
    names(gh)[1]<-"pid"
    temp1<-dat_S[dat_S$dropOut>t-1 & dat_S$obs==t, c(1,2,12,13,14,15,16,17)]
    gh<-merge(gh, temp1[,c(1,4)], all.x=TRUE)
   if(length(temp1[,1])!=nrPatients[i]){gh[-temp1[,1], 2]<-predictedLI_1[-temp1[,1],1]}
    d<-data.frame(cbind(rep(1, nrPatients[i]), gh[,-c(1)]))
    names(d)<-namPred[c(1,2)]
    predTemp<-rep(0,nrPatients[i])
    predTemp[temp[,which(names(temp)=="pid")]]<-temp[,which(names(temp)=="deltaY")]
    predTemp[-temp[,which(names(temp)=="pid")]]<-
                                   unname(predict(mod1, d)[-temp[, which(names(temp)=="pid")]])

    predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+predTemp)
    predictedLI_1<-t(apply(predictedLI_2, 1, rev))
  } else {
    mod1<-lm(deltaY~1,data=temp)
    predictedLI_1<-cbind(predictedLI_1, temp$deltaY)
    predictedLI_2<-cbind(predictedLI_2, temp$deltaY)
  }
}
LILast_S<-rbind(LILast_S,apply(predictedLI_2,2,mean))

dTemp<-data.frame(matrix(rep(0,nrPatients[i]*6*9),
                         nrow=nrPatients[i]*6,ncol=9, byrow=TRUE))
names(dTemp)<-c("pid","obs","y","mis",
                "Y_tminus1", "Y_tminus2","Y_tminus3",
                "Y_tminus4","Y_tminus5")
dTemp$pid<-dat_M$pid
dTemp$obs<-dat_M$obs

dTemp$y<-rep(0,nrPatients[i]*6)
dTemp$y[seq(1,nrPatients[i]*6,6)]<-predictedLI_2[,1]
dTemp$y[seq(2,nrPatients[i]*6,6)]<-predictedLI_2[,2]
dTemp$y[seq(3,nrPatients[i]*6,6)]<-predictedLI_2[,3]
dTemp$y[seq(4,nrPatients[i]*6,6)]<-predictedLI_2[,4]
dTemp$y[seq(5,nrPatients[i]*6,6)]<-predictedLI_2[,5]
dTemp$y[seq(6,nrPatients[i]*6,6)]<-predictedLI_2[,6]

dTemp$mis<-rep(0, nrPatients[i]*6)
```

```
tm<-dTemp[,c(1,2,3)]
dTemp$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 1))))
dTemp$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 2))))
dTemp$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 3))))
dTemp$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 4))))
dTemp$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$pid),Lag, 5))))



SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dTemp[dTemp$mis==0 & dTemp$obs==t, c(1,2,3,5,6,7,8,9)]
  if(t!=1){
    vtemp<-names(temp)[4]
    predictors<-"Y_tminus1"
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    SWS<-SWS%∗%t(cbind(diag(t),c(unname(mod1$coeff), rep(0,t-2))))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
    SWS<-SWS%∗%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  }
}
SWEEP_S_LILast<-rbind(SWEEP_S_LILast, SWS[-1])

SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dat_S[dat_S$mis==0 & dat_S$obs==t, c(1,2,3,13,14,15,16,17)]
  if(t!=1){
    vtemp<-names(temp)[seq(4, 4+t-2)]
    predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2∗(t-1)]))
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    h<-expression(paste("mod",t+1,"<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    SWS<-SWS%∗%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
    SWS<-SWS%∗%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
  }
}
SWEEP_S<-rbind(SWEEP_S, SWS[-1])



SWS<-diag(length(w)+1)
for(t in seq(6,1)){
  temp<-dat_S[dat_S$mis==0 & dat_S$obs==t, c(1,2,3,13,14,15,16,17)]
  if(t!=1){
    vtemp<-names(temp)[4]
    predictors<-"Y_tminus1"
    out<-names(temp)[3]
    h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
    eval(parse(text=eval(h)))
    SWS<-SWS%∗%t(cbind(diag(t),c(unname(mod1$coeff), rep(0,t-2))))
  } else {
    mod1<-lm(y˜1,data=temp)
    mod2<-lm(y˜1,data=temp)
```

```
                                          SWS<-SWS%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
                                        }
                                      }
                                      SWEEP_S_last<-rbind(SWEEP_S_last, SWS[-1])



                                      SWM<-diag(length(w)+1)
                                      for(t in seq(6,1)){
                                        temp<-dat_M[dat_M$mis==0 & dat_M$obs==t, c(1,2,3,13,14,15,16,17)]
                                        if(t!=1){
                                          vtemp<-names(temp)[seq(4, 4+t-2)]
                                          predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
                                          out<-names(temp)[3]
                                          h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
                                          eval(parse(text=eval(h)))
                                          h<-expression(paste("mod",t+1,"<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
                                          eval(parse(text=eval(h)))
                                          SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
                                        } else {
                                          mod1<-lm(y˜1,data=temp)
                                          mod2<-lm(y˜1,data=temp)
                                          SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
                                        }
                                      }
                                      SWEEP_M<-rbind(SWEEP_M, SWM[-1])

                                      SWM<-diag(length(w)+1)
                                      for(t in seq(6,1)){
                                        temp<-dat_M[dat_M$mis==0 & dat_M$obs==t, c(1,2,3,13,14,15,16,17)]
                                        if(t!=1){
                                          vtemp<-names(temp)[4]
                                          predictors<-"Y_tminus1"
                                          out<-names(temp)[3]
                                          h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
                                          eval(parse(text=eval(h)))
                                          SWM<-SWM%*%t(cbind(diag(t),c(unname(mod1$coeff), rep(0,t-2))))
                                        } else {
                                          mod1<-lm(y˜1,data=temp)
                                          mod2<-lm(y˜1,data=temp)
                                          SWM<-SWM%*%t(cbind(diag(length(unname(mod1$coeff))),unname(mod1$coeff)))
                                        }
                                      }
                                      SWEEP_M_last<-rbind(SWEEP_M_last, SWM[-1])


  #IPW approach
### get required data
dat_SNo1<-dat_S[dat_S$obs!=1,]
dat_SNo1$obs_startZero<-dat_SNo1$obs_startZero-1
for (h in seq(1,nrPatients[i])){
temp<-c(0, Lag(dat_SNo1$mis[dat_SNo1$pid==h],1)[c(-1)])
dat_SNo1$prevMis[dat_SNo1$pid==h]<-temp
}
dat_MNo1<-dat_M[dat_M$obs!=1,]
dat_MNo1$obs_startZero<-dat_MNo1$obs_startZero-1

for (h in seq(1,nrPatients[i])){
temp<-c(0, Lag(dat_MNo1$mis[dat_MNo1$pid==h],1)[c(-1)])
dat_MNo1$prevMis[dat_MNo1$pid==h]<-temp
}
```

```
lambdas_M<-NULL
lambdasY_M<-NULL
lambdas_S<-NULL
lambdasY_S<-NULL
stable<-NULL

 for (h in seq(2,length(w))){
dM<-dat_MNo1[dat_MNo1$obs==h & dat_MNo1$prevMis==0,]
dS<-dat_SNo1[dat_SNo1$obs==h & dat_SNo1$prevMis==0,]

tagMSM<-0
tryCatch( ms<-glm(R˜prevY, family=binomial, data=dM),
error=function(err) tagMSM<<-1)
# weights calculated from a logit that has only the last observations as predictor
predY<-expit(cbind(rep(1,nrPatients[i]), dat_MNo1$prevY[dat_MNo1$obs==h])%*%ms$coefficients)
# true weights
pred<-1-dat_MNo1[dat_MNo1$obs==h, which(names(dat_MNo1)=="probMis")]

#save time/individualspecific weights for Martingale rand effect

lambdas_M<-cbind(lambdas_M,pred)
lambdasY_M<-cbind(lambdasY_M,predY)


tagMSM<-0
                                            tryCatch( ms<-glm(R˜prevY, family=binomial, data=dS),
                                                     error=function(err) tagMSM<<-1)
# weights calculated from a logit that has only the last observation as predictor
# this time for Laird-Waare data
predY<-expit(cbind(rep(1,nrPatients[i]), dat_SNo1$prevY[dat_SNo1$obs==h])%*%ms$coefficients)
# true weights
pred<-1-dat_SNo1[dat_SNo1$obs==h, which(names(dat_SNo1)=="probMis")]

lambdas_S<-cbind(lambdas_S,pred)
lambdasY_S<-cbind(lambdasY_S,predY)

#stabilized weights
#stable<-c(stable, mean(d$R))
#no stabilization
stable<-c(stable,1)
}
stable<-matrix(stable, nrow=1, ncol=length(w)-1)
#stable<-matrix(missingM[j, 2:length(w)], nrow=1, ncol=5)
#stable<-matrix(rep(1,5), nrow=1, ncol=5)
lambdas_M<-cbind(lambdas_M,stable[rep(1:1, times = nrPatients[i]), ])
lambdas_S<-cbind(lambdas_S,stable[rep(1:1, times = nrPatients[i]), ])

dat_M$ipw.weights<-c(t(cbind(rep(1,nrPatients[i]),t(apply(lambdas_M, 1, calcStblWeights)))))
                                          dat_M$ipw.weightsY<-c(t(cbind(rep(1,nrPatients[i]),t(apply(lambdasY_M, 1, calcStblWeights)))))
dat_S$ipw.weights<-c(t(cbind(rep(1,nrPatients[i]),t(apply(lambdas_S, 1, calcStblWeights)))))
dat_S$ipw.weightsY<-c(t(cbind(rep(1,nrPatients[i]),t(apply(lambdasY_S, 1, calcStblWeights)))))

YM_IPW<-NULL
YM_IPW_Y<-NULL
YS_IPW<-NULL
YS_IPW_Y<-NULL
  #fit 1 ordinary glm for  baseline outcome and 5 weighted glm's using weights calculated
#above k<-6
for (k in seq(1, length(w))){
dy2<-dat_M[dat_M$obs==k & dat_M$mis==0,]
#     dy2<-dat_M[dat_M$obs==k & dat_M$mis==0 & dat_M$overlap==1,]
#                  if (k!=1){
#                       dy2$w<-tempM$ipw.weights[which(dat_MNo1$obs==k & dat_MNo1$mis==0)]
#                  }
if (k==5){
minProb_M<-c(minProb_M, min(1/dy2$ipw.weights))
medProb_M<-c(medProb_M, median(1/dy2$ipw.weights))
}
if (k==1){
tagMSM<-0
tryCatch(
msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 data = dy2))),
error=function(err) tagMSM<<-1)
tagMSMY<-0
tryCatch(
msmY <- (svyglm(y ~ 1, design = svydesign(id=~pid,
data = dy2))),
error=function(err) tagMSMY<<-1)
tagMSMM<-0
tryCatch(
msmM <- (svyglm(y ~ 1, design = svydesign(id=~pid,
data = dy2))),
error=function(err) tagMSMM<<-1)

} else {
tagMSM<-0
tryCatch( msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 weights = ~ ipw.weights,
 data = dy2))),
error=function(err) tagMSM<<-1)

tagMSMY<-0
tryCatch( msmY <- (svyglm(y ~ 1, design = svydesign(id=~pid,
weights = ~ ipw.weightsY,
data = dy2))),
error=function(err) tagMSMY<<-1)
tagMSMM<-0
tryCatch( msmM <- (svyglm(y ~ 1, design = svydesign(id=~pid,
weights = ~ ipw.weightsM,
data = dy2))),
error=function(err) tagMSMM<<-1)
}
# save coefficents for jth iteration for YM_IPW
YM_IPW<-c(YM_IPW, ifelse(!tagMSM, unname(msm$coefficients), NA))
YM_IPW_Y<-c(YM_IPW_Y, ifelse(!tagMSMY, unname(msmY$coefficients), NA))
YM_IPW_M<-c(YM_IPW_M, ifelse(!tagMSMM, unname(msmM$coefficients), NA))

dy2<-dat_S[dat_S$obs==k & dat_S$mis==0,]
#     dy2<-dat_M[dat_M$obs==k & dat_M$mis==0 & dat_M$overlap==1,]
#                  if (k!=1){
#                       dy2$w<-tempM$ipw.weights[which(dat_MNo1$obs==k & dat_MNo1$mis==0)]
#                  }
if (k==5){
minProb_S<-c(minProb_S, min(1/dy2$ipw.weights))
medProb_S<-c(medProb_S, median(1/dy2$ipw.weights))
}
if (k==1){
tagMSM<-0
tryCatch(
msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 data = dy2))),
error=function(err) tagMSM<<-1)
```

```
tagMSMY<-0
tryCatch(
msmY <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
data = dy2))),
error=function(err) tagMSMY<<-1)
tagMSMM<-0
tryCatch(
msmM <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
data = dy2))),
error=function(err) tagMSMM<<-1)

} else {
tagMSM<-0
tryCatch( msm <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
 weights = ˜ ipw.weights,
 data = dy2))),
error=function(err) tagMSM<<-1)

tagMSMY<-0
tryCatch( msmY <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
weights = ˜ ipw.weightsY,
data = dy2))),
error=function(err) tagMSMY<<-1)
tagMSMM<-0
tryCatch( msmM <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
weights = ˜ ipw.weightsM,
data = dy2))),
error=function(err) tagMSMM<<-1)
}
# save coefficents for jth iteration for YM_IPW
YS_IPW<-c(YS_IPW, ifelse(!tagMSM, unname(msm$coefficients), NA))
YS_IPW_Y<-c(YS_IPW_Y, ifelse(!tagMSMY, unname(msmY$coefficients), NA))
YS_IPW_M<-c(YS_IPW_M, ifelse(!tagMSMM, unname(msmM$coefficients), NA))
}
Y_M_IPW<-rbind(Y_M_IPW,YM_IPW)
Y_M_IPWY<-rbind(Y_M_IPWY,YM_IPW_Y)
Y_M_IPWM<-rbind(Y_M_IPWM,YM_IPW_M)

Y_S_IPW<-rbind(Y_S_IPW,YS_IPW)
Y_S_IPWY<-rbind(Y_S_IPWY,YS_IPW_Y)
Y_S_IPWM<-rbind(Y_S_IPWM,YS_IPW_M)

#fit 1 ordinary glm for  baseline outcome and 5 weighted glm’s using weights calculated
#above
for (k in seq(1, length(w))){
dy2<-dat_M[dat_M$obs==k & dat_M$mis==0,]
if (k==5){
minProb_M<-c(minProb_M, min(1/dy2$ipw.weights))
medProb_M<-c(medProb_M, median(1/dy2$ipw.weights))
}
if (k==1){
tagMSM<-0
tryCatch(
msm <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
 data = dy2))),
error=function(err) tagMSM<<-1)
tagMSMY<-0
tryCatch(
msmY <- (svyglm(y ˜ 1, design = svydesign(id=˜pid,
data = dy2))),
error=function(err) tagMSMY<<-1)
```

```
} else {
tagMSM<-0
tryCatch( msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 weights = ~ ipw.weights,
 data = dy2))),
error=function(err) tagMSM<<-1)

tagMSMY<-0
tryCatch( msmY <- (svyglm(y ~ 1, design = svydesign(id=~pid,
weights = ~ ipw.weightsY,
data = dy2))),
error=function(err) tagMSMY<<-1)
}
# save coefficents for jth iteration for YM_IPW
YM_IPW<-c(YM_IPW, ifelse(!tagMSM, unname(msm$coefficients), NA))
YM_IPW_Y<-c(YM_IPW_Y, ifelse(!tagMSMY, unname(msmY$coefficients), NA))

## do all the same for Laird-Waare data
dy2<-dat_S[dat_S$obs==k & dat_S$mis==0,]
                }
if (k==5){
minProb_S<-c(minProb_S, min(1/dy2$ipw.weights))
medProb_S<-c(medProb_S, median(1/dy2$ipw.weights))
}
if (k==1){
tagMSM<-0
tryCatch(
msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 data = dy2))),
error=function(err) tagMSM<<-1)
tagMSMY<-0
tryCatch(
msmY <- (svyglm(y ~ 1, design = svydesign(id=~pid,
data = dy2))),
error=function(err) tagMSMY<<-1)

} else {
tagMSM<-0
tryCatch( msm <- (svyglm(y ~ 1, design = svydesign(id=~pid,
 weights = ~ ipw.weights,
 data = dy2))),
error=function(err) tagMSM<<-1)

tagMSMY<-0
tryCatch( msmY <- (svyglm(y ~ 1, design = svydesign(id=~pid,
weights = ~ ipw.weightsY,
data = dy2))),
error=function(err) tagMSMY<<-1)
}
# save coefficents for jth iteration for YM_IPW
YS_IPW<-c(YS_IPW, ifelse(!tagMSM, unname(msm$coefficients), NA))
YS_IPW_Y<-c(YS_IPW_Y, ifelse(!tagMSMY, unname(msmY$coefficients), NA))

}
#save estimates for jth simualted sample
Y_M_IPW<-rbind(Y_M_IPW,YM_IPW)
Y_M_IPWY<-rbind(Y_M_IPWY,YM_IPW_Y)

Y_S_IPW<-rbind(Y_S_IPW,YS_IPW)
Y_S_IPWY<-rbind(Y_S_IPWY,YS_IPW_Y)
```

```
                                    })



matrixResults<-rbind(apply(SWEEP_M, 2, mean), apply(SWEEP_M, 2, median), apply(SWEEP_M, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_M_LIALL, 2, mean), apply(SWEEP_M_LIALL, 2, median), apply(SWEEP_M_LIALL, 2, sd),
                     rep(" ",length(w)),
                     apply(LIALL_M, 2, mean), apply(LIALL_M, 2, median), apply(LIALL_M, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_M_last, 2, mean), apply(SWEEP_M_last, 2, median), apply(SWEEP_M_last, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_M_LILast, 2, mean), apply(SWEEP_M_LILast, 2, median), apply(SWEEP_M_LILast, 2, sd),
                     rep(" ",length(w)),
                     apply(LILast_M, 2, mean), apply(LILast_M, 2, median), apply(LILast_M, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_M_constant_LI_Constat, 2, mean), apply(SWEEP_M_constant_LI_Constat, 2, median),
                                                                                 apply(SWEEP_M_constant_LI_Constat, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_M_All_LI_Constat, 2, mean), apply(SWEEP_M_All_LI_Constat, 2, median),
                                                                                      apply(SWEEP_M_All_LI_Constat, 2, sd),
                     rep(" ",length(w)),
                     apply(LI_Constant_M, 2, mean), apply(LI_Constant_M, 2, median), apply(LI_Constant_M, 2, sd),
                     rep(" ",length(w)),
                     apply(LI_M, 2, mean), apply(LI_M, 2, median), apply(LI_M, 2, sd),
                     rep(" ",length(w)),
                     apply(Y_M_IPW, 2, mean), apply(Y_M_IPW, 2, median), apply(Y_M_IPW, 2, sd),
                     rep(" ",length(w)),
                     apply(Y_M_IPWY, 2, mean), apply(Y_M_IPWY, 2, median), apply(Y_M_IPWY, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S, 2, mean), apply(SWEEP_S, 2, median), apply(SWEEP_S, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S_LIALL, 2, mean), apply(SWEEP_S_LIALL, 2, median), apply(SWEEP_S_LIALL, 2, sd),
                     rep(" ",length(w)),
                     apply(LIALL_S, 2, mean), apply(LIALL_S, 2, median), apply(LIALL_S, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S_last, 2, mean), apply(SWEEP_S_last, 2, median), apply(SWEEP_S_last, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S_LILast, 2, mean), apply(SWEEP_S_LILast, 2, median), apply(SWEEP_S_LILast, 2, sd),
                     rep(" ",length(w)),
                     apply(LILast_S, 2, mean), apply(LILast_S, 2, median), apply(LILast_S, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S_constant_LI_Constat, 2, mean), apply(SWEEP_S_constant_LI_Constat, 2, median),
                                                                                            apply(SWEEP_S_constant_LI_Constat, 2, sd),
                     rep(" ",length(w)),
                     apply(SWEEP_S_All_LI_Constat, 2, mean), apply(SWEEP_S_All_LI_Constat, 2, median),
                                                                                            apply(SWEEP_S_All_LI_Constat, 2, sd),
                     rep(" ",length(w)),
                     apply(LI_Constant_S, 2, mean), apply(LI_Constant_S, 2, median), apply(LI_Constant_S, 2, sd),
                     rep(" ",length(w)),
                     apply(LI_S, 2, mean), apply(LI_S, 2, median), apply(LI_S, 2, sd),
                     rep(" ",length(w)),
                     apply(Y_S_IPW, 2, mean), apply(Y_S_IPW, 2, median), apply(Y_S_IPW, 2, sd),
                     rep(" ",length(w)),
                     apply(Y_S_IPWY, 2, mean), apply(Y_S_IPWY, 2, median), apply(Y_S_IPWY, 2, sd),
                     rep(" ",length(w)))

matrixResults<-cbind(rbind(cbind(c("Y_M",
                                 c("","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
 "","","","",
 "","","","",
                                   "","", "")),
                                 c("SWEEP", "","","",
                                   "SWEEP_LIALL", "","","",
                                   "LI_Pred_ALL","","","",
                                   "SWEEP_Last", "","","",
                                   "SWEEP_LILast", "","","",
                                   "LI_Pred_last","","","",
                                   "SWEEP_Const_LI_Const","","","",
                                   "SWEEP_All_LI_Const","","","",
                                   "LI_Pred_Const","","","",
                                   "LI","","","",
"IPW_TrueWeights","","","",
"IPW_lastObsLogit","","","",""),
                                c("Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
"Mean","Median","SD","",
"Mean","Median","SD","",
                                  "Mean","Median","SD","")),
                           cbind(c("Y_S",
                                 c("","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
                                   "","","","",
 "","","","",
 "","","","",
                                   "","","","",
                                    "","", "")),
                                 c("SWEEP", "","","",
                                   "SWEEP_LIALL", "","","",
                                   "LI_Pred_ALL","","","",
                                   "SWEEP_Last", "","","",
                                   "SWEEP_LILast", "","","",
                                   "LI_Pred_last","","","",
                                   "SWEEP_Const_LI_Const","","","",
                                   "SWEEP_All_LI_Const","","","",
                                   "LI_Pred_Const","","","",
                                   "LI","","","",
"IPW_TrueWeights","","","",
"IPW_lastObsLogit","","","",""),
                                c("Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","",
                                  "Mean","Median","SD","")
)
),
            matrixResults)


#set a folder to which you want to save the results for each combination of a and b parameters
                                    setwd(’C:/Results’)
                                    write.table(data.frame(matrixResults),
                                                file = paste(’n_is_’, nrPatients[i],
                                                             as.character(prev_Y_or_M[1,b]),
                                                             ’a_is_’,
                                                             a,
                                                             "missOn_",
                                                             prev_Y_or_M[1,b],
                                                             ’.csv’, sep=’’),
                                                sep = ",", col.names = TRUE, row.names=FALSE,
                                                qmethod = "double")
                  }
}

}
```

C.2. Simulations from chapter 3

```
library(mvtnorm)
library(nlme)
library(Hmisc)
library(mclust)
library(reshape2)
library(stringi)


getCondMeans<-function(z){
  z<-data.frame(z[,-c(4)])
  names(z)<-c("mis","Y_t", "Y_tMinus1")
  mod1<-lm(Y_t˜Y_tMinus1, data=z[z$mis==0,])
  mod2<-lm(Y_t˜Y_tMinus1, data=z[z$mis==1,])
  return(c(mean(predict(mod1)), mean(predict(mod2))))
}
#function used for estimating a series of values for observed sample as well as or
#each Bootstrap sample
# it receives x = set of parameters (alpha, rho, treatment group..)
# and the observed or bootstrapped sample dat_obs with dropout indicators

getCorrectedEst_wH<-function(x, dat_obs){
  trArm<-x[1]
  alpha<-x[2]
  rho<-x[3]
  adAlpha<-NULL
  deltaOnLastY<-x[4]
  predictedLI_1<-NULL
  predictedLI_2<-NULL
  meansDropOutsAdhere_marg<-NULL
  fitLogitData<-NULL
```

```
adjustedAlpha<-NULL
meansDropOutsAdhere_cond<-NULL
conditionalOR_Y_t<-NULL
twoBandsOR<-NULL
predictedMeanOfobserved<-NULL
twoBandsOR_t<-NULL
EdgeAndMiddlePortions_ALL<-NULL
EdgeAndMiddlePortions_Obs_Miss<-NULL
namPred<-c("interc", "Y_tminus1", "Y_tminus2", "Y_tminus3","Y_tminus4","Y_tminus5")
sbm<-NULL
for(t in seq(1,6)){
  temp<-dat_obs[dat_obs$R==1 & dat_obs$time==t-1 & dat_obs$Arm==trArm, c(1,5,2,8,9,10,11,12,13)]
  names(temp)[1]<-"pid"
  if(t!=1){
    temp$band<-ifelse(temp$Y_tminus1<100, 1,
                      ifelse(temp$Y_tminus1>170,3,2))
    l<-length(dat_obs[dat_obs$time==t-1 & dat_obs$Arm==trArm, c(1)])
    gh<-matrix(nrow=l, ncol=2)
    gh<-dat_obs[dat_obs$time==t-1 & dat_obs$Arm==trArm, c(1,6)]
    names(gh)[1]<-"pid"
    gh<-merge(gh, temp[, c(1, seq(4, 4+t-1,1), length(names(temp)))], all.x=TRUE)
    if(length(temp[,1])!=l){
      gh[-which(gh[,1]%in%temp[,1]), seq(4, 4+t-2,1)]<-predictedLI_1[-which(gh[,1]%in%temp[,1]),seq(1, length(predictedLI_1[1,]))]
      gh[-which(gh[,1]%in%temp[,1]),length(gh[1,])]<-ifelse(predictedLI_1[-which(gh[,1]%in%temp[,1]),1]<100, 1,
                                               ifelse(predictedLI_1[-which(gh[,1]%in%temp[,1]),1]>170,3,2))
    }
    gh<-data.frame(gh)
    BandsPresent<-unname(sapply(split(gh, gh$R), function(x){
      ty<-c(0,0,0)
      ty[as.numeric(rownames(table(x$band)))]<-as.numeric(rownames(table(x$band)))
      return(ty)}))
    SameBandsPresent<-prod(BandsPresent[,1][BandsPresent[,1]!=0]%in%BandsPresent[,2])
    sbm<-c(sbm, SameBandsPresent)
    #if (t==6 & prod(sbm)==0) print(x)
    if (SameBandsPresent & deltaOnLastY %in% c(1,2) & length(unique(temp$band))>1) {
      mod1<-lm(delta_LDL˜(1+Y_tminus1)*factor(band), data=temp)
      if (prod(BandsPresent[,2][BandsPresent[,2]!=0]%in%BandsPresent[,1])){
        temo<-gh[gh$R==0,]
        temo$out<-seq(1, length(gh[gh$R==0,1]))
        A<-unname(model.matrix(out ˜ (1+Y_tminus1)*as.factor(band), temo))
      } else{
        temo<-gh[gh$R==1,]
        temo$out<-seq(1, length(gh[gh$R==1,1]))
        B<-model.matrix(out ˜ (1+Y_tminus1)*as.factor(band), temo)
        temo<-gh[gh$R==0,]
        temo$out<-seq(1, length(gh[gh$R==0,1]))
        if (length(unique(temo$band))>1){
          C<-model.matrix(out ˜ (1+Y_tminus1)*as.factor(band), temo)
        } else {C<-model.matrix(out ˜ (1+Y_tminus1), temo)
        }
        dj<-which(!unlist(dimnames(B)[2])%in%unlist(dimnames(C)[2]))
        A<-matrix(nrow=length(gh[gh$R==0,1]), ncol=length(unlist(dimnames(B)[2])))
        A[,dj]<-matrix(rep(0, length(dj)*length(gh[gh$R==0,1])), nrow=length(gh[gh$R==0,1]), ncol=length(dj))
        A[,-dj]<-unname(C)
      }
    } else{
      vtemp<-names(temp)[seq(5, 5+t-2)]
      predictors<-gsub(’,’,’’,toString(c(rbind(vtemp, rep("+", t-2)))[-2*(t-1)]))
      out<-names(temp)[4]
      h<-expression(paste("mod1<-lm(",out,"˜",predictors, ", data=temp)", sep=""))
      eval(parse(text=eval(h)))
      temo<-data.frame(gh[gh$R==0, seq(4, 4+t-2)])
  temo$out<-seq(1, length(gh[gh$R==0,1]))
  names(temo)[seq(1, 1+t-2)]<-vtemp
  h<-expression(paste("A<-model.matrix(out~",predictors, ", data=temo)", sep=""))
  eval(parse(text=eval(h)))
}

predictedMeanOfobserved<-c(predictedMeanOfobserved, mean(predict(mod1)))
if (deltaOnLastY==1) {
  adjustedAlpha<-ifelse(A[,2]<100 | A[,2]>170, alpha/4,alpha)
}
if (deltaOnLastY==2) {
  adjustedAlpha<-ifelse(A[,2]<100 | A[,2]>170,
                         ifelse(abs(alpha)+log(abs(alpha))>0,sign(alpha)*round(log(abs(alpha)),1),alpha/2),alpha)
}
delta21<-sign(alpha)*rho^(t-1)*ifelse(rep(deltaOnLastY, length(A[,2]))==2 | rep(deltaOnLastY, length(A[,2]))==1,
                                       abs(adjustedAlpha), abs(alpha))
reSdErr<-sqrt(deviance(mod1)/df.residual(mod1))

gh[-which(gh[,1]%in%temp[,1]),3]<-A%*%unname(coef(mod1))+delta21+rnorm(length(A[,1]),mean=0, reSdErr)



predictedLI_2<-cbind(predictedLI_2, predictedLI_2[,t-1]+gh[,3])
predictedLI_1<-t(apply(predictedLI_2, 1, rev))

tempLogitEstDat<-cbind(1-dat_obs[dat_obs$time==t-1 & dat_obs$Arm==trArm, c(6)], predictedLI_1[,c(1,2)])
tempLogitEstDat<-cbind(tempLogitEstDat, ifelse(tempLogitEstDat[,2]<100 | tempLogitEstDat[,2]>170, 1, 0), t)
meansDropOutsAdhere_marg<-rbind(meansDropOutsAdhere_marg, aggregate(tempLogitEstDat[,2], by=list(tempLogitEstDat[,1]), mean)[,2])
meansDropOutsAdhere_cond<-rbind(meansDropOutsAdhere_cond, getCondMeans(tempLogitEstDat))
tempBands<-ifelse(tempLogitEstDat[,2]<100, 1,
                  ifelse(tempLogitEstDat[,2]>170, 3,2))
if (length(which(!c(1,2,3)%in%unique(tempBands)))==0){
  EdgeAndMiddlePortions_ALL<-rbind(EdgeAndMiddlePortions_ALL, round(unname(table(tempBands)/sum(table(tempBands))), 2))
} else { temx<-ifelse(rep(which(!c(1,2,3)%in%unique(tempBands)),3)==1, c(0,round(unname(table(tempBands)/length(tempBands)),2)),
             ifelse(rep(which(!c(1,2,3)%in%unique(tempBands)),3)==2, c(round(unname(table(tempBands)/length(tempBands))[1],2), 0,
                                           round(unname(table(tempBands)/length(tempBands))[2],2)),
                 c(round(unname(table(tempBands)/length(tempBands))[1],2), round(unname(table(tempBands)/length(tempBands))[2],2),0)))

EdgeAndMiddlePortions_ALL<-rbind(EdgeAndMiddlePortions_ALL, temx)



}
if (length(which(!c(1,2,3)%in%unique(tempBands[tempLogitEstDat[,1]==0])))==0){
  temx1<-round(unname(table(tempBands[tempLogitEstDat[,1]==0])/sum(table(tempBands[tempLogitEstDat[,1]==0]))), 2)
} else{
  missProps<-which(!c(1,2,3)%in%unique(tempBands[tempLogitEstDat[,1]==0]))
  temx1<-c(0,0,0)
  temx1[-missProps]<-round(unname(table(tempBands[tempLogitEstDat[,1]==0])/length(tempBands[tempLogitEstDat[,1]==0])),2)
}
if (length(which(!c(1,2,3)%in%unique(tempBands[tempLogitEstDat[,1]==1])))==0){
  temx2<-round(unname(table(tempBands[tempLogitEstDat[,1]==1])/sum(table(tempBands[tempLogitEstDat[,1]==1]))), 2)
} else{
  missProps<-which(!c(1,2,3)%in%unique(tempBands[tempLogitEstDat[,1]==1]))
  temx2<-c(0,0,0)
  temx2[-missProps]<-round(unname(table(tempBands[tempLogitEstDat[,1]==1])/length(tempBands[tempLogitEstDat[,1]==1])),2)
}
EdgeAndMiddlePortions_Obs_Miss<-rbind(EdgeAndMiddlePortions_Obs_Miss, c(c(temx1), c(temx2)))

fitLogitData<-rbind(fitLogitData, tempLogitEstDat)
tempLogitEstDat<-data.frame(tempLogitEstDat)
names(tempLogitEstDat)<-c("dropInd", "Y_t", "Y_tminus1","Y_tTwo_bands")

if (alpha!=0) tempLogitEstDat[, c(2,3)]<-tempLogitEstDat[,c(2,3)]/abs(alpha)
```

```
    tag1<-0
    tryCatch(mod11<-glm(dropInd~-1+Y_t*factor(Y_tTwo_bands), data=tempLogitEstDat, family=binomial),
             error= function(err){
               print(paste(err))
               tag1<-1
             },
             warning=function(w){
               w_nings_TwoBands_OP<-rbind(w_nings_TwoBands_OP, c(w, paste(w)))
             })
    tag2<-0
    h<-expression(paste("mod2<-glm(dropInd~-1+Y_t*factor(Y_tTwo_bands) + Y_tminus1",
                                               ", data=tempLogitEstDat, family=binomial)", sep=""))
    tryCatch(eval(parse(text=eval(h))),
             error= function(err){
               print(paste(err))
               tag2<-1
             },
             warning=function(w){
               w_nings_TwoBands_OP<-rbind(w_nings_TwoBands_OP, c(w, paste(w)))
             })
    if (tag1==1 | tag2==1 | !exists("mod2") | !exists("mod11")) {twoBandsOR_t<-cbind(twoBandsOR_t, c(NA, NA, NA, NA))
    } else{twoBandsOR_t<-cbind(twoBandsOR_t, c(coef(mod11)[1], coef(mod11)[4], coef(mod2)[1], coef(mod2)[5]))}
  } else {
    mod1<-lm(delta_LDL~1,data=temp)
    predictedMeanOfobserved<-c(predictedMeanOfobserved, mean(predict(mod1)))
    predictedLI_1<-cbind(predictedLI_1, temp$delta_LDL)
    predictedLI_2<-cbind(predictedLI_2, temp$delta_LDL)
    hgd<-ifelse(temp$delta_LDL<100, 1, ifelse(temp$delta_LDL>170, 3,2))
    temx<-ifelse(rep(which(!c(1,2,3)%in%unique(hgd)),3)==1, c(0,unname(table(hgd)/length(hgd))),
                 ifelse(rep(which(!c(1,2,3)%in%unique(hgd)),3)==2, c(unname(table(hgd)/length(hgd))[1],
                                                                  0, unname(table(hgd)/length(hgd))[2]),
                        c(unname(table(hgd)/length(hgd))[1], unname(table(hgd)/length(hgd))[2],0)))
    EdgeAndMiddlePortions_ALL<-rbind(EdgeAndMiddlePortions_ALL, temx)
    EdgeAndMiddlePortions_Obs_Miss<-rbind(EdgeAndMiddlePortions_Obs_Miss, rep(temx,2))
    meansDropOutsAdhere_marg<-rbind(meansDropOutsAdhere_marg, c(mean(temp$delta_LDL), mean(temp$delta_LDL)))
    meansDropOutsAdhere_cond<-rbind(meansDropOutsAdhere_cond, c(mean(temp$delta_LDL), mean(temp$delta_LDL)))
    twoBandsOR_t<-cbind(twoBandsOR_t, c(1,1,1,1))
  }
}

adAlpha<-rbind(adAlpha,c(alpha,ifelse(abs(alpha)+log(abs(alpha))>0, sign(alpha)*round(log(abs(alpha))),alpha/2)))

fitLogitData<-data.frame(fitLogitData)
names(fitLogitData)<-c("dropInd", "Y_t", "Y_tminus1", "Y_tTwo_bands", "time")
if (alpha!=0)  fitLogitData[, c(2,3)]<- fitLogitData[,c(2,3)]/abs(alpha)
tag1<-0
tryCatch(mod12<-glm(dropInd~-1+as.factor(time)+Y_t*factor(Y_tTwo_bands), data=fitLogitData, family=binomial),
         error= function(err){
           print(paste(err))
           tag1<-1
         },
         warning=function(w){
           w_nings_TwoBands_OP<-rbind(w_nings_TwoBands_OP, c(w, paste(w)))
         })
tag2<-0
h<-expression(paste("mod21<-glm(dropInd~-1+as.factor(time)+Y_t*factor(Y_tTwo_bands) + Y_tminus1",
                                                                           ", data=fitLogitData, family=binomial)", sep=""))
tryCatch(eval(parse(text=eval(h))),
         error= function(err){
           print(paste(err))
           tag2<-1
         },
           warning=function(w){
             w_nings_TwoBands_OP<-rbind(w_nings_TwoBands_OP, c(w, paste(w)))
           })
  if (tag1==1 |tag2==1 | !exists("mod21")| !exists("mod12")) {twoBandsOR<-unname(t(as.matrix(c(NA, NA, NA, NA))))
  } else{twoBandsOR<-unname(t(as.matrix(c(coef(mod12)[6], coef(mod12)[8],
                                                 unname(summary(mod21)$coefficients[4,4]), coef(mod21)[6], coef(mod21)[9]))))}

  tag<-0
  h<-expression(paste("mod13<-glm(dropInd˜Y_t + Y_tminus1", ", data=fitLogitData, family=binomial)", sep=""))
  tryCatch(eval(parse(text=eval(h))),
           error=function(err){
             print(paste(err))
             tag<-1
           },
           warning=function(w){
             w_nings_conditional_OP<-rbind(w_nings_conditional_OP, c(w, paste(w)))
           })
  if(tag==1 | !exists("mod13")){
    conditionalOR_Y_t<-c(conditionalOR_Y_t, NA)
  } else { conditionalOR_Y_t<-unname(t(as.matrix(c(exp(coef(mod13))[2]))))}

  return(cbind(apply(predictedLI_2,2,mean),
               apply(predictedLI_2,2,sd),
               meansDropOutsAdhere_marg,
               predictedMeanOfobserved,
               meansDropOutsAdhere_cond,
               conditionalOR_Y_t[rep(1,6), ],
               twoBandsOR[rep(1,6), ],
               t(twoBandsOR_t),
               EdgeAndMiddlePortions_ALL,
               EdgeAndMiddlePortions_Obs_Miss,
               c(sbm,1)))
}

makeLongBootData<-function(y,rep){
  y<-y[order(y$Arm, y$"dropOutTime"), ]
  hg<-which(y$"dropOutTime"!=Lag(y$"dropOutTime", 1) | y$"dropOutTime"!=Lag(y$"dropOutTime", -1))
  temp<-matrix(c(1, hg, length(y$"dropOutTime")), ncol=2, byrow=TRUE)
  jh<-apply(temp, 1, function(x){
    return(sample(seq(x[1],x[2]),x[2]-x[1]+1, replace=rep))
  }
  )
  jh<-unlist(jh)
  bootObsSample<-y[jh, ]
  bootObsSample<-cbind(seq(1, length(y[,1])), bootObsSample[order(bootObsSample$Participant),])
  names(bootObsSample)[c(1,2)]<-c("Participant", "ParticipantTrue")
  longBoot<-melt(bootObsSample[,-which(names(bootObsSample)=="dropOutTime")],
                 id=c("Participant", "ParticipantTrue", "Arm"))[,c(1,5,4,3,2)]
  longBoot<-longBoot[order(longBoot$Participant,longBoot$Arm),]
  longBoot$time<-as.integer(stri_sub(longBoot$variable, 2,2))
  names(longBoot)[2]<-"LDL_dropout"
  longBoot<-longBoot[,-c(which(names(longBoot)=="variable"))]
  longBoot$R<-ifelse(!is.na(longBoot$LDL_dropout),1,0)
  temp<-NULL
  for (i in unique(longBoot$Participant)){
    x<-longBoot$LDL_dropout[longBoot$Participant==i]
    temp<-c(temp,c(NA,x[-6]))
  }
  longBoot$LDL_prev<-temp

  longBoot$delta_LDL<-longBoot$LDL_dropout
  longBoot$delta_LDL[longBoot$time!=0]<-longBoot$delta_LDL[longBoot$time!=0]-longBoot$LDL_prev[longBoot$time!=0]
  tm<-longBoot[,c(1,6,2)]
  longBoot$Y_tminus1<-c(unname(unlist(sapply(split(tm[,3],  tm$Participant),Lag, 1))))
  longBoot$Y_tminus2<-c(unname(unlist(sapply(split(tm[,3],  tm$Participant),Lag, 2))))
  longBoot$Y_tminus3<-c(unname(unlist(sapply(split(tm[,3],  tm$Participant),Lag, 3))))
  longBoot$Y_tminus4<-c(unname(unlist(sapply(split(tm[,3],  tm$Participant),Lag, 4))))
  longBoot$Y_tminus5<-c(unname(unlist(sapply(split(tm[,3],  tm$Participant),Lag, 5))))

  longBoot$dropout<-c(sapply(split(longBoot[,c(1,6,7)],  longBoot$Participant),
                             function(x){
                               return(rep(ifelse(length(which(x$R%in%c(0)))==0, 7,which(x$R%in%c(0))), 6))}))

  return(longBoot)
}




w_nings_conditional_OP<-NULL
w_nings_marginal_OP<-NULL
w_nings_TwoBands_OP<-NULL
## read in observed once out forever out modified data for 2 groups
## 1=treatment 4= control, remember to set work. dir
## so that this dataset can be read without errors
Y_Mis<- data.frame(read.csv("WideDataForm_Modified_OnceOutForeverOut_group1_4.csv", header=T), stringsAsFactors=FALSE)

sensParamTemp<-expand.grid(c(1,4), seq(-25, 25, 0.5), c(1), c(0))
## DeltaDependsOnLast_Y is a parameter that determines if alpha depends on last observed value or
## not (we only use the part of the getCorrectedEst_wH function that assumes constant shift alpha)

names(sensParamTemp)<-c("trArm", "alpha", "rho", "DeltaDependsOnLast_Y")
sensParamTemp<-sensParamTemp[order(sensParamTemp$trArm,sensParamTemp$DeltaDependsOnLast_Y),]
rownames(sensParamTemp)<-seq(1, length(sensParamTemp[,1]))
Z<-as.list(data.frame(t(matrix(unlist(sensParamTemp), nrow=length(sensParamTemp[,1]), ncol=4, byrow=FALSE))))

#####################################################################################################################
#####################################################################################################################
#####################################################################################################################
########## this is a part of the code that runs and calculates
########## unignorable LI estimate for each value of alpha and for each treatment group for observed data Y_Mis only
############
set.seed(1000)
## write results in this folder
setwd('C:Results')
## run makeLongBootData function with rep=FALSE if you want to get observed data set
## run makeLongBootData function with rep=TRUE if you want to get a single bootstrap data set

t1<-makeLongBootData(Y_Mis, FALSE)

tryCatch(kj_variableModFor0<-do.call(rbind,lapply(X=Z, FUN=getCorrectedEst_wH, t1)),
         error =function(err){
           print(paste(err))
           tag<-1
         })

rownames(kj_variableModFor0)<-NULL
ds1<-sensParamTemp[rep(seq(1, length(sensParamTemp[,1])), rep(6, length(sensParamTemp[,1]))), -c(3)]
ds1<-cbind(rep(c(1,2,3,4,5,6), length(sensParamTemp[,1])),
           ds1)
ds_WH<-cbind(ds1, kj_variableModFor0)

ds_WH<-data.frame(ds_OnlyPrevOutcome)
```

```
names(ds_WH)<-c("time", "TrArm", "alpha", "OnLastY", "mean", "stErr", "meanObserved_marg","meanMissing_marg",
                             "PredictedMeanObserved","meanObserved_Cond", "meanMissing_Cond",
                             "conditionalORSingle", "logOR_main_1", "logORInteraction_1",
                             "pValue_Interaction", "logOR_main_2", "logORInteraction_2",
                             "logOR_main", "logORInteraction","logOR_main_adjForprevY",
                             "logORInteraction_adjForprevY", "100andLess", "100To170",
                             "170AndMore", "100andLessObs", "100To170Obs", "170AndMoreObs",
                             "100andLessMis", "100To170Mis", "170AndMoreMis", "overlapBandsObsMis")


write.table(ds_WH,
            file = "FullHist_TrArm14_delta_0_Shift_25_to25_per0_5.csv",
            sep = ",", col.names = TRUE, row.names=FALSE,
            qmethod = "double")

#####################################################################################################################
#####################################################################################################################
#####################################################################################################################


#####################################################################################################################
#####################################################################################################################
#####################################################################################################################
########## this is a part of the code that runs possibly 10 000 times
########## it samples with replacement from Y_Mis each time and calculates for
########## each such bootstrap dataset a value of LI non-ignorable estimate
########## for each value of alpha
########## This part of the code writes out results in 10 datasets per 1000 result-sets
########## Wrning: runtime can be quite long so try measuring it for
########## some small number of iterations so you can asses the runtime for 10 000


set.seed(1000)
te2<-NULL

randseeD<-NULL

system.time(
  for (s in seq(1,10000)){
    oldseed <- NULL
    if (exists(".Random.seed"))
      oldseed <- .Random.seed
    t1<-makeLongBootData(Y_Mis, TRUE)
    tag<-0
### there is a series of error catching maneveurs
### if the code errors out and stops the last seed is kept
### so that it can be restarted for that set of parameters
### that it errored out for
    tryCatch(kj<-do.call(rbind,lapply(X=Z, FUN=getCorrectedEst_wH, t1)),
             error =function(err){
               print(paste(err))
               print("rbindLApllapply(X=Z, FUN=getCorrectedEst_wH, t1)")
               print(s)
               tag<-1
             })
    if (tag==1){ write.table(data.frame(tail(randseeD, 2)),
                             file = eval(expression(paste("SeedByError.csv",sep=""))), append=TRUE,
                             sep = ",", col.names = TRUE, row.names=FALSE,
                             qmethod = "double")
    }
    rownames(kj)<-NULL
    ds1<-sensParamTemp[rep(seq(1, length(sensParamTemp[,1])), rep(6, length(sensParamTemp[,1]))), ]
    ds1<-cbind(rep(rep(s, 6),length(sensParamTemp[,1])),
               rep(c(1,2,3,4,5,6), length(sensParamTemp[,1])),
               ds1)
    tag<-0
    tryCatch(randseeD<-rbind(randseeD,oldseed),
             error =function(err){
               print(paste(err))
               print("rbind(randseeD,oldseed)")
               tag<-1
             })
    if (tag==1) {write.table(data.frame(tail(randseeD, 2)),
                             file = eval(expression(paste("SeedByError.csv",sep=""))), append=TRUE,
                             sep = ",", col.names = TRUE, row.names=FALSE,
                             qmethod = "double")}
    tag<-0
    ds_WH<-cbind(ds1, kj)
    tryCatch(te2<-rbind(te2, ds_WH),
             error =function(err){
               print(paste(err))
               print("rbind(te2, ds_WH)")
               print(names(te2))
               print(names(ds_WH))
               print(s)
               tag<-1
             })
    if (tag==1) { write.table(data.frame(tail(randseeD, 2)),
                              file = eval(expression(paste("SeedByError.csv",sep=""))), append=TRUE,
                              sep = ",", col.names = TRUE, row.names=FALSE,
                              qmethod = "double")
    }
    if(s%%100==0){
      write.table(data.frame(randseeD),
                  file = eval(expression(paste("Seeds_", ((s-1)%/%1000)+1,".csv",sep=""))), append=TRUE,
                  sep = ",", col.names = FALSE, row.names=FALSE,
                  qmethod = "double")

      remove(randseeD)
      randseeD<-NULL
    }
    if (s%%1000==0){
      names(te2)<-c("bootSample", "time", "TrArm", "alpha", "rho", "DeltaDependsOnLast_Y",
                    "mean", "stErr", "meanObserved_marg","meanMissing_marg",
                    "PredictedMeanObserved","meanObserved_Cond", "meanMissing_Cond",
                    "conditionalORSingle", "logOR_main_1", "logORInteraction_1","pValue_Interaction",
                    "logOR_main_2",  "logORInteraction_2",
                    "logOR_main", "logORInteraction","logOR_main_adjForprevY",
                    "logORInteraction_adjForprevY", "100andLess", "100To170",
                    "170AndMore", "100andLessObs", "100To170Obs", "170AndMoreObs",
                    "100andLessMis", "100To170Mis", "170AndMoreMis", "overlapBandsObsMis")

      write.table(te2,
                  file = eval(expression(paste("Results_", s%/%1000,".csv",sep=""))),
                  sep = ",", col.names = TRUE, row.names=FALSE,
                  qmethod = "double")
      remove(te2)
      te2<-NULL
    }
  })
remove(ls())

########################################################################################################################
#########################################################################################################################
```

```
#######################################################################################################################
########################################################################################################################
###############################################################################################################################
##### this program makes TrEff_j.csv data sets out of Results_1, ..., Results_j, j=1,...10 datasets.
##### These data sets save all the estimates from 1000 Bootstrapped samples EACH, so TrEff_j.csv
####  collects data out of which one can make Boot CI's  that come from j*1000 bootstrapped datasets,
####  the TrEff_j datasets are used to plot the sensitivity map
################################################################################################################################
############### set folder from which Results_1, ..., Results_k are to be read
####### we need to set some global variables
####### timeOfInterest is a time from 1 to 6, depending on outcome at which time point we wish to plot
####### we set timeOfInterest, default is the last timepoint 6
####### you can decide how many bootstrap samples you want to use, seting nrSam t0 10
####### uses estimates from 10 000 bootstrap samples
timeOfInterest<-6
nrSam<-10
for (k in seq(1, nrSam)){
    h<-expression(paste("Results<-data.frame(read.csv('Results_",k, ".csv', header=T), stringsAsFactors=FALSE)", sep=""))
    system.time(eval(parse(text=eval(h))))
    #see documentation in makePlotData_withBootstrap_TreatGrupMean_Miss_Obs.R
    Results<-Results[Results$time==timeOfInterest, c(1,3,4,7,9,10)]
    #pj<-pj[pj$time==6,]

    system.time(temps<-lapply(split(Results,
                                    list(Results$bootSample)
                                    ),
                              function(x){
                                ## since we are making BOOT Ci's for purpose of testing the effect here
                                ## by checking if Boot Ci' of Treatment overlaps in any way with 95% boot Ci'
                                ## of the placebo we have to do this for each point in the alphaPlacebo X alphaTreatemnt grid
                                ## so we make the matrix of every possible combination of alphaTreatment times alphaPlacebo
                                ## means so mean in treatment for alpha =-15 Minus mean of LDl in placebo for alpha=10
                                               dtem1<-cbind(expand.grid(x$alpha[x$TrArm==1],x$alpha[x$TrArm==4]),
                                               expand.grid(x$mean[x$TrArm==1], x$mean[x$TrArm==4]))
                                              #  save this matrix in a matrix with
                                              #  first two columns are the coordinates of alphaTreat X alphaPl
                                              # 3rd column is the treatment effect Mean in Treatment - Mean in Placebo
                                              # 4th column is the number of the bootsample this difference corresponds to
                                              # since we do this for every bootsample in one Results_j we will get
                                              # 1000 differences for 1 alphaTR X alphaPl combination
                                              dsa<-cbind(dtem1[, c(1,2)],
                                                         dtem1[,3]-dtem1[,4],
                                                         rep(1000*(k-1)+x$bootSample[1],length(dtem1[,4]))
                                                         )
                                  return(dsa)}
                              )
                )

    ###make a matrix out of the list, matrix has 1000 X |alphaTr| X |alphaPl| rows and 4 columns

    system.time(temps<-do.call(rbind, temps))
    ## remove current Results data since we used this Results_k
    remove(Results)
    names(temps)<-c( "alphaTr", "alphaPl", "trEff", "boot")
    ## make 1 variable denoting which combination of alphaTr _ alphaPl the row is,
    ### we separate say 0.5 and -0.5 by "0.5_-0.5"
    system.time(temps$alphaTrAlphaPl<-apply(temps[, c(1,2)], 1, function(x){return(paste(x[1], x[2], sep="_"))}))
    #sort the matrix by the alphaTr and then alphaPl
    temps<-temps[order(temps$alphaTr,temps$alphaPl),]

    system.time(temps1<-lapply(split(temps,
                                     list(temps$alphaTrAlphaPl)),
                               function(x){
                                 # for each combination, basically each 1000 rows in temp
                                 # order treatment effect
                                            #sdBoot<-sd(x$trEff)
                                            temp<-x$trEff[order(x$trEff)]
                                            # CIBoot_left<-temp[2]
                                            #CIBoot_right<-temp[23]

                                            #check if it is longer than 250, if not save all if yes save bottom/top 250
                                            # see explanation in makePlotData_withBootstrap_TreatGrupMean_Miss_Obs.R
                                            if (length(temp)<251){
                                              CIBoot_left<-temp
                                              CIBoot_right<-temp
                                            } else {
                                              CIBoot_left<-temp[seq(1,250)]
                                              CIBoot_right<-temp[seq(length(temp)-249,length(temp))]
                                            }
                                            ## save a 250 times 4 matrix, left, right values
                                            ## and 3rd and 4th is the alpha combination for which these 250 values are saved
                                            ## and save the number of BootSamples these values are coming
                                            ## from it can be 250 (most of the time)
                                            ## or less
                                            dsa<-cbind(CIBoot_left,
                                                       CIBoot_right,
                                                       rep(x$alphaTrAlphaPl[1],length(CIBoot_right)),
                                                       rep(length(temp), length(CIBoot_right)))
                                            return(dsa)}
                                          ))

    system.time(temps1<-do.call(rbind, temps1))
    system.time(temps1<-data.frame(temps1))
    system.time(names(temps1)<-c("CIBoot_L", "CIBoot_R", "alphaTrAlphaPl", "SamplesNot0"))

    #save these in a file
#
    if (k<2){
      system.time(write.table(temps1,
                              file = eval(expression(paste("TrEff_1.csv",sep=""))), append=TRUE,
                              sep = ",", col.names = TRUE, row.names=FALSE,
                              qmethod = "double"))
    }
    if (k<3){
      system.time(write.table(temps1,
                              file = eval(expression(paste("TrEff_2.csv",sep=""))), append=TRUE,
                              sep = ",", col.names = TRUE, row.names=FALSE,
                              qmethod = "double"))
    }
    system.time(write.table(temps1,
                            file = eval(expression(paste("TrEff_10.csv",sep=""))), append=TRUE,
                            sep = ",", col.names = TRUE, row.names=FALSE,
                            qmethod = "double"))
    remove(temps, temps1)
  }



########################################################################################################################
#########################################################################################################################
########################################################################################################################
#########################################################################################################################
#### read in data for plotting made by the previous script program
#### we can read in data to make Boot CI’s with 1000 (TrEff_1.csv), 2000 (TrEff_2.csv)
```

```
#### and 10 000 (TrEff_10.csv)
#### structure of the data to read in is
#### c("CIBoot_L", "CIBoot_R", "alphaTrAlphaPl", "SamplesNot0"))
system.time(d<-data.frame(read.csv("TrEff_10.csv", header=T), stringsAsFactors=FALSE))
# separate alphaTrAlphaPl into two columns so that we can pick out only those
# combinations from 10 to -10
#
system.time(tempS<-do.call(rbind,
                           lapply(d$alphaTrAlphaPl,
                                  function(x){
                                              t<-stri_locate_first_fixed(as.character(x), "_")[1]
                                              alphaTr_<-as.numeric(stri_sub(as.character(x), 1,t-1))
                                              alphaPl_<-as.numeric(stri_sub(as.character(x), t+1, nchar(as.character(x))))
                                              return(c(alphaTr_, alphaPl_))
                                          }
                                  )
                           )
            )


d<-cbind(d, tempS)
remove(tempS)
names(d)[c(length(names(d))-1,length(names(d)))]<-c("alphaTr", "alphaPl")

##pick out only those from 10 to -10
d<-d[abs(d$alphaTr)<10.5 & abs(d$alphaPl)<10.5, ]
#sort lexically on alphaTralphaPl combination
d<-d[order(d$alphaTrAlphaPl), ]

### make a new alphaTralphaPl variable that only saves those 10 to -10
# because the old one will remain a factor with levels -25 25
# even if we remove values bigger than 10 and smaller than -10

system.time(d$alphaTrAlphaPl<-apply(d[, c(length(names(d))-1,length(names(d)))],
                                    1,
                                    function(x){
                                                return(paste(x[1], x[2], sep="_"))
                                      }
                                    )
            )
## set the number of Result_j datasets that was used to make the data set TrEff_j we read in
nrSam<-2

# pick out 25*nrSam from sorted left and right values from TrEff for each alphaTralphaPl combination
# this will get you 95/% Ci w.r.t. which number of Results_j datasets was used to make TrEff
system.time(BootCIS<-lapply(split(d,
                                  list(d$alphaTrAlphaPl)),
                                  function(x){
                                    temp<-as.numeric(as.character(x$CIBoot_L))
                                    temp<-temp[order(temp)]
                                              CIBoot_left<-temp[nrSam*25]
                                              temp<-as.numeric(as.character(x$CIBoot_R))
                                              temp<-temp[order(temp)]
                                              CIBoot_right<-temp[length(temp)-(nrSam*25-1)]
                                              # make a 3 column row vector: alphaTralphaPl combination
                                              #                             left value
                                              #                             right value
                                              dsa<-c(toString(x$alphaTrAlphaPl[1]),
                                                     CIBoot_left,
                                                     CIBoot_right)
                                              return(dsa)}))
```

```
#make a matrix out of the list: number of different alphaTralphaPl combinations X 250 rows and 3 columns
system.time(BootCIS<-do.call(rbind, BootCIS))

BootCIS<-BootCIS[!is.na(BootCIS[,2]),]


BootCIS<-data.frame(BootCIS)
#separate alphaCombinations into 2 columns
AlphaComb<-do.call(rbind, lapply(rownames(BootCIS),  function(x){
  t<-stri_locate_first_fixed(x, "_")[1]
  alphaTr_<-as.numeric(stri_sub(x, 1,t-1))
  alphaPl_<-as.numeric(stri_sub(x, t+1, nchar(x)))
  return(c(alphaTr_, alphaPl_))}))

# merge with BOOT CI's for each alphaCombination
BootCIS<-cbind(AlphaComb, BootCIS)
names(BootCIS)<-c("alphaTr", "alphaPl", "alphaCom", "CIBoot_L", "CIBoot_R")
rownames(BootCIS)<-NULL
BootCIS$CIBoot_L<-as.numeric(as.character(BootCIS$CIBoot_L))
BootCIS$CIBoot_R<-as.numeric(as.character(BootCIS$CIBoot_R))
BootCIS<-BootCIS[,-c(length(names(BootCIS))-1,length(names(BootCIS)))]


### import the mean estimate per treatment group and per alpha parameter
res<-data.frame(read.csv("FullHist_TrArm14_delta_0_Shift_25_to25_per0_5.csv.csv", header=T), stringsAsFactors=FALSE)
names(res)[3]<-"alpha_"
names(res)[4]<-"OnLastY"
names(res)[3]<-"alpha"

ds_WH<-res[res$time==6 &  res$"OnLastY"%in%c(0) & res$alpha%in%seq(-10,10,0.5), c(2,3,5)]


treaEffects_combAlpha<-cbind(expand.grid(ds_WH$alpha[ds_OnlyPrevOucome$TrArm==1],ds_WH$alpha[ds_OnlyPrevOucome$TrArm==4]),
                             expand.grid(ds_WH$mean[ds_OnlyPrevOucome$TrArm==1], ds_WH$mean[ds_OnlyPrevOucome$TrArm==4]))

### make alphaCombination variable for the original data so that we can merge with Boot
## Ci's with the same alphaTralphaPl combination
treaEffects_combAlpha$alphaCom<-unname(apply(treaEffects_combAlpha[, c(1,2)], 1,
                                                           function(x){return(paste(x[1], x[2], sep="_"))}))
## make treat. effect estimates for each alphas combination
treaEffects_combAlpha$effect<-treaEffects_combAlpha[,3]-treaEffects_combAlpha[,4]
## merge these with their BOOT Ci's
BootCIS<-merge(BootCIS, treaEffects_combAlpha[, c(5,6)], by="alphaCom")
names(BootCIS)[6]<-"trEff"
### make variable that decides if the direction:
### -1 = kept significant (direction for this alphacombination is negative which means tratment lowered
###                        LDL more than placebo and it is also significant )
### 1 = treatment effect is opposite of the one expected placebo lowered LDL more than treatment
###     and the effect is significant
### 0 = treatment effect is any direction but not significant
###
keepDirection_signif_<-ifelse(sign(BootCIS$CIBoot_L)==sign(BootCIS$CIBoot_R) & sign(BootCIS$CIBoot_L)==(-1), -1,
                              ifelse(sign(BootCIS$CIBoot_L)==sign(BootCIS$CIBoot_R) & sign(BootCIS$CIBoot_L)==(1),1,0))

### make intermediate data for plotting
dsa<-cbind(BootCIS[, c(2,3)], BootCIS$trEff, BootCIS[, c(4,5)], keepDirection_signif_)
dsa<-cbind(BootCIS[, c(2,3)], BootCIS[,11], BootCIS[, c(4,5)], keepDirection_signif_)

names(dsa)<-c( "alphaTr", "alphaPl", "trEff", "leftBootCI", "rightBootCI", "keepdirSign_BootCI")
##### label the direction with creating new variable aligned with the keepDirection_signif_

dsa$Direction_Boot<-ifelse(dsa$keepdirSign_BootCI==-1, "Preserved direction",
                          ifelse(dsa$keepdirSign_BootCI==0, "Not significant", "Changed direction"))


plotD_Alpha_const<-dsa[which(dsa$alphaTr%in%(seq(-10,10,0.5)) & dsa$alphaPl%in%(seq(-10,10,0.5)) ),]
plotDP_Alpha_const<-dsa[which(dsa$alphaTr%in%(seq(-10,10,5)) & dsa$alphaPl%in%(seq(-10,10,5)) ),]

##########################################################################################################################
###########################################################################################################################
##########################################################################################################################
###########################################################################################################################
##########################################################################################################################
############# Treatment effect sensitivity plot for constant shift alpha -10 to 10 per 0.5
#########################################################################################################################
#10 to -10, Boot CI
v <- ggplot(plotD_Alpha_const, aes(alphaTr,alphaPl, z=trEff))
v2<-v + #geom_point(data=plotD_Alpha_const, aes(alphaTr,alphaPl, colour=round(zval, 1),
  #shape=factor(Direction)), size=2.4)+
  geom_point(data=plotD_Alpha_const, aes(alphaTr,alphaPl, colour=factor(Direction_Boot),
                                        shape=factor(Direction_Boot)), size=5.4)
v2<-v2+  stat_contour(data=plotDP_Alpha_const, size=1.5, breaks=seq(0, -16, -4), colour="black")
v2<-v2 + geom_text(aes(fontface=2), x=-9, y=2, label="-16",size=8, colour="black")+
  geom_text(aes(fontface=2), x=-9, y=-6, label="-12",size=8 ) +
  geom_text(aes(fontface=2), x=-9, y=-9.5, label="-8",size=8 ) +
  geom_text(aes(fontface=2), x=-3.2, y=-9.5, label="-4",size=8)+
  geom_text(aes(fontface=2), x=8.9, y=-9.5, label="0",size=8)+
  theme_bw()+
  scale_color_manual(values=c("darkorange3", "gold"),"Effect")+
  scale_shape(name="Effect")+
  theme(plot.title = element_text(size = rel(2)))+
  xlab(expression(paste(delta[21], "( ", bar(LDL[1])," ;", bold(alpha),") Shared Incentive", sep=" "))) +
  ylab(expression(paste(delta[21], "( ", bar(LDL[1])," ;", bold(alpha),") Control", sep=" ")))+
  theme(legend.title = element_text(colour="black", size=20, face="bold"))+
  theme(legend.text = element_text(colour="black", size=14, face="bold"))+
  theme(legend.text = element_text(colour="black", size=14, face="bold"))+
  theme(axis.title.x = element_text(size = 25), axis.title.y = element_text(size = 25),
        axis.text = element_text(size = 20))+
  labs(title = "Treatment effect (shift constant)")+
  theme(plot.margin = unit(c(0.8, 0.8, 0.8, 0.8), "cm"))
```

BIBLIOGRAPHY

Aalen, O (1980). A Model for Nonparametric Regression Analysis of Counting Processes. In: *Mathematical Statistics and Probability Theory*. Ed. by W Klonecki, A Kozek, and J Rosiski. Vol. 2. Lecture Notes in Statistics. Springer New York, 1–25.

Aalen, OO (2012). Armitage lecture 2010: Understanding treatment effects: the value of integrating longitudinal data and survival analysis. *Statistics in Medicine* 31.18, 1903–1917.

Aalen, OO and Gunnes, N (2010). A dynamic approach for reconstructing missing longitudinal data using the linear increments model. *Biostatistics* 11.3, 453–472.

Aalen, OO, Andersen, PK, Borgan, Ø, Gill, RD, and Keiding, N (2009). History of applications of martingales in survival analysis. *Electronic Journal for History of Probability and Statistics* 5.1, 1–28.

Commenges, D, Ggout-petit, A, Victor, U, and Bordeaux, S (2008). *Likelihood inference for incompletely observed stochastic processes: ignorability conditions*.

Daniels, MJ and Hogan, JW (2008). *Missing Data in Longitudinal Studies: Strategies for Bayesian Modeling and Sensitivity Analysis*. Vol. 1. Chapman and Hall/CRC.

Dawid, AP and Didelez, V (2010). Identifying the consequences of dynamic treatment strategies: A decision-theoretic overview. *Statistics Surveys* 4, 184–231.

Diaz, I and Laan, MJ van der (2013). Sensitivity Analysis for Causal Inference under Unmeasured Confounding and Measurement Error Problems. *The International Journal of Biostatistics* 9.2, 149–160.

Diggle, P, Farewell, D, and Henderson, R (2007). Analysis of longitudinal data with drop-out: objectives, assumptions and a proposal. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 56.5, 499–550.

Eichler, M and Didelez, V (2010). On Granger causality and the effect of interventions in time series. *Lifetime Data Analysis* 16.1, 3–32.

Fleming, TR and Harrington, DP (2011). *Counting processes and survival analysis*. Vol. 169. John Wiley & Sons.

Florens, JP and Mouchart, M (1982). A Note on Noncausality. *Econometrica* 50.3, 583–591.

Florens, J-P and Mouchart, M (1985). CONDITIONING IN DYNAMIC MODELS. *Journal of Time Series Analysis* 6.1, 15–34.

Florens, JP, Mouchart, M, and Rolin, JM (1993). Noncausality and Marginalization of Markov Processes. *Econometric Theory* 9.2, 241–262.

Fosen, J, Ferkingstad, E, Borgan, r, and Aalen, O (2006). Dynamic path analysis a new approach to analyzing time-dependent covariates. *Lifetime Data Analysis* 12.2, 143–167.

Granger, CWJ (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica* 37.3, 424–438.

Gunnes, N, Farewell, DM, Seierstad, TG, and Aalen, OO (2009). Analysis of censored discrete longitudinal data: Estimation of mean response. *Statistics in Medicine* 28.4, 605–624.

Joffe, MM, Yang, PW, and Feldman, HI (2010). Selective Ignorability Assumptions in Causal Inference. *The International Journal of Biostatistics* 6.2, 1–25.

Koopmans, TC and Reiersol, O (1950). The Identification of Structural Characteristics. *The Annals of Mathematical Statistics* 21.2, 165–181.

Laan, MJ van der and Rose, S (2011). *Targeted Learning:Causal Inference for Observational and Experimental Data*. Vol. 1. Springer-Verlag New York.

Laird, NM and Ware, JH (1982). Random-effects models for longitudinal data. *Biometrics* 38.4, 963974.

Pearl, J (2009). *Causality: Models, Reasoning and Inference*. 2nd. New York, NY, USA: Cambridge University Press. ISBN: 052189560X, 9780521895606.

Rø ysland, K (2011). A martingale approach to continuous-time marginal structural models. *Bernoulli* 17.3, 895–915.

Rø ysland, K (2012). Counterfactual analyses with graphical models based on local independence. *The Annals of Statistics* 40.4, 2162–2194.

Robins, J (1986). A new approach to causal inference in mortality studies with a sustained exposure periodapplication to control of the healthy worker survivor effect. *Mathematical Modelling* 7.9, 1393–1512.

Robins, JM and Ritov, Ya (1997). Toward a curse of dimensionality appropriate (CODA) asymptotic theory for semi parametric models. *Journal of the American Statistical Association* 90.429, 106–121.

Robins, JM, Rotnitzky, A, and Zhao, LP (1995). Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90.429, 106–121.

Roderick, J, Little, A, and Rubin, DB (1986). *Statistical analysis with missing data*. J. Wiley.

Scharfstein, DO, Rotnitzky, A, and Robins, JM (1999). Adjusting for nonignorable drop-out using semi-parametric nonresponse models (with comments). *Journal of the American Statistical Association* 94, 1096–1146.

Scharfstein, DO, McDermott, A, Olson, W, and Wiegand, F (2014). Global Sensitivity Analysis for Repeated Measures Studies With Informative Dropout: A Fully Parametric Approach. *Statistics in Biopharmaceutical Research* 6.4, 338–348.

Wright, S (1921). Correlation and causation. *Journal of agricultural research* 20.7, 557–585.