Methods of Joint Modeling for Left-Truncated Data

by Lucinda McGivern

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Master of Science

in

Mathematics & Statistics

Guelph, Ontario, Canada

ABSTRACT

METHODS OF JOINT MODELING FOR LEFT-TRUNCATED DATA

Lucinda McGivern
University of Guelph, 2022

Advisor:
Dr. Julie Horrocks

Survival data may be subject to a form of selection bias known as truncation. This thesis addresses left-truncated data, which arises when a sample is selected to include only those individuals whose truncation time precedes their event time. Statistical analyses in this thesis used joint modeling techniques, which consist of a longitudinal and survival submodel. Comparative studies of joint model methods that ignored and accounted for truncation were carried out on two real datasets and three simulations. Simulation results demonstrated that the analyses which accounted for truncation produced less biased regression parameter estimates than those that ignored truncation.

Acknowledgements

I would like to thank my advisor Dr. Julie Horrocks, advisory committee member Dr. Gerarda Darlington, examiner Jeremy Balka, examination committee chair Ayesha Ali, and graduate program assistant Susan McCormick.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The goal of this thesis is to investigate methods that adjust joint models for truncation. Truncation is a form of data selection that can introduce bias into an analysis.

Joint models use a smoothed time-varying longitudinal covariate as the time-varying predictor in the survival submodel. They do not assume a constant value of the time-varying covariate between measurement times, and are able to account for time-varying covariates measured with error.

In survival analysis, a dataset is truncated when a sample is selected such that only those individuals whose event time of interest lies within a certain observational window are included. In particular, a sample is "left-truncated" when the individuals who have already experienced the event of interest before their truncation time are excluded from the relevant analysis.

Using the R (R Core Team 2020) package JMbayes (Rizopoulos 2016), we will explore methods of joint modeling that can accommodate truncated data. We will begin by applying a joint model analysis to data originating from an ongoing study of the offspring of bipolar individuals (Duffy et al., 2014), to characterize the relationship between Hamilton Anxiety scores (Hamilton 1960) repeatedly measured on the same individual over time and the time to onset of depression and/or bipolar disorder. We will also carry out a joint model analysis on an AIDS dataset accessed from the R (R Core Team 2020) package JM (Rizopoulos 2010), to examine the relationship between longitudinally measured CD4-lymphocytes cell counts in HIV patients and the time to death. We will further carry out a simulation study for this thesis, using the JMbayes package (Rizopoulos 2016) to carry out a joint model analysis of left-truncated data. This thesis will culminate with some broad conclusions about the results of our analysis, and suggestions for future topics of study.

1.1 Truncation

A characteristic of certain survival (time-to-event) datasets is the presence of truncation. Truncation is a form of sampling bias that is introduced into a dataset when particular individuals are omitted from the dataset. Individuals omitted from the dataset are said to be truncated and have no recorded data. Specifically, survival data is left-truncated in the case that we omit from our analysis those individuals whose event time occur before some truncation time (Klein & Moeschberger 2006). Truncation time is sometimes referred to as entry time, or *delayed* entry time, though it can be understood simply as the beginning of each individual's "observational window." Klein and Moeschberger (2006) detail a study in which psychiatric patients were admitted into hospital at a random age, and followed until the study's end or death. Patients who died before admission to hospital are truncated. Special statistical methods are needed to account for truncation in the analysis of such data, as omitting these individuals is a potential source of bias.

Chapter 2

The Linear Mixed Effects Model

In order to accommodate the longitudinal characteristics of the datasets under study, we will make use of the linear mixed effects model. The linear mixed effects model can be expressed as

$$\begin{aligned} Y_i =& X_i\beta + Z_i b_i + \epsilon_i \\ b_i &\sim \mathcal{N}(0, D) \\ \epsilon_i &\sim N(0, R_i) \end{aligned} \tag{1}$$

where Y_i is the $n_i \times 1$ response of individual i, $i = 1, \ldots, n$, X_i is an $n_i \times p$ matrix of covariates for individual i, with associated $p \times 1$ vector of fixed-effect regression parameters given by β. The $q \times 1$ vector of random effects, b_i, is linked to the response Y_i by the $n_i \times q$, $q \leq p$ design matrix Z_i. The random effects, b_i are assumed to be independent of each other and multivariate normally distributed with mean zero and covariance matrix D. The $n_i \times 1$ vector of measurement errors, ϵ_i, are also assumed to be independent of one another and multivariate normally distributed with mean zero. The measurement errors ϵ_i are further assumed to be independent of b_i. The covariance matrix of the measurement errors, $R_i = \mathrm{Cov}(\epsilon_i)$, is typically assumed to be the diagonal matrix $\sigma^2 I_{n_i}$, so that the measurement errors are independently distributed with a common variance σ^2 (Fitzmaurice, Laird and Ware 2004).

The conditional mean of the response Y_i, given b_i, is

$$E(Y_i \mid b_i) = X_i\beta + Z_i b_i \tag{2}$$

and the conditional covariance of Y_i is

$$\mathrm{Cov}(Y_i \mid b_i) = \mathrm{Cov}(\epsilon_i) = R_i. \tag{3}$$

Unconditionally, the mean of Y_i is,

$$
\begin{aligned}
E(Y_i) &= E(X_i\beta + Z_i b_i) \\
&= X_i\beta + Z_i E(b_i) \\
&= X_i\beta
\end{aligned}
\tag{4}
$$

with covariance

$$
\begin{aligned}
\mathrm{Cov}(Y_i) &= \mathrm{Cov}(Z_i b_i) + \mathrm{Cov}(\epsilon_i) \\
&= Z_i \,\mathrm{Cov}(b_i)\, Z_i' + \mathrm{Cov}(\epsilon_i) \\
&= Z_i D Z_i' + R_i.
\end{aligned}
\tag{5}
$$

Chapter 3

Survival Analysis

Survival analysis comprises the statistical methods used to model and analyze situations in which the time to the occurrence of some event is of interest (Lawless 2011). Often, the event of interest is the death of the individual under study, where the "lifetime" of the individual is the amount of time measured from some specific starting point until death (Lawless 2011). When the event of interest is not death, "lifetime" is replaced by "event time", which we will use throughout the remainder of this thesis.

A common feature of survival data is the presence of "censoring", which occurs when it is impossible to determine precisely when the event of interest has occurred (Lawless 2011). Right-censoring may occur in a prospective study when an individual drops out of the study before the event of interest occurs, or may simply occur because the study ends before all individuals in the dataset have encountered the event of interest. In this thesis, right-censoring occurs in all datasets studied.

Time-dependent (longitudinal) covariates associated with survival data vary over time, although the exact value of the covariate between measurement times is often not observed (Lawless 2011). In order to produce data for a longitudinal covariate between these observed measurement times,

survival models must make some assumptions about the value of a longitudinal covariate. A common assumption, called "Last Value Carried Forward" (LVCF), is that the value of the longitudinal covariate remains constant in the interval between measurement times and changes only when an observed measurement is taken. The LVCF results in a time-dependent covariate of the survival model that is a step function, which can introduce bias. Joint models avoid the need for this assumption, as will be explained in section 5.

An important function in survival analysis is the hazard function, which specifies the instantaneous rate of the event at time t, conditioned on the fact that the individual survives up to time t (Lawless 2011). Let the random variable T^* represent the event time under study. The hazard function can be written as

$$h(t) = \lim_{dt \to 0} \frac{\Pr\left(t \leq T^* < t + dt \mid T^* \geq t\right)}{dt}, \quad t > 0. \tag{6}$$

Another important function in survival analysis is the survival function, which describes the probability of an individual surviving to time t and can be written as

$$\mathcal{S}(t) = \exp\left\{-\int_0^t h(s)ds\right\}. \tag{7}$$

The cumulative distribution function of T^* is

$$F(t) = \Pr(T^* \leq t) = \int_0^t f(x)dx \tag{8}$$

where $f(t)$ is the probability density function of the random variable T^*. The survival function is

$$S(t) = \Pr(T^* \geq t) = \int_t^\infty f(x)dx \tag{9}$$

so that the relationship between the probability density function and the survival function is

$$f(t) = -\frac{d}{dt}S(t). \tag{10}$$

3.1 Truncation and Right-Censoring

Survival data is left-truncated when we only include in our analysis those individuals whose event time T^* occurs after some truncation time Y_L (Klein & Moeschberger 2006). In other words, we only observe T^* given that $T^* > Y_L$. Those individuals for whom $T^* \leq Y_L$ are omitted from the dataset, and thus left-truncation is introduced.

Right censoring arises in survival data when an individual's event time is only known to exceed some observed value (Lawless 2011). Right censoring occurs when individuals drop out of a study before its termination, or when the individuals do not experience the event of interest before the end of the observation period. Let T_i^* and C_{ri} denote the true event time and right censoring time for the i^{th} subject, respectively. Let δ_i be an event indicator variable defined by $\delta_i = I\left(T_i^* \leq C_{ri}\right)$, where I is the indicator function which takes on the value of 1 when $T_i^* \leq C_{ri}$ and 0 otherwise. Let the corresponding event time for individual i be defined as $T_i = \min\left(T_i^*, C_{ri}\right)$.

For left-truncated data with truncation independent of the event time, the likelihood function is

$$L = \prod_{i \epsilon D} \frac{f(t_i)}{S(Y_{Li})} \prod_{i \epsilon R} \frac{S(C_{ri})}{S(Y_{Li})}. \tag{11}$$

(Klein & Moeschberger 2006), where R is the set of individuals whose event times were right-censored, D is the set of individuals whose exact event times were observed and, for individual $i = 1, \ldots, n$, $f(t_i)$ is the probability density function at the observed event time t_i, $S(C_{ri})$ is the survival function at the right censoring time C_{ri}, and $S(Y_{Li})$ is the survival function evaluated at the truncation time Y_{Li}. In the event that left-truncation is not present in the dataset, the probability density function and survival function are no longer conditioned on the probability that individual i survives until truncation time Y_{Li}, so that the denominator terms are simply equal to one.

A typical survival analysis in R first creates a survival object with the command $Surv(time, status)$, in which $time$ refers to the event or censoring time, and $status$ refers to an event indicator variable (Therneau 2020). This convention implicitly assumes that the start time of each subject is zero, and fails to account for any truncation that may be present in the dataset. The survival object

is then used as the response in a survival model. In contrast, the counting process syntax (Therneau & Grambsch 2000) creates a survival object using the command $Surv(start, stop, status)$ which allows for the specification of a unique non-zero start time for each individual. The $start$ time corresponds to the truncation time Y_{Li} in equation (11), while the $stop$ time corresponds to t_i in equation (11), which is the event or censoring time for individual i. This survival object is then used as the response in a survival model, which uses the likelihood in equation (11), thus adjusting for truncation.

In the example introduced in section 1.3 (Klein & Moeschberger 2006), psychiatric patients are followed from the time of their admission into hospital until death or study's end, so that the truncation time is the patient's age at admission into the psychiatric hospital. In practice, we might introduce truncation into a dataset if we discard those individuals with no measurements of the longitudinal variable of interest prior to the observation of some survival outcome. In the context of the bipolar study dataset, there are several cases in which the offspring's time of diagnosis occurs before any longitudinal covariate values are measured. Truncation, and potentially bias, are introduced into the dataset when such individuals are discarded.

Chapter 4

Joint Models

4.1 Joint Models for Longitudinal and Survival Data

Often in medical studies, data on a longitudinal variable and a survival response are collected on the same individual. Joint models consist of a longitudinal submodel and a survival submodel. Joint models do not assume LVCF, as they do not assume a constant value of the longitudinal covariate between observation times. As such, these models can account for the impacts of measurement error in the time-varying covariate as it affects the hazard of survival.

4.2 The Linear Mixed Effects Submodel

The longitudinal response for individual $i, i = 1, \ldots, n$ at an arbitrary time t can be written as

$$\begin{aligned} Y_i(t) &= m_i(t) + \epsilon_i(t) \\ \epsilon_i(t) &\sim N\left(0, \sigma^2\right) \end{aligned} \tag{12}$$

where $m_i(t)$ is the true, unobserved response for individual i at time t. The time-dependent error terms, $\epsilon_i(t)$, are assumed to be mutually independent (Rizopoulos 2012). The unobserved response for individual i at time t is given by

$$\begin{aligned} m_i(t) =& x_i'(t)\beta + z_i'(t)b_i \\ b_i &\sim \mathcal{N}(0, D) \end{aligned} \tag{13}$$

where $z_i'(t)$ represents the vector of covariates associated with b_i, the vector of random effects, and $x_i'(t)$ is the vector of covariates associated with β, the vector of fixed effect parameters.

4.3 The Survival Submodel

The survival submodel makes use of the unobserved true longitudinal response, $m_i(t)$, for individual i, $i = 1, \ldots, n$, at time t through the specification

$$h_i(t) = h_0(t) \exp\{w_i'\gamma + \alpha m_i(t)\}, t > 0 \tag{14}$$

where $h_i(t)$ is the hazard for the i^{th} individual at time t, $h_0(t)$ is the baseline hazard function, and α is the parameter that measures the association between $h_i(t)$ and the unobserved true longitudinal response $m_i(t)$ (Rizopoulos 2016). The $1 \times r$ vector of time-independent covariates for individual i, $w_i = (w_{i1}, \ldots, w_{ir})'$, is associated with the $r \times 1$ vector of regression parameters, $\gamma = (\gamma_1, \ldots, \gamma_r)$. The hazard ratio associated with a one unit increase in $m_i(t)$ at fixed time t is given by $\exp(\alpha)$.

Chapter 5

Bayesian Analysis

Bayesian inference makes use of Bayes' theorem to estimate the distribution of a parameter θ. In frequentist inference, parameters are true, fixed unknown quantities that can be described by a single point estimate. For a population parameter θ, we can find standard errors on this point estimate to produce a confidence interval. Frequentist inference is typically carried out by maximum likelihood or least squares estimation. In contrast, Bayesian inference assumes that the parameter θ is itself a random variable.

Bayesian estimates of the distribution of θ are updated from some prior distribution based on available data (Gelman, Carlin, Stern & Rubin 2004). For a continuous parameter θ, Bayes' theorem is written as

$$f(\theta \mid D) = \frac{f(D \mid \theta) f(\theta)}{\int f(D \mid \theta) f(\theta) d\theta}$$

where $\mathbf{D}$ is the data, $f(D \mid \theta)$ is the likelihood of the data, $f(\theta)$ is prior distribution of θ, and $f(\theta \mid D)$ is posterior distribution of θ given D (Gelman et al. 2004). The issue of choosing a prior distribution of θ is often non-trivial. A non-informative prior, which is relatively flat, is used in the case that we have no prior opinion about the shape of the distribution of θ. A $(1-\alpha)100\%$ credible interval for a Bayesian estimator is constructed using the middle $(1-\alpha)100\%$ of the posterior distribution (Edwards, Lindman & Savage 1963).

5.1 Markov Chain Monte Carlo Methods

Computational issues present challenges to carrying out Bayesian inference in practice. In complicated models, the integral in the denominator of the posterior distribution may be intractable. This can be addressed through the use of Markov Chain Monte Carlo (MCMC) methods (Clayton & Bernardinelli 1996). A Markov Chain is a sequence of random variables $\{x_1, x_2, \ldots\}$, such that

the density function of the random variable at the $n+1^{th}$ iteration of the sequence, $f(x_{n+1})$, has the property $f(x_{n+1} \mid x_n, x_{n-1}, \ldots, x_2, x_1) = f(x_{n+1} \mid x_n)$ (Gilks, Richardson & Spiegelhalter 1995). In other words, the future value of the process depends on the present, but not on the past. Under particular conditions, the Markov Chain reaches an equilibrium distribution $\pi(.)$ (Gilks et al. 1995). In an MCMC method, we use a Markov Chain in which $\pi(.)$ is the posterior distribution to generate a random sample, rather than calculating the posterior distribution. In particular, we generate a sequence of parameter values $\theta_1, \ldots \theta_j, \ldots$, so that, for large enough j, the chain has converged to an equilibrium distribution. To ensure convergence, an MCMC method may use a burn-in period, where the first s samples of the chain are discarded (Gilks et al. 1995). This process gives the Markov Chain time to converge to its equilibrium distribution, in the event that it had a starting point distant from its equilibrium. Similarly, an MCMC method should make use of enough iterations to reach convergence. MCMC methods may also employ thinning, which is the process of discarding all but each k^{th} point in a sample, $k = 2, 3, ...$, to reduce iterations between simulations. Convergence may be assessed through diagnostic tools such as trace plots.

5.2 Software

To our knowledge, the JMbayes (Rizopoulos 2016), JMbayes2 (Rizopoulos, Papageorgiou & Miranda Afonso 2022) and rstanarm (Goodrich, Gabry, Ali & Brilleman 2022) packages are the only packages in R (R Core Team 2020) capable of adjusting joint models for truncation. Note that none of these packages are able to handle clustering within this truncated data. As such, any clustering that was present in our real datasets was ignored in the corresponding analyses. Considerations regarding clustering in truncated datasets are addressed in the conclusion.

In this thesis, we fit all joint models using the jointModelBayes() function in the JMbayes package (Rizopoulos 2016). The JMbayes package requires as input a linear mixed effects model object and a survival model object. All linear mixed effects models were fit using the nlme package (Pinheiro, Bates, DebRoy, Sarkar & R Core Team 2020) in R, and all survival models using the survival package (Therneau 2020). The baseline hazard for the survival submodel in JMbayes is

assumed to be a Weibull baseline hazard, and was approximated using penalized B-splines with 15 knots based on the percentiles of the distribution (the default settings in the JMbayes model-fitting function). We used normal priors for the regression parameters of the longitudinal and survival models, and gamma priors for the precision parameters of the linear mixed effects model. All models fit using JMbayes had a burn-in period of 3000 iterations period, after which the algorithm ran for an additional 20000 iterations and was thinned to produce a final chain length of 2000 iterations.

Chapter 6

SharcNet

Fitting joint models under a Bayesian framework is computationally expensive, and subsequently requires substantial computer time to fit even a single joint model. As the complexity and number of models increase, as in the case of a simulation study, the prospect of fitting such models using a general purpose computer becomes untenable. To contend with the computational demands posed by the Bayesian joint model, we carried out the simulation using the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca), which is a shared network of computer clusters hosted from a consortium of 14 universities, 3 colleges, and a research institute, all within Ontario. These clusters share computing resources and are able to meet the enormous memory and processing demands that may arise in large scale simulations.

To make use of the available CPUs and memory on the SHARCNET system, we wrote our code in "parallel". Parallel computing delegates multiple tasks to different processors to be completed simultaneously. In contrast, running an R script in serial would entail that all recruited CPUs work to complete a single task at a time, until the entire script has been executed in sequence. Within the data generation and model fitting process, we made use the lapply() function in R. This parallel function carries out matrix computations on an entire list of objects simultaneously (as opposed to the serial for-loop method, which would allocate all available resources to carrying out a single iteration of the loop at a time). The lapply() function alone, however, will not sufficiently reduce

computation time, as it still delegates that all available CPUs carry out each task. We made use of the parLapply function from the doParallel package (Corporation & Weston 2022) in R, which allowed us to carry out the lapply function over a list of multiple CPU cores.

All of the simulation study was done on the Graham cluster of the Compute Canada umbrella. We generated 5000 datasets and fit two joint models per iteration - one model that accounted for truncation and one that did not. For each iteration, the associated longitudinal and survival model had to first be fit, so these submodels had to be fit serially. The joint models were fit in parallel across 32 CPU cores using 10Gb of memory per CPU. The simulation settings that used constant truncation times were given a maximum of 3 hours to run, while the simulation setting that used a random, indpendent truncation time required a maximum of 12 hours to run. We saved only the summary statistics from each iteration of the simulation to minimize the amount of memory and resources required. These statistics were the average of each regression coefficient estimate taken over all iterations and the average of the posterior standard deviations taken over all iterations.

Chapter 7

Data Analyses

In order to explore the effects of truncation bias, we fit a joint model that accounted for truncation and a joint model that ignored truncation. The joint model that accounted for truncation made use of the $(start, stop, status)$ counting process syntax introduced in section 3.1, allowing us to specify differing, non-zero start times among the individuals used in our analysis. The joint model that ignored truncation made use of the $(time, status)$ syntax, which implicitly assumed a constant start time of zero among all individuals.

7.1 Bipolar Data Analysis

In this thesis, we will carry out a joint model analysis using data described by Duffy et al. (2014). This dataset comprises 305 individuals who were deemed to be at-risk for bipolar disorder

type as they had exactly one parent diagnosed with the illness. First degree relatives of bipolar individuals have an eight- to tenfold lifetime risk of developing bipolar disorder as compared to the general population (Duffy, Grof, Robertson & Alda 2000). Due to the heritability of the illness, longitudinal studies of the offspring of bipolar individuals may be useful in predicting disease onset. Recruitment of the parental group and complete data collection processes are detailed in Duffy et al. (2014).

Anxiety levels in the offspring under study were measured through the Hamilton Anxiety Rating Scale, a psychiatrist-administered survey used to assess the severity of anxiety symptoms (Hamilton 1960). The survey consists of 14 items, with each symptom scored on a scale of 0 (symptom not present) to 4 (severe). The Hamilton Anxiety (HAM-A) score was measured on each individual in the dataset at approximately yearly intervals. Other predictors such as sex, parental onset age, the Hollingshead Four-Factor Index of Socioeconomic Status (SES Score), and parental response to lithium were also recorded.

The original dataset consists of 124 male and 181 female offspring. Of these 305 individuals, 32 did not have any HAM-A scores recorded and were omitted from our analysis. There were an additional 106 individuals who had HAM-A measurements only taken after their event time. These individuals were omitted from our analysis, which introduced truncation into the dataset. Ultimately, the sample we analyzed consisted of 167 individuals, all of whom had HAM-A scores measured before their event or censoring times.

Within the 167 person sample used for this analysis, 148 did not experience the event over the duration of the study and were censored. There was a mean of 3.174 HAM-A measurements per individual. Frequencies of the number of visits per individual are presented in Table 1. Visits occurred approximately yearly, and descriptive statistics for the visit times are presented in Table 2. Descriptive statistics for the time-fixed predictors of the dataset are presented in Table 3. A plot of HAM-A vs time for each individual is presented in Figure 1, with measurements for each individual connected by a line.

In order to conform to the normality assumption for the residuals of the longitudinal submodel,

the log of the HAM-A scores was taken and used throughout our analysis. Given that HAM-A scores have a lower bound of zero, a value of 1 was added to every score in the dataset before applying the log transformation. A histogram of the log-transformed HAM-A scores is presented in Figure 2.

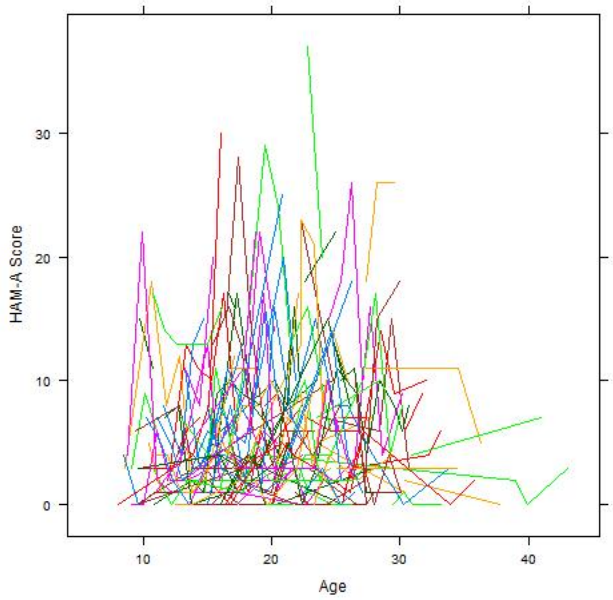

Figure 1. HAM-A scores versus age for the bipolar dataset.

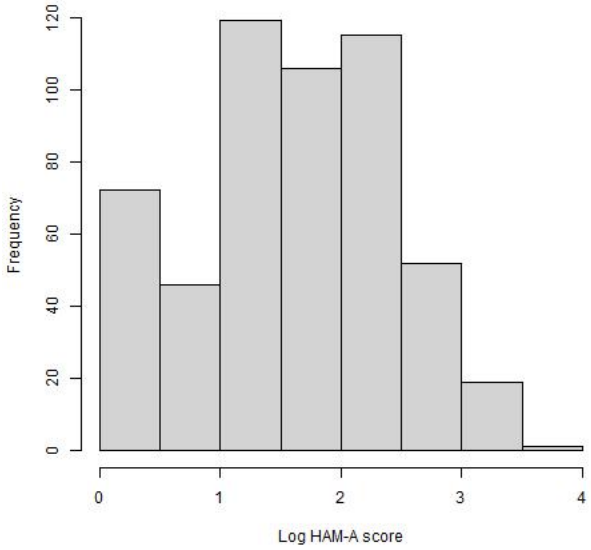

Figure 2. Histogram of log-transformed HAM-A scores for the bipolar dataset.

Table 1. A frequency table of the number of HAM-A measurement visits for individuals in the bipolar dataset.

Number of Visits	**Frequency**
1	54
2	33
3	19
4	20
5	15
6	7
7	10
8	1
9	5
10	1
11	1
12	1

Table 2. Descriptive statistics for the HAM-A measurement visit times of the bipolar dataset.

Variable	**Mean**	**Minimum**	**Maximum**
Age at first visit	18.30	6.56	42.13
Number of years followed	6.407	0.05	17.250
Age at last visit	24.71	7.68	45.81
Visits per individual	3.17	1	12

Table 3. Descriptive statistics for the time-fixed predictors of the bipolar dataset.

Variable	**Count**
Parental Socioeconomic Status	
SES Score of 1, 2, or 3	28
SES Score of 4	57
SES Score of 5	82
Sex	
Female	91
Male	76
Parental Onset Age	
Mean	26.67
Parental Lithium Response	
Positive	65
Negative	102

For the bipolar dataset, the longitudinal submodel fit in each joint model was a linear mixed effects model with a linear effect of time and random intercept and slopes terms:

$$Y_i(t) = m_i(t) + \epsilon_i(t) = \beta_0 + \beta_1 t + b_{i0} + b_{i1} t + \epsilon_i(t) \tag{15}$$

$$b_i \sim \mathcal{N}(0, D)$$

$$\epsilon_i(t) \sim N\left(0, \sigma^2 I_{n_i}\right)$$

where, for individual i at time t, $Y_i(t)$ is the log of the HAM-A score, $m_i(t)$ is the true, unobserved HAM-A score, and $\epsilon_i(t)$ is the measurement error. The term β_1 is the regression parameter associated with observation time t, β_0 is the intercept, and $b_i = (b_{i0}, b_{i1})$ is the vector of random effects for individual i. Each b_i vector is multivariate normal with mean 0 and covariance matrix

$$D = \begin{pmatrix} \sigma^2_{b_0} & \sigma_{b_{10}} \\ \sigma_{b_{10}} & \sigma^2_{b_1} \end{pmatrix}.$$

The survival submodel fit in each joint model was

$$h_i(t) = h_0(t) \exp\{\gamma_1 w_{i1} + \gamma_2 w_{i2} + \gamma_3 w_{i3} + \gamma_4 w_{i4} + \alpha m_i(t)\}, t > 0 \tag{16}$$

where, for individual i, w_{i1} is the individual's sex, which is coded 1 for female and 0 for male. The w_{i2} covariate represents the individual's parental lithium response status, coded 1 if the parental response to lithium was positive and 0 otherwise. The covariate w_{i3} is the indicator of a parental SES score of 1, 2, or 3, and the covariate w_{i4} indicates an SES score of 4. The parental SES level 5 was used as the reference group for the SES predictor in the survival submodel. Each γ_k is the associated regression coefficient. The association parameter, which quantifies the relationship between $m_i(t)$ and the hazard, is written as α. The estimated regression coefficients, as well as the 95% credible intervals, posterior distribution standard deviations, and p-values (as described in section 8.3) are presented in Table 4. The relevant R code for this analysis can be found in Appendix A.2.

There were notable differences between the results of the analysis that accounted for truncation and the analysis that ignored truncation. The estimate of β_0 was 6.85% larger in the case that truncation was accounted for than in the case when truncation was ignored, while the estimate of β_1 was 33.01% larger. This trend did not extend to the regression parameter estimates of the survival submodels. The regression parameter estimate of γ_2 was 80.75% smaller in magnitude in the case that truncation was accounted for, as compared to the case when it was ignored. The estimate of γ_4 was similarly 80.52% larger in magnitude in the case that truncation was accounted for. There was also a substantial difference in the p-value, as accounting for truncation produced an association estimate with a p-value one-tenth the size as compared to the case in which truncation was ignored: the p-value of the association parameter was 0.002 when accounting for truncation, and only 0.02 when truncation was ignored.

7.2 AIDS Analysis

Here we will carry out a joint model analysis of the AIDS dataset, accessed from the R (R Core Team 2020) package JM (Rizopoulos 2010). This dataset was collected from December of 1990 until September of 1992, under a study which monitored 467 patients with human immunodeficiency virus (HIV) until they died or developed acquired immunodeficiency syndrome (AIDS) (Abrams,

Table 4. Joint model parameter estimates, standard deviations, 95% credible intervals, and associated p-values for the joint models fit in the bipolar dataset. The log of the HAM-A scores is the response in the longitudinal submodel, and the time to diagnosis is the response in the survival submodel.

Accounting for Truncation					
	Coefficient	St. Dev	2.5%	97.5%	P-Value
Longitudinal Submodel					
β_0 (Intercept)	0.9326	0.1877	0.5370	1.2386	$<$0.001
β_1 (t)	0.0411	0.3027	-0.5466	0.6097	0.864
σ_{b_0}	1.672				
σ_{b_1}	3.815				
$\sigma_{b_{01}}$	0.0807				
σ	0.612				
Survival Submodel					
γ_1 (Sex)	0.2443	0.5453	-0.7314	1.3781	0.676
γ_2 (Lithium Response)	-0.0951	0.5816	-1.2864	1.0232	0.873
γ_3 (Parent SES 123)	0.4846	0.6745	-0.9163	1.7481	0.460
γ_4 (Parent SES 4)	-0.2928	0.6241	-1.5875	0.8999	0.626
α (Association)	0.8345	0.3333	0.2382	1.5046	0.002
Ignoring Truncation					
	Coefficient	St. Dev	2.5%	97.5%	P-Value
Longitudinal Submodel					
β_0 (Intercept)	0.8728	0.2047	0.4985	1.2933	$<$0.001
β_1 (t)	0.0309	0.2995	-0.5700	0.6097	0.939
σ_{b_0}	2.055				
σ_{b_1}	3.815				
$\sigma_{b_{01}}$	0.055				
σ	0.595				
Survival Submodel					
γ_1 (Sex)	0.2541	0.5260	-0.7601	1.3371	0.630
γ_2 (Lithium Response)	-0.4398	0.5681	-1.5571	0.6267	0.443
γ_3 (Parent SES 123)	0.5089	0.6972	-0.9382	1.7819	0.410
γ_4 (Parent SES 4)	-0.1622	0.6297	-1.4991	1.0348	0.799
α (Association)	0.8597	0.3225	0.2317	1.4803	0.020

Goldman, Launer, Korvick, Neaton, Crane, Grodesky, Wakefield, Muth & Kornegay 1994). Study subjects received one of two anti-HIV therapies, Didanosine (ddl) or Zalcitabine (ddC), in a randomized clinical trial consisting of individuals who had failed to respond to, or were intolerant to, zidovudine (AZT) therapy. CD4 lymphocyte count was measured longitudinally on each subject. CD4 lymphocyte count typically increases as HIV symptoms are effectively managed and disease progression is subsequently halted, as CD4 cells' function is to fight infection (Goldman, Carlin, Crane, Launer, Korvick, Deyton & Abrams 1996). In this thesis, we will explore the association between the longitudinal variable of CD4 lymphocyte count and the time to the subject's death.

The original AIDS dataset consists of 467 patients, approximately 71 percent of whom survived until study's end and were censored. The CD4 cell counts for each patient, by gender, are plotted over time in Figure 3. There were an average of 3 visits per patient, with a minimum of 1 visit and a maximum of 5. The frequencies of visits per patient are presented in Table 5. There was a mean of 3.74 years between visits. A histogram of the CD4 cell counts, as presented in Figure 4, demonstrates right-skewness in the dataset. This was addressed by applying the square root transformation to the CD4 cell counts, which was used as the response in the linear mixed effects model throughout this analysis.

There were no individuals in the dataset who experienced the event before their baseline CD4 measurements. Thus there is no truncation present in the original AIDS dataset. A left-truncated dataset was produced for analysis by omitting baseline (time 0) CD4 measurements from our analysis. Then only those individuals with at least one CD4 measurement prior to their event time were included in the analysis, which truncated the dataset by 61 individuals, or approximately 13%. The truncated sample consisted of 406 individuals, approximately 76% of whom survived until study's end and were censored. There was a mean of 4.54 years between visit times. Note that The artificial truncation that we introduced was based on Rizopoulos' (2016) treatment of the pbc dataset (Rizopoulos 2010), in which all time zero longitudinal measurements were omitted from the analysis. Summary statistics for the predictors of the full and artificially left-truncated datasets are presented in Table 6.

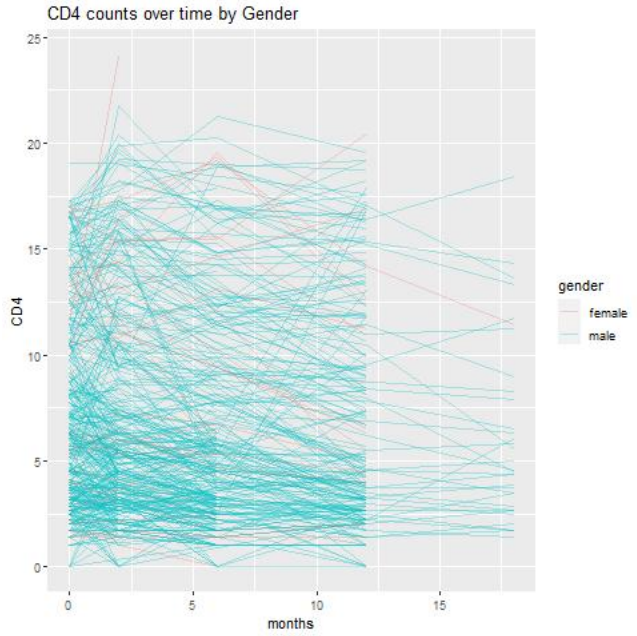

Figure 3. CD4 cell counts over time for each patient in the AIDS dataset, by gender.

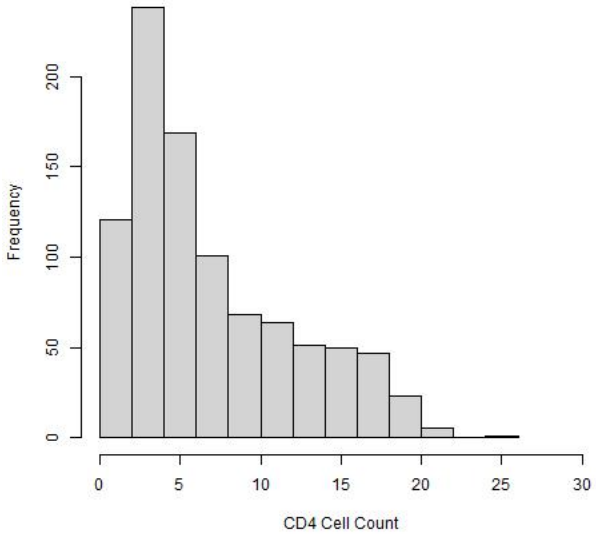

Figure 4. Histogram of CD4 cell counts for the AIDS dataset.

Table 5. A frequency table of the number of visits for individuals in the full AIDS dataset.

Number of Visits	Frequency
1	61
2	91
3	122
4	169
5	24

Table 6. Descriptive statistics for predictors of the full and truncated AIDS datasets.

Variable	Full Dataset		Left-Truncated Dataset	
	Frequency	%	Frequency	%
AZT Status				
Intolerance	292	62.53	261	64.29
Failure	175	37.47	145	35.71
Sex				
Female	45	9.64	33	8.13
Male	422	90.36	373	91.87
Previous Infection Status				
AIDS	307	65.74	259	63.79
No AIDS	160	34.26	147	36.21
Drug Status				
ddC	237	50.75	208	51.23
ddI	230	49.25	198	48.79

In this section, we fit a joint model that accounted for truncation as well as a joint model which ignored the effects of truncation to the artificially truncated AIDS dataset. The longitudinal submodel fit for each joint model was a linear mixed effect model with a linear effect of time and random intercept and slopes terms:

$$Y_i(t) = m_i(t) + \epsilon_i(t) = \beta_0 + \beta_1 t + b_{i0} + b_{i1} t + \epsilon_i(t) \tag{17}$$

$$b_i \sim \mathcal{N}(0, D)$$

$$\epsilon_i(t) \sim N\left(0, \sigma^2 I_{n_i}\right)$$

where, for individual $i, i = 1, \ldots n$ at time t, $Y_i(t)$ is the response of the square root of the CD4 cell count, $m_i(t)$ is the true, unobserved response for individual i at time t, β_0 is the intercept, β_1 is the regression parameter associated with the observation time t, and $b_i = (b_{i0}, b_{i1})$ is the vector of random effects where each b_i vector is multivariate normal with mean 0 and covariance matrix

$$D = \begin{pmatrix} \sigma^2_{b_0} & \sigma_{b_{10}} \\ \sigma_{b_{10}} & \sigma^2_{b_1} \end{pmatrix}.$$

The survival submodel fit in each joint model was

$$h_i(t) = h_0(t) \exp\{\gamma_1 w_{i1} + \gamma_2 w_{i2} + \gamma_3 w_{i3} + \gamma_4 w_{i4} + \alpha m_i(t)\}, t > 0 \tag{18}$$

where, for individual $i = 1 \ldots n$, w_{i1} indicates drug status, coded as 1 if the individual received Didanosine and 0 if they received Zalcitabine. The covariate w_{i2} indicates the individual's gender, and is coded as 1 if the individual was male and 0 if they were female. The covariate w_{i3} indicates the individual's AZT status, which was coded as 1 if they had failed to respond to zidovudine therapy, and 0 if they were intolerant to zidovudine therapy. The covariate w_{i4} indicates the individual's previous infection status, which was coded as 1 if they had previously been diagnosed with AIDS and 0 otherwise. Each γ_k is the regression parameter associated with the w_{ik} covariate. The association parameter, which quantifies the effect of $m_i(t)$ on the hazard, is denoted as α.

We additionally fit a joint model to the full (untruncated) dataset, using these same longitudinal and survival submodels. Output for the three resulting joint models (accounting for truncation, ignoring truncation, and full dataset) are presented in Table 7. The relevant code for the AIDS dataset analysis can be found in Appendix A.1.

Results for the models fit to the truncated dataset (accounting for truncation, ignoring truncation) will first be discussed. The linear mixed effects submodels were very similar in both approaches. However, some differences in the survival submodels are apparent. The model that accounted for truncation produced an estimate for the male gender indicator that was approximately 34% larger in the truncated dataset than in the model that ignored truncation. The association parameter estimate and p-value in the model that accounted for truncation were virtually the same as in the model that ignored truncation. The p-value of the previous infection status estimate accounting for truncation was less than 50% of the magnitude of the corresponding p-value in the analysis that ignored truncation. This difference can be attributed to the fact that the γ_4 estimate was larger in the analysis that accounted for truncation.

We will also present a comparison of the results between the model that ignored truncation, and the model that was fit to the full dataset. The longitudinal regression parameter estimates were larger in analyses that ignored truncation than in the full dataset analysis. The male gender indicator estimate was 159.2% smaller in the full dataset than it was when truncation was ignored. The estimate of β_1 was much more significant in the full dataset analysis, as it had a p-value that was approximately 92.7% smaller than when truncation was ignored. This is a function of the fact that the regression parameter estimate was larger in magnitude in the full dataset as compared to the truncated dataset, while the associated 97.5^{th} quantile of the MCMC chain estimate was smaller. The association parameter, α, was highly significant across both models, but was larger in magnitude when fit to truncated data.

There appears to be a distinct difference in the longitudinal and survival regression parameter estimates between the analysis carried out on the full dataset and the analyses carried out on the truncated dataset. The longitudinal regression parameter estimates were smaller in the case that

Table 7. Joint model parameter estimates, standard deviations, 95% credible intervals, and associated p-values for the joint models fit in the AIDS dataset. The square root of the CD4 cell count is the response in the longitudinal submodel, and the time to death is the response in the survival submodel.

Accounting for Truncation					
	Coefficient	St. Dev	2.5%	97.5%	P-Value
Longitudinal Submodel					
β_0 (Intercept)	2.7122	0.0516	2.6083	2.8111	<0.001
β_1 (t_{ij})	-0.1008	0.1105	-0.3241	0.1085	0.358
σ_{b_0}	0.9171				
σ_{b_1}	2.2269				
$\sigma_{b_{01}}$	-0.0401				
σ	0.3650				
Survival Submodel					
γ_1 (drugddI)	0.3183	0.2216	-0.1156	0.7853	0.149
γ_2 (gendermale)	0.3889	0.5128	-0.5092	1.5402	0.451
γ_3 (AZTfailure)	-0.0280	0.2529	-0.5261	0.4613	0.935
γ_4 (prevOIAIDS)	0.7519	0.3093	0.1717	1.3761	0.009
α (Association)	-0.9582	0.1395	-1.2287	-0.6855	<0.001
Ignoring Truncation					
	Coefficient	St. Dev	2.5%	97.5%	P-Value
Longitudinal Submodel					
β_0 (Intercept)	2.7211	0.0533	2.6163	2.8234	<0.001
β_1 (t_{ij})	-0.1057	0.1149	-0.3290	0.1109	0.37
σ_{b_0}	0.9304				
σ_{b_1}	2.2328				
$\sigma_{b_{01}}$	-0.0423				
σ	0.3647				
Survival Submodel					
γ_1 (drugddI)	0.3210	0.2221	-0.1071	0.7642	0.144
γ_2 (gendermale)	0.2903	0.4500	-0.5362	1.2301	0.530
γ_3 (AZTfailure)	0.0085	0.2637	-0.5166	0.5329	0.989
γ_4 (prevOIAIDS)	0.7199	0.3345	0.1155	1.4002	0.020
α (Association)	-0.9824	0.1356	-1.2619	-0.7358	<0.001
Full Dataset					
	Coefficient	St. Dev	2.5%	97.5%	P-Value
Longitudinal Submodel					
β_0 (Intercept)	2.5377	0.0430	2.4544	2.6246	<0.001
β_1 (t_{ij})	-0.2057	0.0964	-0.3989	-0.0210	0.027
σ_{b_0}	0.8746				
σ_{b_1}	1.9896				
$\sigma_{b_{01}}$	0.0590				
σ	0.3602				
Survival Submodel					
γ_1 (drugddI)	0.2596	0.1961	-0.1221	0.6466	0.198
γ_2 (gendermale)	-0.1718	0.3201	-0.8202	0.4240	0.617
γ_3 (AZTfailure)	0.0016	0.2196	-0.4126	0.4410	0.990
γ_4 (prevOIAIDS)	0.7629	0.2872	0.1972	1.3037	0.008
α (Association)	-0.8780	0.1062	-1.0951	-0.6820	<0.001

the full dataset was analysed, and the estimate of β_1 was significant only in the case that the full dataset was analysed. For the survival submodel, the association parameter was the only regression parameter estimate that was larger in the full dataset analysis than in the truncated analyses. These differences between the full and truncated analyses may be the case that the truncation time for each individual, from which observation began, could be caused by a dependence between truncation time and event time. To explore this possibility, the joint model analysis that accounted for truncation (as presented in Table 7) was again carried out, with the additional term of each individual's truncation time as a covariate in the survival submodel (Liu, Li & Zhang 2018). Under this new analysis, the regression coefficient associated with the starting observation time for each individual was 0.0411, with a p-value of 0.523, which indicates that the truncation time is not associated with the survival outcome in this dataset, after adjusting for drug status, gender, AZT status, and previous infection status.

Chapter 8

Simulation

The purpose of this simulation study is to assess the effects of ignoring truncation in a joint model analysis. This simulation was loosely based on the bipolar dataset, from which we derived the true parameter values of the generated data. We generated longitudinal data for every individual from a random intercepts linear mixed effects model with observation time as the sole predictor. We generated the corresponding event times from a survival submodel with a single binary time-fixed predictor. We discarded those individuals whose event times preceded their first longitudinal observation time. This introduces truncation into the data set. More details follow in section 7.2.

To assess the results of the simulation, the average of the estimates of each parameter θ, as well as the relative percentage bias and the average standard deviation taken over all iterations were presented. For each parameter θ, the average of the regression parameter estimates is

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_i, \tag{19}$$

where $\hat{\theta}_i$ is the estimate from i^{th} iteration, N is the number of simulations, and $\bar{\theta}$ is the mean of the regression parameter estimates taken over all iterations.

The relative percentage bias (Demirtas 2007) is

$$RE(\%) = \frac{\bar{\theta} - \theta}{\theta} \times 100. \tag{20}$$

The average of the posterior standard deviations of the regression parameter estimates taken over the entire simulation are reported.

Finally, the empirical standard deviation of the regression parameter estimate was calculated as

$$SD(\hat{\theta}) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{\theta}_i - \bar{\theta}\right)^2. \tag{21}$$

8.1 Generation of Event Times

Generating event times was challenging as we needed to accommodate a continuous time-dependent covariate when generating event data from a survival model. Leemis (1987) showed that to generate a survival time T from a distribution with cumulative hazard $H(t)$, let

$$T = H^{-1}(-log(U)) \tag{22}$$

where $U \sim Uniform(0, 1)$. Bender (2006) derived a method for generating event times from the exponential model with time-fixed covariates, and Austin (2012) extended these methods to accommodate a continuous time-dependent covariate. Stefan (2019) further extended these methods to applications in joint modeling. Our simulation made use of these results to incorporate both time-fixed and time-dependent covariates in the generation of event times.

The longitudinal data in our simulation study was generated from a special case of the linear mixed effects model of Section 5.2, namely

$$\begin{aligned} y_{ij} &= m_{ij} + \epsilon_{ij} \\ &= \beta_0 + \beta_1 t_{ij} + b_i + \epsilon_{ij} \end{aligned} \tag{23}$$

where y_{ij} is the response of individual i, $i = 1, ..., n$ at time t_{ij}, $j = 1, ..., n_i$, β_0 is the intercept, β_1 is the slope associated with time t_{ij}, b_i is the random intercept for the i^{th} individual, and ϵ_{ij} is the random error for individual i at time t_{ij}. The unobserved response of individual i at time t_{ij}, $j = 1, ..., n_i$ is represented by $m_{ij} = \beta_0 + \beta_1 t_{ij} + b_i$. We assume $b_i \sim N(0, \sigma_b^2)$ and $\epsilon_{ij} \sim N(0, \sigma^2)$, where b_i and ϵ_{ij} for all i and j are independent of each other. This special case of the linear mixed effects model is referred to as a random intercepts model.

The survival data used in our simulation was generated for $i = 1, ..., n$ and $j = 1, ..., n_i$ observations per individual at time t from a Cox model (Andersen & Gill 1982) given by the equation

$$h_i(t) = h_0(t) \exp\{w_i'\gamma + \alpha m_i(t)\}, \quad t > 0 \tag{24}$$

where $h_0(t)$ is the baseline hazard function of individual i at time t, w_i is a time-fixed binary covariate and γ is the associated regression parameter. The parameter α measures the association between $h_i(t)$ and the unobserved time-dependent response $m_i(t)$. $h_i(t)$ is the hazard for the i^{th} individual at time t. The cumulative hazard function for the hazard function for individual i (given in equation 24) is

$$H_i(t) = \int_0^t h_0(u) \exp\left\{w_i'\gamma + \alpha m_i(u)\right\} du. \tag{25}$$

Assuming the unobserved response of individual i at time t has the form given in equation (23), and further assuming $h_0(t) = \lambda$, the cumulative hazard function can be written as (Stefan, 2019)

$$\begin{aligned} H_i(t) &= \int_0^t \lambda \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + \beta_1 u + b_i\right)\right\} du \\ &= \lambda \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\} \int_0^t \exp\left\{\alpha\beta_1 u\right\} du \\ &= \frac{\lambda}{\alpha\beta_1} \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\} \left(\exp\left\{\alpha\beta_1 t\right\} - 1\right). \end{aligned} \tag{26}$$

Note that $\beta_1 > 0$ is necessary in order to have a valid cumulative hazard. The inverse of this function is (Stefan 2019)

$$H_i^{-1}(u) = \frac{1}{\alpha\beta_1} \log\left\{1 + \frac{\alpha\beta_1 u}{\lambda \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\}}\right\}. \tag{27}$$

Using the result from equation (22), we can generate the event time as

$$T_i = \frac{1}{\alpha\beta_1} \log\left\{1 + \frac{\alpha\beta_1(-log(U_i))}{\lambda \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\}}\right\} \tag{28}$$

where $U_i \sim U(0,1)$.

8.2 Generating a Simulated Dataset

In this simulation, the true parameter values were loosely based on bipolar dataset described in Section 2.1. A value of 0 or 1 for w_i was assigned by drawing from a $Bernoulli(0.45)$ distribution

Table 8. True values used in all settings of the simulation study, derived from the bipolar dataset.

	Setting 1	Setting 2	Setting 3
Proportion Male	0.45	0.45	0.45
β_0	1.08	1.08	1.08
β_1	**0.023**	**0.05**	0.023
σ_b	0.63	0.63	0.63
σ	0.69	0.69	0.69
α	0.96	0.96	0.96
γ	0.29	0.29	0.29
λ	0.005	0.005	0.005
First Visit Date	18.30	18.30	$lognormal(2.82, 0.39)$
Last Visit Date	24.30	24.30	24.30

(corresponding to the variable sex, 45% of the bipolar dataset was male). We generated from the random intercepts model given in equation (23). We generated random effects $b_i \sim N(0, \sigma_b^2)$ for $i = 1, 2, ..., n$, and random errors $\epsilon_{ij} \sim N(0, \sigma^2)$ for $i = 1, 2, ..., n$ and $j = 1, ..., 4$. We then generated survival times from equation (28).

We carried out this simulation under three different settings of parameters. In the first and second simulation settings, every individual in the dataset had the same number of visits and same visit dates. The first visit date was the mean first visit date of the bipolar dataset (18.30 years), the last visit date was approximately equal to the mean last visit date of the bipolar dataset (24.3 years). The remaining visit dates were evenly allotted over this interval to produce 4 visits per individual, which approximately equals the mean number of observations per person in the bipolar dataset before truncation is introduced. The value of λ was set to 0.005 by trial and error in the first simulation setting, to produce event times with a median value representative of the real dataset. As presented in Table 9, the bipolar dataset event times had a median of 25.24 years, while the generated event times under the first simulation setting had a median of 28.97 years (using $\beta_0 = 1.08$ and $\beta_1 = 0.023$). The true values used in the generation of the data for the first and second settings of the simulation are presented in Table 8. It should be noted that the only distinction between these two settings lies in the true β_1 value, which is 0.023 in the first setting and 0.05 in the second.

Our third simulation setting investigated randomly and independently truncated data. We thus generated a random truncation time for each individual, which was independent of the event time.

To decide what distribution to use for the truncation time, we fit both a lognormal and exponential model to the distribution of first observation times in the original bipolar dataset, and found that the lognormal distribution fit better. Plots of these densities fit to a histogram of the first observation times are presented in Figure 5. The true values used in the generation of the data for this third simulation setting are also presented in Table 8.

Fitting a lognormal distribution

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}}, x > 0$$

to the set of first observation times by a maximum likelihood estimation method using thefitdist() function from the R MASS library (Venables & Ripley 2002) yielded an estimate of the μ parameter of 2.821 and of the σ parameter of 0.389. We then generated a truncation time for each individual in the dataset from this lognormal distribution, and replaced their first observation time in our simulated dataset (18.3 years) with this randomly and independently generated truncation time. The remaining HAM-A observation times were kept the same as in settings 1 and 2 of the simulation (20.3 years, 22.3 years, and 24.3 years), though any observations that occurred before an individual's truncation time were also excluded from our analysis. In the case that an individual's truncation time occurred after all of the HAM-A observation times, they were considered to have no HAM-A observations and were similarly excluded from the analysis.

For all three simulation settings, we fit joint models using the random intercepts linear mixed effects model for the longitudinal submodel and Cox model for the survival submodel. For each setting, we present two analyses. In one analysis, we accounted for truncation by specifying the non-zero truncation time. In the second analysis, we ignored the effects of truncation by using a truncation time of zero in the call to fit the joint model. R code for the simulation study is presented in Appendix A.3.

The data generation steps are repeated below as an algorithm for $N = 5000$ iterations.

- Set $n = 251$, $n_i = 4$, and observation times $t_{i1} = 18.3$, $t_{i2} = 20.3$, $t_{i3} = 22.3$, $t_{i4} = 24.3$ for

$i = 1, ..., n$.

- Set values for parameters β_0, β_1, σ_b, σ, α, γ, λ.

- For i in $1 : n$, generate mutually independent $U_i \sim Uniform(0, 1)$, binary covariate w_i with $Bernoulli(0.45)$, $b_i \sim N(0, \sigma_b)$, and event time

$$T_i = \frac{1}{\alpha\beta_1} \log \left\{ 1 + \frac{\alpha\beta_1(-log(U_i))}{\lambda \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\}} \right\}.$$

  - For j in $1 : n_i$, generate $\epsilon_{ij} \sim N(0, \sigma)$ and longitudinal response

$$y_{ij} = \beta_0 + \beta_1 t_{ij} + b_i + \epsilon_{ij}.$$

8.3 Details of Bayesian Analysis within Simulation

The MCMC algorithm was run for 20000 iterations after a burn-in period of 3000 iterations were discarded. The chain is then thinned such that 2000 samples are ultimately kept. For the association parameter α, the longitudinal submodel regression coefficients, and survival submodel regression coefficients, independent univariate diffuse normal priors are used. The precision parameter from the linear mixed effect $\tau = \frac{1}{\sigma^2}$ has a gamma prior (Rizopoulos 2016).

Characteristics of the MCMC chain were also examined. The JMbayes model summary object returns a regression parameter estimate 'Value', which is the mean of each chain, as well as the sample standard deviation of each chain. This sample standard deviation, called 'Std.Dev' in the model output, uses the length of the final chain l. The 2.5% and 97.5% quantiles of the final MCMC chain are also presented to form a 95% credible interval for each regression parameter estimate. For each regression parameter estimate from the longitudinal or survival submodel θ, the tail probabilities, or p-values, are calculated as

$$2 \cdot min\{P(\theta > 0), P(\theta < 0)\}$$

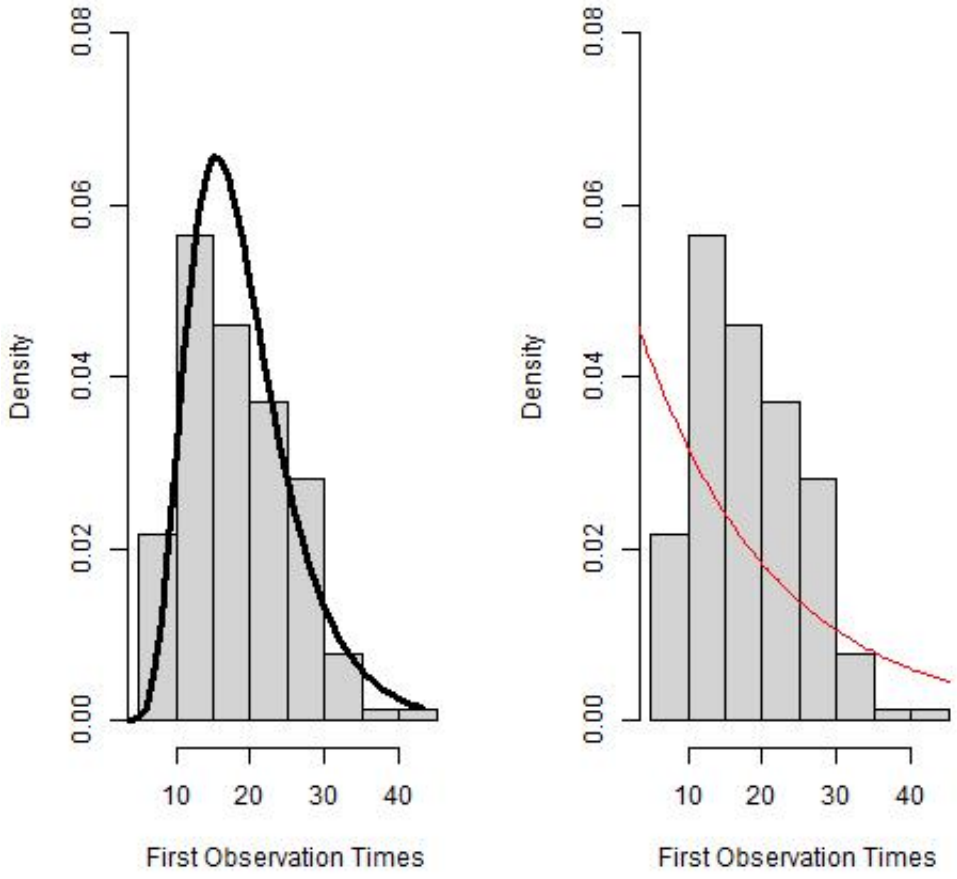

Figure 5. A histogram of the bipolar dataset's first observation times, with a $lognormal(2.821, 0.389)$ curve superimposed (left) and with an $exp(0.055)$ curve superimposed (right).

(Rizopoulos 2016).

We assessed the convergence of the MCMC chains by examining the relevant trace plots, which plots the Markov Chain parameter value against MCMC iteration number, for randomly selected iterations of the simulation. The trace plots for the regression parameter estimates from the longitudinal submodel and survival submodels from one iteration of the simulation are presented in Figure 6. In an MCMC chain that has converged, the chain will appear to be a thick line with some fluctuations above and below the line, indicating that the parameter values have centered around some value. An MCMC chain that has not converged will remain in the same state for an extended period, producing a flat area in the trace plot, or it will take many consecutive steps in the same direction (Gabry, Simpson, Vehtari, Betancourt & Gelman 2019). The trace plots below were produced from one iteration of the simulation. They centre around a value with random movements above and below that value, suggesting that convergence has been achieved. The trace plot for the association parameter demonstrates slower mixing, indicating a larger degree of autocorrelation between samples.

8.4 Simulation Results: Constant Truncation Time, $\beta_1 = 0.023$

We carried out the first simulation over 5000 iterations, using the setting values presented in Table 8. In order to check whether the generated event times had the desired distribution, we plotted theoretical versus estimated cumulative hazard function $H_i(t)$ for randomly selected iterations of the simulation. We calculated the Nelson-Aalen estimates of the cumulative hazard function for the generated event times using the *mice* package (Van Buuren & Groothuis-Oudshoorn 2011), and plotted them as points against the generated event times. We overlayed this plot with the Nelson-Aalen estimate of the theoretical cumulative hazard function for the event times

$$H_i(t) = \frac{\lambda}{\alpha\beta_1} \exp\left\{w_i'\gamma + \alpha\left(\beta_0 + b_i\right)\right\}\left(\exp\left\{\alpha\beta_1 t\right\} - 1\right)$$

generated with $b_i = 0$ and $w_i = 0$. This plot is presented in Figure 7, with the theoretical cumulative

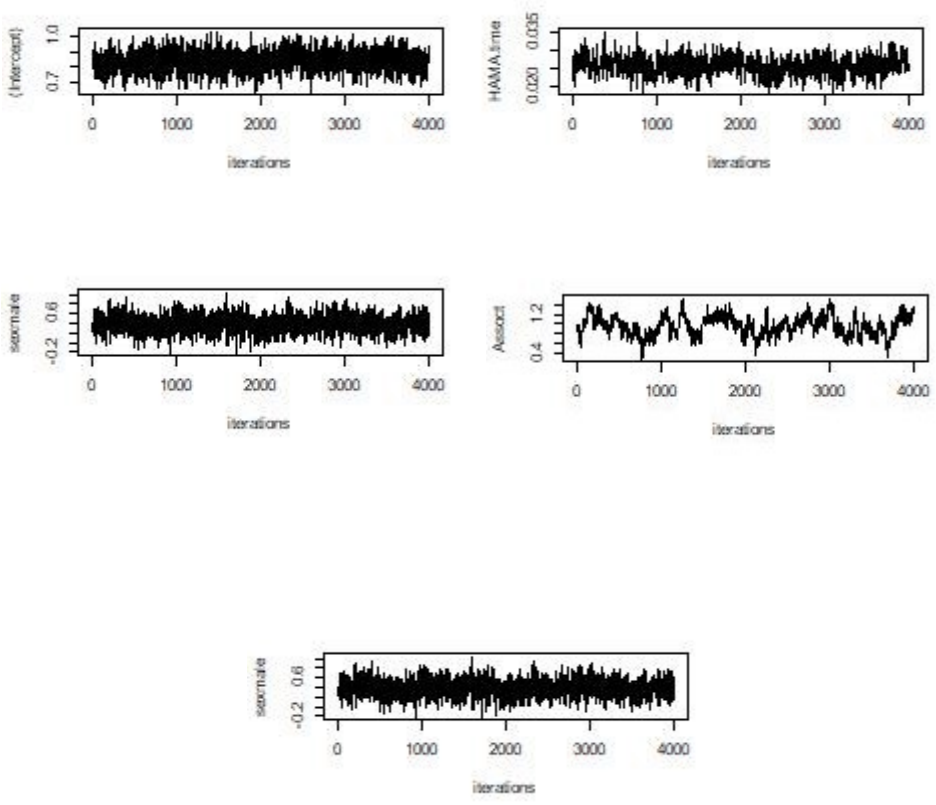

Figure 6. Trace plots for the β_0 (top left), β_1 (top right), and σ (middle left) parameters of the longitudinal submodel, and for the α and γ parameters of survival submodels from one iteration of the simulation, run for 4000 iterations after thinning and burn-in.

hazard in red. We found that the distribution of the generated event times closely followed that of the theoretical event times, although there was some departure from the theoretical distribution in the upper tail of the distribution where the data was sparse.

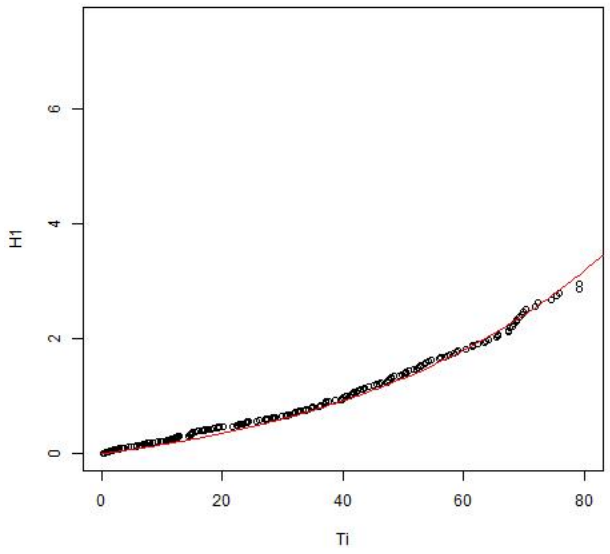

Figure 7. Nelson-Aalen estimate of the cumulative hazard function for the generated event times in black, and theoretical event times in red.

We produced a second set of Nelson-Aalen plots using data generated with $b_i = 0.1$ and $w_i = 0$, and with $b_i = 0$ and $w_i = 1$, as presented in Figures 8 and 9. We can see that, again, there is some departure from the theoretical distribution in the upper tail of the distribution where there is less data.

We discarded any individuals whose event time T_i is less than the first observation time $t_{i1} = 18.3$, in order to omit those individuals whose event times preceded their first observation time. Excluding these individuals from our analysis introduces truncation into the dataset. Approximately 33% of the individuals in the dataset were truncated. Discarding these subjects reduced the number of individuals from $n = 251$ to an average of 167 individuals per iteration. This number was representative of the real bipolar dataset, in which 167 individuals were included in the analysis after discarding those observations that were taken after an individual's event time. Summary statistics for these generated event times are presented in Table 9. The mean regression coefficient estimates

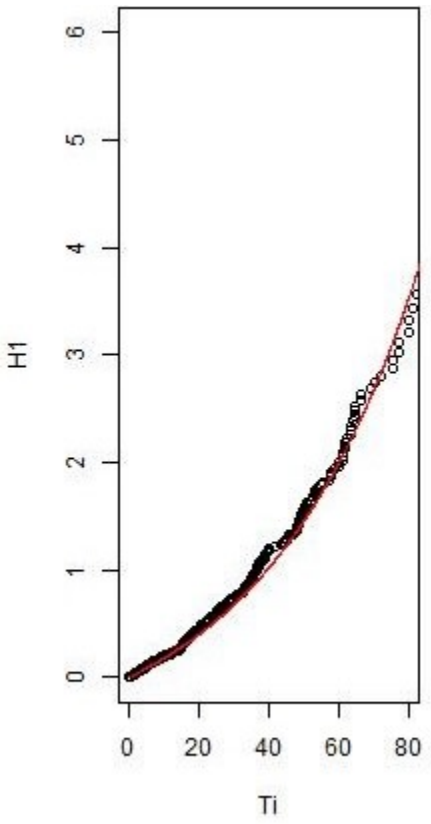

Figure 8. Nelson-Aalen estimates of the cumulative hazard function for the generated event times in black, and theoretical event times in red generated using $b_i = 0.1$ and $w_i = 0$.

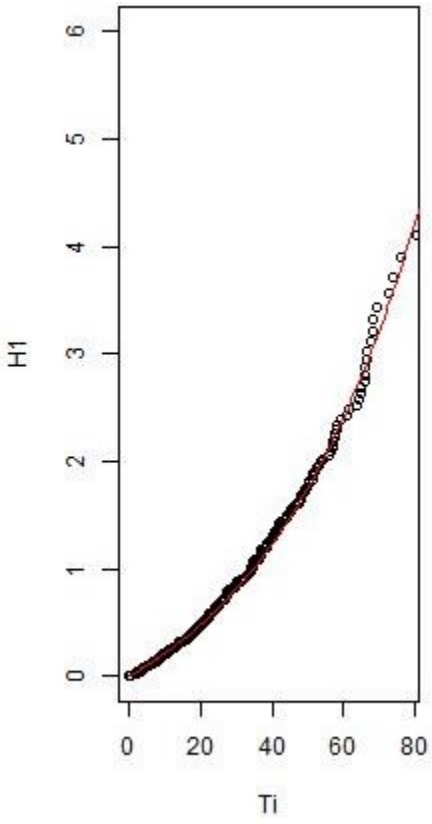

Figure 9. Nelson-Aalen estimates of the cumulative hazard function for the generated event times in black, and theoretical event times in red generated using $b_i = 0$ and $w_i = 1$.

Table 9. Summary statistics for the generated event times from simulation setting 1, before and after truncating the data, as well as for the real event times from the bipolar dataset after truncation.

	Generated Event Times		Real Bipolar Dataset Event Times
	Before Truncation	After Truncation	
Minimum	3.000e-4	18.30	7.68
First Quartile	13.36	29.02	20.59
Median	28.97	42.00	25.14
Mean	34.49	47.34	25.05
Third Quartile	50.08	60.46	28.36
Maximum	212.54	212.54	45.81

produced from running 5000 iterations of this simulation are presented in Table 10.

The estimates for the longitudinal submodels were closer to the true parameter values in the analysis that accounted for truncation than the analysis that ignored truncation. The estimate of β_1 demonstrated a larger relative percent bias in the analysis that accounted for truncation (-13.043%) as compared to the analysis that ignored truncation (-8.700%). The relative percent bias was approximately the same for the intercept estimate between both analyses. The average standard deviations and empirical standard deviations of the β_0, β_1, σ_b and σ estimates were very close across both analyses.

For the survival submodel, the analysis that accounted for truncation performed notably better than the analysis that ignored truncation. The relative percent bias of the γ estimate was 1.053% in the analysis that accounted for truncation, and 4.211% in the analysis that ignored truncation. Similarly, the relative percent bias of the α estimate was 3.942% in the analysis that accounted for truncation, and 10.581% in the analysis that ignored truncation. The empirical standard deviation of the survival regression parameter estimates were also larger in the analysis that ignored truncation. The empirical standard deviation of the γ estimate was 0.174 in the analysis that accounted for truncation, and 0.176 in the analysis that ignored truncation. This trend was also observed in the estimate of α, where the empirical standard deviation was 0.221 in the analysis that accounted for truncation and 0.242 in the analysis that ignored truncation.

Table 10. Mean parameter estimates and measures of standard deviation, taken over 5000 iterations of the simulation setting 1 and using the true values presented in Table 8.

Accounting for Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.007	-7.189	0.067	0.272
β_1 (time)	0.023	0.020	-13.043	0.003	0.013
σ_b	0.629	0.617	-1.098	0.056	0.038
σ	0.685	0.683	-0.292	0.023	0.022
Survival Submodel					
γ	0.285	0.288	1.053	0.174	0.177
α	0.964	1.002	3.942	0.211	0.221
Ignoring Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.004	-7.465	0.067	0.273
β_1 (time)	0.023	0.021	-8.700	0.003	0.013
σ_b	0.629	0.615	-2.226	0.056	0.038
σ	0.685	0.684	-0.146	0.023	0.023
Survival Submodel					
γ	0.285	0.297	4.211	0.176	0.181
α	0.964	1.066	10.581	0.222	0.24

Table 11. Summary statistics for the generated event times, before and after truncation, using the updated true β_1 value of 0.05 in the simulation setting 2.

	Generated Event Times		Real Bipolar Dataset Event Times
	Before Truncation	After Truncation	
Minimum	3.000e-5	18.30	7.68
First Quartile	11.61	25.03	20.59
Median	22.60	32.52	25.14
Mean	24.62	34.92	25.05
Third Quartile	35.30	42.46	28.36
Maximum	114.84	114.84	45.81

8.5 Simulation Results: Constant Truncation Time, $\beta_1 = 0.05$

We carried out a second simulation over 5000 iterations, with an updated β_1 value of 0.050 as presented in Table 8. We again introduced truncation into the dataset by discarding any individuals whose event time T_i is less than the first observation time $t_{i1} = 18.3$. The number of observations was reduced by approximately 47% (from 1004 to an average of 530.9 per iteration) after omitting the HAM-A measurement times that occurred after an individual's event time, reducing the dataset from $n = 251$ individuals to an average of 150 individuals per dataset. Summary statistics for the generated event times are presented in Table 11. The results of this analysis are presented in Table 12.

For the longitudinal submodels, the relative percent bias of the β_0, β_1, σ_b, and σ estimates fell within 1% of one another between both analyses. The relative percent bias of the β_0 estimate was -7.189% in the analysis that accounted for truncation, and -7.650% in the analysis that ignored truncation. The relative percent bias of the mean of the β_1 estimate taken over all 5000 iterations was approximately -8.000% in both analyses. Similarly, the relative percent bias of the mean σ_b estimate was -2.226% in the analysis that accounted for truncation, and -3.021% in the analysis that ignored truncation. The relative percent bias of the mean σ estimate was approximately 0% in both analysis. Comparing the analyses that accounted for versus ignored truncation, the average posterior standard deviation of the regression parameter estimates were the same to three decimal places, and the mean empirical standard deviations were the same to two decimal places.

Table 12. Mean parameter estimates and measures of standard deviation, taken over 5000 iterations of the simulation setting 2 and using an updated β_1 value of 0.05.

Accounting for Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.007	-7.189	0.070	0.298
β_1 (time)	0.050	0.046	-8.000	0.003	0.014
σ_b	0.629	0.615	-2.226	0.060	0.038
σ	0.685	0.683	-0.291	0.025	0.024
Survival Submodel					
γ	0.285	0.287	0.702	0.185	0.187
α	0.964	1.007	4.461	0.228	0.244
Ignoring Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.002	-7.650	0.070	0.299
β_1 (time)	0.050	0.046	-8.000	0.003	0.014
σ_b	0.629	0.610	-3.021	0.060	0.039
σ	0.685	0.685	0.000	0.025	0.025
Survival Submodel					
γ	0.285	0.302	5.965	0.190	0.196
α	0.964	1.146	18.880	0.256	0.312

The survival submodels demonstrated some considerable differences between analyses. The relative percent bias of the survival regression parameter estimates were demonstrably larger in the analysis that ignored truncation. The γ estimate had a relative percent bias of just 0.702% in the model that accounted for truncation, while it was 5.965% in the model that ignored truncation. The α estimates had a relative percent bias of 4.461% in the model that accounted for truncation, and 18.880% in the model that ignored truncation, indicating that the analysis that accounted for truncation performed much better in the estimation of the survival submodel parameters. The empirical standard deviations of the survival regression parameter estimates were moreover larger in the analysis that ignored truncation. The empirical standard deviation of the γ estimate was 0.187 in the model that accounted for truncation, and 0.196 in the model that ignored truncation. Similarly, the empirical standard deviation of the α estimate was 0.244 in the model that accounted for truncation, and 0.312 in the model that ignored truncation. This trend was also evident in the average posterior standard deviation of the survival estimates, which were smaller in the analysis that accounted for truncation.

8.6 Simulation Results: Random, Independent Truncation

We carried out a third simulation over 5000 iterations with randomly and independently truncated data. This differs from the previous two simulation settings, where the truncation times were constant (18.30 years). As described in section 9.2, we generated a truncation time for each individual in the dataset from a $lognormal(2.8206476,\, 0.3894846)$ distribution, and replaced the first observation time of 18.3 years with a randomly, independently generated truncation time. Any HAM-A observation times that preceded an individual's truncation time were omitted from the dataset. After generating a longitudinal response for each individual vector and event time T_i, we truncated the dataset by omitting any individuals whose event time T_i was less than their truncation time L_i. We then carried out the two joint model analyses which ignored and accounted for truncation. Summary statistics for these generated event times and truncation times are presented in Table 13. The number of observations was reduced by approximately 37% (from 721 to 455 per

Table 13. Summary statistics for the generated event times, before and after truncating the data, as well as for the independent truncation times.

	Generated Event Times		Truncation Times
	Before Truncation	After Truncation	
Minimum	0.000	3.72	2.37
First Quartile	13.38	26.58	11.77
Median	28.99	39.96	14.89
Mean	34.51	45.10	15.11
Third Quartile	50.15	58.80	18.37
Maximum	216.59	216.59	24.30

iteration, on average), which reduced each the number of individuals in each dataset from $n = 251$ to an average of 148 per dataset. The results of this analysis are presented in Table 14.

The presence of random, independent truncation in the data seemed to diminish the disparity in the results between the two analyses as compared to the analyses without random, independent truncation. The estimates of the longitudinal intercept were close to the true parameter values in both analyses. The mean β_1 estimate had a comparatively large relative percent bias in both analyses. The estimates of the variance components were close to the true values in both analyses. In both analyses, the estimate of σ_b and the estimate of σ had a very small relative percent bias.

There was little difference between the two analyses in the results of the survival submodel. The analysis that accounted for truncation and the analysis that ignored truncation both produced a mean estimate of γ with a very small relative bias (approximately 2% when accounting for truncation and 0% when ignoring truncation). The estimate of α demonstrated the most notable difference between these two analyses. The estimate of α had a relative percent bias of 10.789% in the analysis that accounted for truncation, and 12.241% in the analysis that ignored truncation. The average posterior standard deviation and mean empirical standard deviation of the survival submodel regression parameter estimates were also similar in the analysis that accounted for truncation. The mean empirical standard deviation of the γ estimate was 0.221 in the model that accounted for truncation and 0.223 in the model that ignored truncation. The mean empirical standard deviation of the α estimate was 0.301 in the model that accounted for truncation, and 0.317 in the model that ignored truncation. Overall, the differences between these analyses were less pronounced in the case that

Table 14. Mean parameter estimates and measures of standard deviation, taken over 5000 iterations of the simulation and using independently truncated data.

Accounting for Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.038	-4.332	0.070	0.166
β_1 (time)	0.023	0.019	-17.39	0.003	0.008
σ_b	0.629	0.618	-1.749	0.065	0.040
σ	0.685	0.684	-0.146	0.027	0.027
Survival Submodel					
γ	0.285	0.280	1.754	0.213	0.223
α	0.964	1.068	10.789	0.275	0.301
Ignoring Truncation	True Parameter	Average Estimate	Relative Percent Bias	Average Posterior Std. Dev	Empirical Std. Dev
Longitudinal Submodel					
β_0	1.085	1.037	-4.424	0.070	0.167
β_1 (time)	0.023	0.019	-17.39	0.003	0.008
σ_b	0.629	0.618	-1.749	0.065	0.040
σ	0.685	0.685	0.000	0.028	0.027
Survival Submodel					
γ	0.285	0.285	0.000	0.215	0.226
α	0.964	1.087	12.759	0.282	0.317

the data was randomly, independently truncated than in the simulations featuring non-random or dependently truncated data.

Chapter 9

Conclusion and Future Work

In this thesis, a comparative joint model analysis, using methods that both accounted for and ignored truncation, was carried out in two real datasets and in a simulation study. The relative percent bias, posterior standard deviation, and empirical standard deviation of the longitudinal and survival submodel parameter estimates were examined.

The comparative joint model analysis was carried out on a real dataset that recorded the time to diagnosis of bipolar disorder or major depressive disorder in children who had at least one parent with the diagnosis. This analysis revealed that accounting for truncation times produced a marked difference in the significance of the survival association parameter as compared to methods that ignored truncation. This comparative joint model analysis was also carried out on an AIDS dataset, where longitudinal and survival data were collected in patients who were intolerant or unresponsive to a conventional antiretroviral therapy, zidovudine. The joint model analyses that accounted for and ignored truncation produced similar results. Because truncation was artificially created by omitting baseline measurements, we were able to compare results from the truncated dataset to the original full dataset. This artificially truncated dataset omitted all time zero longitudinal measurements from our analysis. In contrast, the full dataset included a time zero longitudinal measurement for every individual. The results carried out on the truncated dataset produced similar results, regardless of whether or not the truncation was accounted for. There was a marked difference in the results analyses carried out on the truncated datasets when compared to that carried out on the full dataset featuring time zero measurements. The AIDS analysis also investigated a potential linear relationship between event times and truncation times, but found no evidence of this relationship.

In the simulation study, survival data was generated following the methods developed by Austin

(2012) and Stefan (2019). Data sets were truncated by both constant and random independent truncation times across three parameter settings. In the case of a constant truncation time and a true value of $\beta_1 = 0.023$, the longitudinal and survival submodel regression parameter estimates had smaller biases in the analysis that accounted for truncation as compared to the analysis that ignored truncation. The differences between these analyses were most pronounced in the survival submodel, and in particular in the association parameter estimate, where the relative percent bias was considerably smaller in the analysis that accounted for truncation. The average standard deviation and empirical standard deviation of the survival regression parameter estimates were also notably smaller in the analysis that accounted for truncation.

In the second simulation setting, the value of β_1 was changed from 0.023 to 0.05, while all other elements of the simulation remained the same. There was a pronounced difference in the performance of the survival submodel when comparing the analysis that accounted for truncation to the analysis that ignored truncation. The relative percent bias of the survival regression parameter estimates were substantially smaller in the analysis that accounted for truncation. The relative percent bias of the α estimate was 4.461% in the analysis that accounted for truncation, and 18.880% in the analysis that ignored truncation. Similarly, the relative percent bias of the γ estimate was 0.702% in the analysis that accounted for truncation, and 5.965% in the analysis that ignoted truncation. The average posterior standard deviation and average empirical standard deviation of the survival regression parameter estimates were similarly smaller in the analysis that ignored truncation. The relative percent bias, empirical standard deviation, and posterior standard deviations of the longitudinal submodel regression parameter estimates were comparable between analyses in this case.

The third simulation setting used the same values as the first, but introduced random, independent truncation to the dataset. There was again little difference between the analyses that ignored and accounted for truncation within the longitudinal submodel. Estimates of the longitudinal intercept, as well as of the estimate of σ_b and the estimate of σ, were very close to the true parameter values in both of these analyses. This trend was also observed in the survival submodel, where the regression parameter estimates were similar between analyses. In this case, the estimate of

the association parameter had a large relative percent bias even when truncation was accounted for. Measures of the posterior standard deviation and empirical standard deviation were smaller in the joint models that accounted for truncation as compared to those that did not. Among these simulation settings, the most pronounced differences between these analyses were apparent in the regression parameter estimates of the survival submodel.

The bipolar dataset featured several offspring that were closely related, and should be treated as clustered data. However, clustering was ignored in this thesis. Further research should aim to adjust joint models for the presence of clustering in addition to truncation. In particular, future research should investigate the computational costs of accommodating the presence of clustering into a longitudinal and/or survival submodel. The datasets featured in this thesis consisted of datasets in which the event times and truncation times were not linearly related to one another. Further research should explore the performance of these joint modeling methods when there is dependence between the truncation and event times.

Bibliography

Abrams, D. I., Goldman, A. I., Launer, C., Korvick, J., Neaton, J., Crane, L., Grodesky, M., Wakefield, S., Muth, K. & Kornegay, S. (1994), 'A comparative trial of didanosine or zalcitabine after treatment with zidovudine in patients with human immunodeficiency virus infection. The Terry Beirn Community Programs for Clinical Research on AIDS', *The New England journal of medicine* **330**(10), 657–662.

Andersen, P. K. & Gill, R. D. (1982), 'Cox's Regression Model for Counting Processes: A Large Sample Study', *The Annals of statistics* **10**(4), 1100–1120.

Clayton, D. & Bernardinelli, L. (1996), Bayesian methods for mapping disease risk, *in* 'Geographical and Environmental Epidemiology', Oxford University Press, Oxford.

Corporation, M. & Weston, S. (2022), *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.17.
URL: *https://CRAN.R-project.org/package=doParallel*

Demirtas, H. (2007), 'The design of simulation studies in medical statistics by Andrea Burton, Douglas G. Altman, Patrick Royston and Roger L. Holder, Statistics in Medicine 2006; 25:4279-4292', *Statistics in medicine* **26**(20), 3818–3821.

Duffy, A., Grof, P., Robertson, C. & Alda, M. (2000), 'The implications of genetic studies of major mood disorders for clinical practice', *The Journal of clinical psychiatry* **61**(9), 630–637.

Duffy, A., Horrocks, J., Doucette, S., Keown-Stoneman, C., McCloskey, S. & Grof, P. (2014), 'The developmental trajectory of bipolar disorder', *British journal of psychiatry* **204**(2), 122–128.

Edwards, W., Lindman, H. & Savage, L. J. (1963), 'Bayesian statistical inference for psychological research.', *Psychological review* **70**(3), 193.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. & Gelman, A. (2019), 'Visualization in Bayesian workflow', *Journal of the Royal Statistical Society. Series A, Statistics in society* **182**(2), 389–402.

Gelman, A., Carlin, J. B., Stern, H. S. & Rubin, D. B. (2004), *Bayesian Data Analysis*, 2nd ed. edn, Chapman and Hall/CRC.

Gilks, W., Richardson, S. & Spiegelhalter, D. (1995), *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC Interdisciplinary Statistics, Taylor & Francis.
URL: *http://books.google.com/books?id=TRXrMWY_i2IC*

Goldman, A. I., Carlin, B., Crane, L. R., Launer, C., Korvick, J. A., Deyton, L. & Abrams, D. I. (1996), 'Response of CD4 lymphocytes and clinical consequences of treatment using ddI or ddC in patients with advanced HIV infection', *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology* **11**(2), 161–169.

Goodrich, B., Gabry, J., Ali, I. & Brilleman, S. (2022), 'rstanarm: Bayesian applied regression modeling via Stan.'. R package version 2.21.3.
URL: *https://mc-stan.org/rstanarm/*

Hamilton, M. (1960), 'A rating scale for depression', *Journal of neurology, neurosurgery, and psychiatry* **23**(1), 56.

Klein, J. P. & Moeschberger, M. L. (2006), *Survival Analysis: Techniques for Censored and Truncated Data*, Statistics for Biology and Health, second edition. edn, Springer, New York, NY.

Lawless, J. F. (2011), *Statistical models and methods for lifetime data*, John Wiley & Sons.

Liu, Y., Li, J. & Zhang, X. (2018), 'Analysis of dependently truncated data in Cox framework', *Communications in statistics. Simulation and computation* **47**(6), 1677–1695.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D. & R Core Team (2020), *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-148.
URL: *https://CRAN.R-project.org/package=nlme*

R Core Team (2020), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org*

Rizopoulos, D. (2010), 'JM: An R package for the joint modelling of longitudinal and time-to-event data', *Journal of Statistical Software (Online)* **35**(9), 1–33. R package version 1.4-8.

Rizopoulos, D. (2012), *Joint models for longitudinal and time-to-event data: With applications in R*, CRC press.

Rizopoulos, D. (2016), 'The R Package JMbayes for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC', *Journal of Statistical Software* **72**(7), 1–45.

Rizopoulos, D., Papageorgiou, G. & Miranda Afonso, P. (2022), *JMbayes2: Extended Joint Models for Longitudinal and Time-to-Event Data*. R package version 0.2-8.
URL: *https://CRAN.R-project.org/package=JMbayes2*

Stefan, G. (2019), A Comparison of Cox and Joint Models for Time-to-Event Data, Master's thesis, University of Guelph.

Therneau, T. M. (2020), *A Package for Survival Analysis in R*. R package version 3.1-12.
URL: *https://CRAN.R-project.org/package=survival*

Therneau, T. M. & Grambsch, P. M. (2000), *The Cox model*, Springer.

Van Buuren, S. & Groothuis-Oudshoorn, K. (2011), 'mice: Multivariate Imputation by Chained Equations in R', *Journal of Statistical Software* **45**(3), 1–67.

Venables, W. N. & Ripley, B. D. (2002), *Modern Applied Statistics with S*, fourth edn, Springer, New York. ISBN 0-387-95457-0.

URL: *http://www.stats.ox.ac.uk/pub/MASS4*

R Code

A.1 Aids Analysis Code

```
#Sys.setenv(JAGS_HOME="C:/Program Files/JAGS/JAGS-4.3.0")
#library(rjags)

library(JMbayes)
library(lattice)
library(dplyr)
library(survival)
library(survminer)

data(aids, package ="JM")
data(aids.id, package ="JM")

#artificially left truncate dataset
aids.trunc <- aids[aids$start != 0, ]

#patient <- data.frame("patient" = unique(aids.trunc$patient))
#aids.trunc.id <- merge(patient, aids.id, by = "patient")
#aids.trunc.id$patient <- factor(sort(as.numeric(aids.trunc.id$patient)))


aids.trunc.id <- aids.trunc %>%
group_by(patient) %>%
mutate(start = replace(start, 1, 0))


#library(data.table)
#aids.id$start2 <- length(aids[ , .SD[which.min(start)], by = patient]$start)
#nrow(aids.id)

#average number of visits per person full
length(aids$patient)/length(levels(aids$patient))
#average number of visits per person truncated
length(aids.trunc$patient)/length(levels(aids.trunc$patient))

####HISTOGRAM OF VISIT FREQUENCY###

#vector of number of visits per individual
v <- as.vector(table(aids$patient))
```

```
df <- aids %>%
group_by(patient) %>%
summarise(counts = n()) %>% as.data.frame()


#pdf("nreq.pdf")
#hist(sort(v), xlab = "Number of Visits", main = "Number of Visits
\nfor Individuals under Study",
#     ylim=c(0,200))
#dev.off()

#table of number of visits
count(df, factor(df$counts))

#MINIMUM VALUE
min(v)

#MAXIMUM VALUE
max(v)

#full: 71% censored
length(which(aids$death == 0))/length(aids$death)
#truncated: 76% censored
length(which(aids.trunc$death == 0))/length(aids.trunc$death)

#29% dead
length(which(aids$death == 1))/length(aids$death)


#number males full
length(which(aids.id$gender == "male")) #422
#number males truncated
aids.surv <- aids.trunc[!duplicated(aids.trunc$patient), ]
length(which(aids.surv$gender == "male")) #373

#number females
length(which(aids.id$gender == "female")) #45
length(which(aids.surv$gender == "female")) #33

#number AZT intolerant
length(which(aids.id$AZT == "intolerance")) #292
#number AZT intolerant truncated
length(which(aids.surv$AZT == "intolerance")) #261
```

```
#number AZT failure
length(which(aids.id$AZT == "failure")) #175
#number AZT truncated
length(which(aids.surv$AZT == "failure")) #145

#previous infection
length(which(aids.id$prevOI == "AIDS")) #307
#previous infection truncated
length(which(aids.surv$prevOI == "AIDS")) #259

#drug status
length(which(aids.id$drug == "ddC")) #237
#drug status truncated
length(which(aids.surv$drug == "ddC")) #208


#drug status
length(which(aids.id$drug == "ddI")) #230
#drug status truncated
length(which(aids.surv$drug == "ddI")) #198


#average time between visits

#consider only id and 'obstime' columns, call it newDF
newDF <- aids[,c(1, 5)]
newDF.t <- aids.trunc[,c(1, 5)]

#take difference of row values over year column
#and look at the result
head(apply( newDF[-1] , 2 , diff ),20)

#notice the negative  values
#these are the values where a year value was subtracted from 0
#i.e. the last Year measurement from the previous id was subtracted
#from the first year measuremnt (0) of the current ID

#take this output and simply remove these negative values
#which represent the nonsensical result in difference in
#measurement times between two different individuals

year.diff <- as.data.frame(apply( newDF[-1] , 2 , diff ))
truedf <- year.diff[year.diff$obstime >= 0, ]

year.diff.t <- as.data.frame(apply( newDF.t[-1] , 2 , diff ))
```

```
truedf.t <- year.diff.t[year.diff.t$obstime >= 0, ]

#compare these differences to head of relevant columns of pbc2 dataset
head(truedf,10)
head(newDF,10)


mean(truedf) ###true df
mean(truedf.t) ###true df truncated
max(truedf)
min(truedf)
(var(truedf))^0.5

#Survival by Treatment
km.aids.drug <-  survfit(Surv(Time, death) ~ drug, data = aids.id)

jpeg("bydrug.jpg")
ggsurvplot(km.aids.drug, #risk.table = TRUE,
conf.int = TRUE,
size = 0.5,
xlab = 'months',
data = aids.id) +
ggtitle('Survival by Drug')
dev.off()

#by AZT tolerance status

km.aids.azt <-  survfit(Surv(Time, death) ~ AZT, data = aids.id)

jpeg("byazt.jpg")
ggsurvplot(km.aids.azt, risk.table = TRUE, conf.int = TRUE,
size = 0.5,
xlab = 'months',
data = aids.id) +
ggtitle('Survival by AZT Tolerance')
dev.off()

#by gender

km.aids.g <-  survfit(Surv(Time, death) ~ gender, data = aids.id)

jpeg("bygender.jpg")
ggsurvplot(km.aids.g, risk.table = TRUE, conf.int = TRUE,
size = 0.5,
xlab = 'months',
```

```
data = aids.id) +
ggtitle('Survival by Gender')
dev.off()

#cd4 count over time per patient, by gender

jpeg("cd4lme.jpg")
ggplot(aids, aes(x = obstime, y = CD4, group = patient, colour = gender)) +
geom_line(alpha = 0.3) +
scale_x_continuous('months') +
ggtitle('CD4 counts over time by Gender')
dev.off()

#spaghetti plot for cd4 cell counts over time

#CD4 Scores plot
jpeg("cd4xy.jpg")
xyplot(CD4 ~ obstime , group = patient, data = aids.trunc, xlab = "Observation Time",
ylab ="CD4 Cell Count", type = "l")
dev.off()

############################################################
#################### Model Fitting ########################
############################################################

#use same lme throughout

cd4.lme <- lme(sqrt(CD4) ~ obstime,
data = aids.trunc,
random = ~ obstime| patient,
method = "REML")
summary(cd4.lme)

#full dataset
#no truncation
cd4.lme.full <- lme(sqrt(CD4) ~ obstime,
data = aids,
random = ~ obstime| patient,
method = "REML")

############################################################
########### Model 1: No Interactions ######################
############################################################

#Wrong: No Truncation
```

```
c.wrong.1 <- coxph(Surv(start, stop, event) ~ drug + gender + AZT +
cluster(patient), data=aids.trunc.id, x=TRUE, model=TRUE)
summary(c.wrong.1)

jm.wrong.1 <- jointModelBayes(cd4.lme, c.wrong.1, timeVar = "obstime")
summary(jm.wrong.1)

#Account for Truncation
#use data in long format
#cluster by patient

c.right.1 <- coxph(Surv(start, stop, event) ~ drug + gender + AZT +
cluster(patient), data=aids.trunc, x=TRUE, model=TRUE)
summary(c.right.1)

jm.truncated.1 <- jointModelBayes(cd4.lme, c.right.1, timeVar = "obstime")
summary(jm.truncated.1)


#full dataset
#no truncation

#survival function
cox.full.1 <- coxph(Surv(Time, death) ~ drug + gender + AZT,
data = aids.id, x = TRUE, model = TRUE)

jm.full.1 <- jointModelBayes(cd4.lme.full, cox.full.1, timeVar = "obstime")
summary(jm.full.1)



############################################################
########### Model 2: No Interactions ###################
########### Previous Infection Status ##################
############################################################

#Wrong: No Truncation

c.wrong.2 <- coxph(Surv(start, stop, event) ~ drug + gender + AZT + prevOI +
cluster(patient), data=aids.trunc.id, x=TRUE, model=TRUE)
summary(c.wrong.2)

jm.wrong.2 <- jointModelBayes(cd4.lme, c.wrong.2, timeVar = "obstime")
summary(jm.wrong.2)
```

```
#Account for Truncation
#use data in long format
#cluster by patient

c.right.2 <- coxph(Surv(start, stop, event) ~ drug + gender + AZT + prevOI+
cluster(patient), data=aids.trunc, x=TRUE, model=TRUE)
summary(c.right.2)

jm.truncated.2 <- jointModelBayes(cd4.lme, c.right.2, timeVar = "obstime")
summary(jm.truncated.2)

#full dataset
#no truncation

#survival function
cox.full.2 <- coxph(Surv(Time, death) ~ drug + gender + AZT + prevOI,
data = aids.id, x = TRUE)

jm.full.2 <- jointModelBayes(cd4.lme.full, cox.full.2, timeVar = "obstime")
summary(jm.full.2)

############################################################
########### Model 3: No Interactions ###################
########### Are drug, gender sufficient? ###############
############################################################

#Wrong: No Truncation

c.wrong.3 <- coxph(Surv(start, stop, event) ~ drug + gender +
cluster(patient), data=aids.trunc.id, x=TRUE, model=TRUE)
summary(c.wrong.3)

jm.wrong.3 <- jointModelBayes(cd4.lme, c.wrong.3, timeVar = "obstime")
summary(jm.wrong.3)

#Account for Truncation
#use data in long format
#cluster by patient

c.right.3 <- coxph(Surv(start, stop, event) ~ drug + gender +
cluster(patient), data=aids.trunc, x=TRUE, model=TRUE)
summary(c.right.3)

jm.truncated.3 <- jointModelBayes(cd4.lme, c.right.3, timeVar = "obstime")
```

```
summary(jm.truncated.3)

#full dataset
#no truncation

#survival function
cox.full.3 <- coxph(Surv(Time, death) ~ drug + gender,
data = aids.id, x = TRUE)

jm.full.3 <- jointModelBayes(cd4.lme.full, cox.full.3, timeVar = "obstime")
summary(jm.full.3)
```

A.2 Bipolar Analysis Code

```
library(tidyr)
library(tidyverse)
library(joineR)
library(dplyr)
library(lattice)
#library(here)
library(psych)

#import
bpd <- read.csv("temp.csv")


#in censored individuals
#replace age at onset w/ age at last interview
for (i in 1:length(bpd$OUTCOME1BAGE)){
if (is.na(bpd$OUTCOME1BAGE[i]) == TRUE){
bpd$OUTCOME1BAGE[i] = bpd$AGELASTINT[i]
}
}

#Remove individuals w/ no HAM-A Scores
length(unique(bpd$ID))#305
bpd <- bpd[complete.cases(bpd$HAMAAGE_1),] #32 w/ no HAM-A obs
bpd <- bpd[complete.cases(bpd$HAMATOT_1),]
length(unique(bpd$ID))#273 individuals left


#SES Relevel
bpd$PARENTSES_1[bpd$PARENTSES_1 == 1] <- 3
bpd$PARENTSES_1[bpd$PARENTSES_1 == 2] <- 3
```

```
#set level 5 as referent
bpd <- within(bpd, PARENTSES_1 <- relevel(as.factor(PARENTSES_1), ref = "5"))

#long data
attach(bpd)
df.temp <-cbind(ID, SEX, LITHRESP, PARENTSES_1, PARONSAGE, AGEFIRSTINT,
AGELASTINT, OUTCOME1B, OUTCOME1BAGE)

#hama ages
hama.age <- cbind(bpd[, 16], bpd[, seq(19, 47, by=2)])
colnames(hama.age)[1] <- "HAMAAGE_1"

#hama scores
hama <- cbind(bpd[, 17], bpd[, seq(18, 46, by=2)])
colnames(hama)[1] <- "HAMATOT_1"

detach(bpd)

hamaageunbalanced <- to.unbalanced(data.frame(cbind(df.temp,hama.age)),id.col=1,
times=1:16,Y.col=10:25,other.col=2:9)
hamaunbalanced<-to.unbalanced(data.frame(cbind(df.temp,hama)),id.col=1,
times=1:16,Y.col=10:25,other.col=2:9)
hamaunbalanced$HAMAAGE_1 <- hamaageunbalanced$HAMAAGE_1
HAMA<- hamaunbalanced

#discard individuals w/ HAMA scores
taken at, after outcome
length(unique(HAMA$ID))
HAMA <- filter(HAMA, HAMA$HAMAAGE_1 < HAMA$OUTCOME1BAGE) #106 people omitted
length(unique(HAMA$ID)) #167


#log transform HAMA
HAMA$loghama <- log(HAMA$HAMATOT_1+ 1)
HAMA$loghama

#survival dataset
HAMA.surv <- HAMA[!duplicated(HAMA$ID), ]

#create survival object start, stop columns
HAMA$start <- HAMA$HAMAAGE_1
splitID <- split(HAMA[c("start", "HAMAAGE_1")], HAMA$ID)
HAMA$stop <- unlist(lapply(splitID, function(d) c(d$start[-1], d$HAMAAGE_1[1])))

#for censored individuals, replace last stop time with age at last interview
```

```
HAMA <- HAMA %>%
group_by(ID) %>%
mutate(stop = ifelse(OUTCOME1B == 0, c(stop[-n()], tail(AGELASTINT,1)), stop)) %>%
as.data.frame()

#for individuals who experienced onset
#replace last stop time with age at onset
HAMA <- HAMA %>%
group_by(ID) %>%
mutate(stop = ifelse(OUTCOME1B == 1, c(stop[-n()], tail(OUTCOME1BAGE,1)), stop)) %>%
as.data.frame()

#for individuals who experienced onset
#replace up to last indicator with 0

HAMA$event <- rep(0, nrow(HAMA))

temp <- HAMA %>%
group_by(ID) %>%
mutate(event = ifelse(OUTCOME1B == 1, c(event[-n()], tail(OUTCOME1B,1)), event)) %>%
as.data.frame()

#check
cbind(head(HAMA$ID, 100), head(HAMA$OUTCOME1B, 100), head(temp$event, 100)) #okay

HAMA$OUTCOME1B <- temp$event

#na check
which(is.na(HAMA$HAMATOT_1))
#at row 207 time 5, there is an HAMA time but not HAMA measurement
#drop this row of long dataset
HAMA <- HAMA[-207, ]


############################## Exploratory Plots ############################

#HAMA Scores plot
jpeg("hamaxy.jpg")
xyplot(HAMATOT_1 ~ HAMAAGE_1 , group = ID, data = HAMA, xlab = "Age",
ylab ="HAM-A Score", type = "l")
dev.off()

#histogram of log-transformed HAMA Scores
jpeg("hamahist2.jpg")
hist(HAMA$loghama, xlab = "Log HAM-A score", ylab = "Frequency",
```

```
main = "")
dev.off()


############################## Exploratory Analysis ##########################

#number males after discarding no HAMA individuals
length(which(HAMA.surv$SEX == "1"))

#number females after discarding no HAMA individuals
length(which(HAMA.surv$SEX == "2"))

#number of individuals who did not experience event
length(which(HAMA.surv$OUTCOME1B=="0"))
#who did experience event
length(which(HAMA.surv$OUTCOME1B=="1"))

#first visit dates
summary(HAMA.surv$HAMAAGE_1)

#average number of years followed
summary(HAMA.surv$AGELASTINT - HAMA.surv$HAMAAGE_1)

#last visit dates
summary(HAMA.surv$AGELASTINT)


#SES Scores
length(which(HAMA.surv$PARENTSES_1 == "1"))
length(which(HAMA.surv$PARENTSES_1 == "2"))
length(which(HAMA.surv$PARENTSES_1 == "3"))
length(which(HAMA.surv$PARENTSES_1 == "4"))
length(which(HAMA.surv$PARENTSES_1 == "5"))

#lithium response
length(which(HAMA.surv$LITHRESP=="1"))
length(which(HAMA.surv$LITHRESP=="0"))

#parental onset age
mean(na.omit(HAMA.surv$PARONSAGE))

#mean visits
visits <- HAMA %>%
group_by(ID) %>%
select(HAMAAGE_1) %>%
```

```
na.omit() %>%
summarise(counts = n()) %>% as.data.frame()

visits <- as.data.frame(visits)

summary(visits)

#table of number of visits
count(visits, factor(visits$counts))

######################## CASE EXAMINE ###################################


#CASE 1
#1 HAMA measurement, same time as last interview, censored
#dropped
#ID 4

#CASE 2
#1 HAMA measurement, preceding last interview, censored
#ID 9
#HAMA measurement at 30.23, last interview at 31.02

HAMA[HAMA$ID==9,] #long dataset
bpd[bpd$ID==9,] #original dataset


#CASE 3
#1 HAMA obs, then event
#none

#2 or more obs, censored
#ID 1
#first interview 11.55
#first HAMA 13.9
#last HAMA 20.63
#last interview 21.52
HAMA[HAMA$ID==1,] #long dataset
bpd[bpd$ID==1,] #original dataset

#2 or more obs, event
#ID 2
#first interview 8.63
#first HAMA 10.98
#event at age 12.95
```

```
#subsequent HAMAs discarded
HAMA[HAMA$ID==2,] #long dataset
bpd[bpd$ID==2,] #original dataset

##########################################################
###################### Model Fitting ####################
##########################################################

#incorrect dataset, ignoring truncation
HAMA.notrunc <- HAMA
#replace start value with 0
HAMA.id<- HAMA.notrunc %>%
group_by(ID) %>%
mutate(start = replace(start, 1, 0)) %>% as.data.frame()



########################## Linear Mixed Model ################################
hama.lme <- lme(loghama ~ HAMAAGE_1,
random = ~ HAMAAGE_1| ID,
data = HAMA, method = "REML")
summary(hama.lme)


############################Model 1#############################

#correct
hama.1 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + cluster(ID),
data=HAMA, x=TRUE, model=TRUE)
summary(hama.1)

jm.hama.1 <- jointModelBayes(hama.lme, hama.1, timeVar = "HAMAAGE_1")
summary(jm.hama.1)

#incorrect
hama.w1 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + cluster(ID),
data=HAMA.id, x=TRUE, model=TRUE)
summary(hama.w1)

jm.hama.w1 <- jointModelBayes(hama.lme, hama.w1, timeVar = "HAMAAGE_1")
summary(jm.hama.w1)

############################Model 2#############################

#correct
```

```
hama.2 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX +
LITHRESP +  cluster(ID),
data=HAMA, x=TRUE, model=TRUE)
summary(hama.2)

jm.hama.2 <- jointModelBayes(hama.lme, hama.2, timeVar = "HAMAAGE_1")
summary(jm.hama.2)


#incorrect
hama.w2 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + LITHRESP+ cluster(ID),
data=HAMA.id, x=TRUE, model=TRUE)
summary(hama.w2)

jm.hama.w2 <- jointModelBayes(hama.lme, hama.w2, timeVar = "HAMAAGE_1")
summary(jm.hama.w2)


############################Model 3#############################

#correct
hama.3 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX +
LITHRESP + as.factor(PARENTSES_1) + cluster(ID),
data=HAMA, x=TRUE, model=TRUE)
summary(hama.3)

jm.hama.3 <- jointModelBayes(hama.lme,hama.3, timeVar = "HAMAAGE_1")
summary(jm.hama.3)


#incorrect
hama.w3 <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + LITHRESP+
as.factor(PARENTSES_1)+ cluster(ID),
data=HAMA.id, x=TRUE, model=TRUE)
summary(hama.w3)

jm.hama.w3 <- jointModelBayes(hama.lme, hama.w3, timeVar = "HAMAAGE_1")
summary(jm.hama.w3)
```

A.3 Simulation Study Code

```
#Simulation Setting 1
##############################################################################
#Sys.setenv(JAGS_HOME="C:/Program Files/JAGS/JAGS-4.3.0")
library(tidyr)
```

```
library(tidyverse)
library(dplyr)
library(JMbayes)
library(survival)
#library(broom)
#require(caTools)

#iterations
n = 5000

#individuals
i = 251

#constant baseline hazard for exp dist assume
lambda = 0.005


#observation times
a = 18.30  #mean first visit date
c = 24.71 #mean last visit date
HAMA.time = rep(seq(a, c, by = 2), i)

#mean visits per subject before discarding individuals
#w/ HAMA scores taken at/after event
v = 4

#log HAM-A scores
#real data
#real.hama <- lme(loghama ~ HAMAAGE_1,
#   random = ~ 1| ID,
#  data = HAMA, method = "REML")

#summary(real.hama)$coefficients[1]
b0 = 1.08455333 #b0
b1 = 0.02331502 #b1
sdp = 0.6290531 #subject specific raneff
sde = 0.6851948 #residual
pm = 0.455 #probability male

##############################################################################
#survival model with only sex as predictor
#real.surv <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + cluster(ID) ,
#      data=HAMA, x=TRUE, model=TRUE)
#summary(real.surv)
```

```
#JM for assoct, sex parameter estimate
#real.JM <- jointModelBayes(real.hama, real.surv, timeVar = "HAMAAGE_1")
#summary(real.JM)
alpha = 0.9642 #assoct



#regression estimate for time fixed parameter
estgamma = 0.2854

#turn it into a function

sim_fun = function(x) {
bi <- rep(rnorm(i, 0, sdp), rep(v, i))
me <- rnorm(v*i, 0, sde)
sexn <- rbinom(i,1,pm)
tempeventtime <- (1 / (b1 * alpha)) * log(1 - b1 *
alpha * log(runif(i, 0, 1)) /
(lambda * exp(estgamma*sexn + alpha *
(b0 + unique(bi)))))
sexn <- rep(sexn, each=4)
eventtime <- rep(tempeventtime, each=4)
simdf <- data.frame(ID = sort(rep(c(1:i), v)), HAMA.time, bi, me, sexn, eventtime)
simdf$resp = with(simdf, b0 + b1*HAMA.time + bi + me )
simdf$sex <- factor(simdf$sexn,
levels=c(0,1),
labels=c("female","male"))
simdf$outcome = with(simdf, rep(1, rep(v*i))) #why do they all get event?

#if outcome happens, stop time = event time
#start times
simdf$start <- simdf$HAMA.time
splitID <- split(simdf[c("start", "HAMA.time")], simdf$ID)
#stop times
simdf$stop <- unlist(lapply(splitID, function(d) c(d$start[-1], d$HAMA.time[1])))

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

simdf
}

set.seed(99)
data1 = lapply(1:n, sim_fun)
```

```
#summary statistics event times before truncation
event.t <- data1[[1]][!duplicated(data1[[1]]$ID), ]$eventtime
ID <- data1[[1]][!duplicated(data1[[1]]$ID), ]$ID
for(i in 2:n){
event.t <- c(event.t, data1[[i]][!duplicated(data1[[i]]$ID), ]$eventtime)
ID <- c(ID, data1[[i]][!duplicated(data1[[i]]$ID), ]$ID)
}
summary(event.t)
length(ID) #full set

sim_fun2 = function(data1){
#remove any observation times generated after the event time
data1 = filter(data1, data1$HAMA.time < data1$eventtime)

#for individuals who experienced onset
#replace last stop time with age at onset
data1 <- data1 %>%
group_by(ID) %>%
mutate(stop = ifelse(outcome == 1, c(stop[-n()], tail(eventtime,1)), stop)) %>%
as.data.frame()

#for individuals who experienced onset
#replace up to last indicator with 0

data1$event <- rep(0, nrow(data1))

temp <- data1 %>%
group_by(ID) %>%
mutate(event = ifelse(outcome == 1, c(event[-n()], tail(outcome,1)), event)) %>%
as.data.frame()

data1$outcome <- temp$event

data1
}

data2 <- lapply(data1, sim_fun2)

#Summary Statistics after truncation
event.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$eventtime
ID.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$ID
for(i in 2:n){
event.trunc <- c(event.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$eventtime)
ID.trunc <- c(ID.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$ID)
}
```

```
summary(event.trunc)
length(ID.trunc) #length ID.trunc

#proportion thrown out
(length(event.t) - length(event.trunc))/length(event.t)

#Individuals per dataset
(length(ID.trunc))/n

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

jm.fit = function(data2){


#linear model
sim.lm <- lme(resp ~ HAMA.time,
random = ~ 1| ID,
data = data2, method = "REML")

#incorrect dataset, ignoring truncation
data2.notrunc <- data2

#replace start value with 0
data2.id<- data2.notrunc %>%
group_by(ID) %>%
mutate(start = replace(start, 1, 0)) %>% as.data.frame()

#cox model w/ truncation
sim.surv <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2, x = TRUE, model = TRUE)

#cox model w/out truncation
sim.surv.nt <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2.id, x = TRUE, model = TRUE)

#joint model, accounting for truncation
sim.jm <- jointModelBayes(sim.lm, sim.surv, timeVar = "HAMA.time")
e1 <- summary(sim.jm)
#summary(sim.jm)

#joint model, ignoring truncation
sim.jm.nt <- jointModelBayes(sim.lm, sim.surv.nt, timeVar = "HAMA.time")
e2<- summary(sim.jm.nt)

estimates <- data.frame(e1$`CoefTable-Long`[1],  #intercept parameter truncated
```

```
e2$`CoefTable-Long`[1], #intercept parameter not truncated
e1$`CoefTable-Long`[3],  #intercept std error truncated
e2$`CoefTable-Long`[3], #intercept std error not truncated
e1$`CoefTable-Long`[5],  #intercept std dev truncated
e2$`CoefTable-Long`[5], #intercept std dev not truncated
e1$`CoefTable-Long`[11], #intercept p-value truncated
e2$`CoefTable-Long`[11], #intercept p-value not truncated
e1$`CoefTable-Long`[2], #hama.time parameter truncated
e2$`CoefTable-Long`[2], #hama.time parameter not truncated
e1$`CoefTable-Long`[4], #hama.time std error truncated
e2$`CoefTable-Long`[4], #hama.time std error not truncated
e1$`CoefTable-Long`[6], #hama.time std error truncated
e2$`CoefTable-Long`[6], #hama.time std error not truncated
e1$`CoefTable-Long`[12], #hama.time parameter p value truncated
e2$`CoefTable-Long`[12], #hama.time parameter p value not truncated
e1$`CoefTable-Event`[1], #sexmale parameter truncated
e2$`CoefTable-Event`[1], #sexmale parameter not truncated
e1$`CoefTable-Event`[4], #sexmale Std.Err truncated
e2$`CoefTable-Event`[4], #sexmale Std.Err not truncated
e1$`CoefTable-Event`[7], #sexmale Std.Dev truncated
e2$`CoefTable-Event`[7], #sexmale Std.Dev not truncated
e1$`CoefTable-Event`[16], #sexmale parameter p-value truncated
e2$`CoefTable-Event`[16], #sexmale parameter p-value not truncated
e1$`CoefTable-Event`[2], #assoct parameter truncated
e2$`CoefTable-Event`[2], #assoct parameter not truncated
e1$`CoefTable-Event`[5], #assoct Std.Err truncated
e2$`CoefTable-Event`[5], #assoct Std.Err not truncated
e1$`CoefTable-Event`[8], #assoct Std.Dev truncated
e2$`CoefTable-Event`[8], #assoct Std.Dev not truncated
e1$`CoefTable-Event`[17], #assoct p-value truncated
e2$`CoefTable-Event`[17], #assoct p-value not truncated
sqrt(e1$D), #stdev int truncated
sqrt(e2$D), #stdev int not truncated
e1$sigma, #stdev residuals truncated
e2$sigma, #stdev residuals not truncated
sim.jm$StDev$D, #stdev of D truncated
sim.jm.nt$StDev$D, #stdev of D not truncated
sim.jm$StDev$sigma, #stdev of sigma truncated
sim.jm.nt$StDev$sigma #stdev of sigma not truncated

)
estimates
}

#out.sim <- lapply(data2, jm.fit)
```

```
#JMmeans <- colMeans(do.call(rbind, out.sim))

############################## In Parallel #####################################

#detectCores()


nodeslist <- unlist(strsplit(Sys.getenv("NODESLIST"), split = ""))
z = as.numeric(Sys.getenv(("SLURM_ARRAY_TASK_ID")))

#cl = makeCluster(nodeslist, type = "PSOCK")
#registerDoParallel(cl)

library(parallel)
library(magrittr)
library(dplyr)

ncores=detectCores()
cl=makePSOCKcluster(ncores)

clusterExport(cl,
c("i", "v", "b0", "b1", "sdp",
"sde", "alpha", "lambda", "estgamma",
"pm", "HAMA.time", "data1"))

clusterEvalQ(cl,
list(attach(loadNamespace("JMbayes"), name = "JMbayes"),
attach(loadNamespace("nlme"), name = "nlme"),
attach(loadNamespace("survival"), name = "survival"),
attach(loadNamespace("dplyr"), name = "dplyr"),
attach(loadNamespace("magrittr"), name = "magrittr"),
attach(loadNamespace("parallel"), name = "parallel")))

#set seed among clusters
#s <- set.seed(99)
#nextRNGStream(s)
clusterSetRNGStream(cl=cl, 99)

#run in parallel
out.sim <- parLapply(cl=cl,data2, jm.fit)


colMeans(do.call(rbind, out.sim))
```

```
######################Empirical Standard Deviations############################

### truncated

#st dev intercept truncated
int.trunc <- numeric(n)
for(i in 1:n){
int.trunc[i] <- as.numeric(out.sim[[i]][1])
}
(var(int.trunc))^0.5
sd(int.trunc) #st dev intercept truncated

#st dev HAMA estimate truncated
hama.trunc <- numeric(n)
for(i in 1:n){
hama.trunc[i] <- as.numeric(out.sim[[i]][9])
}
(var(hama.trunc))^0.5
sd(hama.trunc) #st dev HAMA estimate truncated

#st dev sexmale estimate truncated
sex.trunc <- numeric(n)
for(i in 1:n){
sex.trunc[i] <- as.numeric(out.sim[[i]][17])
}
(var(sex.trunc))^0.5
sd(sex.trunc) #st dev sexmale estimate truncated

#assoct estimate truncated
assoct.trunc <- numeric(n)
for(i in 1:n){
assoct.trunc[i] <- as.numeric(out.sim[[i]][25])
}
(var(assoct.trunc))^0.5
sd(assoct.trunc) #assoct estimate truncated

#stdev int estimate truncated
sdint.trunc <- numeric(n)
for(i in 1:n){
sdint.trunc[i] <- as.numeric(out.sim[[i]][33])
}
(var(sdint.trunc))^0.5
sd(sdint.trunc) #assoct estimate truncated

#stdev res estimate truncated
```

```
sdres.trunc <- numeric(n)
for(i in 1:n){
sdres.trunc[i] <- as.numeric(out.sim[[i]][35])
}
(var(sdres.trunc))^0.5
sd(sdres.trunc) #assoct estimate truncated


############## biased

#intercept not truncated
int.bias <- numeric(n)
for(i in 1:n){
int.bias[i] <- as.numeric(out.sim[[i]][2])
}
(var(int.bias))^0.5
sd(int.bias) #intercept not truncated

#HAMA estimate not truncated
hama.bias <- numeric(n)
for(i in 1:n){
hama.bias[i] <- as.numeric(out.sim[[i]][10])
}
(var(hama.bias))^0.5
sd(hama.bias) #HAMA estimate not truncated

#sexmale estimate not truncated
sex.bias <- numeric(n)
for(i in 1:n){
sex.bias[i] <- as.numeric(out.sim[[i]][18])
}
var((sex.bias))^0.5
sd(sex.bias) #sexmale estimate not truncated


#assoct estimate not truncated
assoct.bias<- numeric(n)
for(i in 1:n){
assoct.bias[i] <- as.numeric(out.sim[[i]][26])
}
(var(assoct.bias))^0.5
sd(assoct.bias) #assoct estimate not truncated

#stdev int estimate truncated
sdint.bias <- numeric(n)
```

```
for(i in 1:n){
sdint.bias[i] <- as.numeric(out.sim[[i]][34])
}
(var(sdint.bias))^0.5
sd(sdint.bias) #assoct estimate truncated

#stdev res estimate truncated
sdres.bias <- numeric(n)
for(i in 1:n){
sdres.bias[i] <- as.numeric(out.sim[[i]][36])
}
(var(sdres.bias))^0.5
sd(sdres.bias) #assoct estimate truncated




stopCluster(cl)

############## Nelson - Aalen Plots ##################

wi=0 #in the case that sexn in sim_fun above is rep(0, i)
bi=0 #replace bi with 0 in sim_fun data generation function above
gamma<-estgamma

library(mice)

Ti <- data2[[1]]$eventtime
deltai=rep(1,length(Ti))
Tdata=data.frame(Ti, deltai)  #data frame of event times, event indicators
H1=nelsonaalen(Tdata,Ti, deltai) #calculate nelsonaalen estimator of H
plot(Ti,H1)
tt=seq(0,140,length=2800)
HTheor=lambda/(alpha*b1)*exp(gamma*wi+alpha*(b0+bi))*(exp(alpha*b1*tt)-1)
lines(tt, HTheor,col="red")


#simulation setting 2
#############################################################################
#Sys.setenv(JAGS_HOME="C:/Program Files/JAGS/JAGS-4.3.0")
library(tidyr)
library(tidyverse)
library(dplyr)
library(JMbayes)
library(survival)
```

```
#library(broom)
#require(caTools)

#iterations
n = 5000

#individuals
i = 251

#constant baseline hazard for exp dist assume
lambda = 0.005


#observation times
a = 18.30  #mean first visit date
c = 24.71 #mean last visit date
HAMA.time = rep(seq(a, c, by = 2), i)

#mean visits per subject before discarding individuals w/
#HAMA scores taken at/after event
v = 4

#log HAM-A scores
#real data
#real.hama <- lme(loghama ~ HAMAAGE_1,
#   random = ~ 1| ID,
#  data = HAMA, method = "REML")

#summary(real.hama)$coefficients[1]
b0 = 1.08455333 #b0
b1 = 0.05 #b1
sdp = 0.6290531 #subject specific raneff
sde = 0.6851948 #residual
pm = 0.455 #probability male

#################################################################################
#survival model with only sex as predictor
#real.surv <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + cluster(ID) ,
#      data=HAMA, x=TRUE, model=TRUE)
#summary(real.surv)

#JM for assoct, sex parameter estimate
#real.JM <- jointModelBayes(real.hama, real.surv, timeVar = "HAMAAGE_1")
#summary(real.JM)
alpha = 0.9642 #assoct
```

```
#regression estimate for time fixed parameter
estgamma = 0.2854

#turn it into a function

sim_fun = function(x) {
bi <- rep(rnorm(i, 0, sdp), rep(v, i))
me <- rnorm(v*i, 0, sde)
sexn <- rbinom(i,1,pm)
tempeventtime <- (1 / (b1 * alpha)) * log(1 - b1 * alpha *
log(runif(i, 0, 1)) / (lambda * exp(estgamma*sexn +
alpha * (b0 + unique(bi)))))
sexn <- rep(sexn, each=4)
eventtime <- rep(tempeventtime, each=4)
simdf <- data.frame(ID = sort(rep(c(1:i), v)), HAMA.time, bi, me, sexn, eventtime)
simdf$resp = with(simdf, b0 + b1*HAMA.time + bi + me )
simdf$sex <- factor(simdf$sexn,
levels=c(0,1),
labels=c("female","male"))
simdf$outcome = with(simdf, rep(1, rep(v*i))) #why do they all get event?

#if outcome happens, stop time = event time
#start times
simdf$start <- simdf$HAMA.time
splitID <- split(simdf[c("start", "HAMA.time")], simdf$ID)
#stop times
simdf$stop <- unlist(lapply(splitID, function(d) c(d$start[-1], d$HAMA.time[1])))

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

simdf
}

set.seed(99)
data1 = lapply(1:n, sim_fun)


#summary statistics event times before truncation
event.t <- data1[[1]][!duplicated(data1[[1]]$ID), ]$eventtime
ID <- data1[[1]][!duplicated(data1[[1]]$ID), ]$ID
for(i in 2:n){
event.t <- c(event.t, data1[[i]][!duplicated(data1[[i]]$ID), ]$eventtime)
```

```
ID <- c(ID, data1[[i]][!duplicated(data1[[i]]$ID), ]$ID)
}
summary(event.t)
length(ID) #full set

#number obs
nobs.full <- data1[[1]]$eventtime
for(i in 2:n){
nobs.full <- c(nobs.full, data1[[i]]$eventtime)
}
length(nobs.full)

sim_fun2 = function(data1){
#remove any observation times generated after the event time
data1 = filter(data1, data1$HAMA.time < data1$eventtime)

#for individuals who experienced onset
#replace last stop time with age at onset
data1 <- data1 %>%
group_by(ID) %>%
mutate(stop = ifelse(outcome == 1, c(stop[-n()], tail(eventtime,1)), stop)) %>%
as.data.frame()

#for individuals who experienced onset
#replace up to last indicator with 0

data1$event <- rep(0, nrow(data1))

temp <- data1 %>%
group_by(ID) %>%
mutate(event = ifelse(outcome == 1, c(event[-n()], tail(outcome,1)), event)) %>%
as.data.frame()

data1$outcome <- temp$event

data1
}

data2 <- lapply(data1, sim_fun2)

#number obs truncated
nobs.trunc <- data2[[1]]$eventtime
for(i in 2:n){
nobs.trunc <- c(nobs.trunc, data2[[i]]$eventtime)
}
```

```
length(nobs.trunc)


#Summary Statistics after truncation
event.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$eventtime
ID.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$ID
for(i in 2:n){
event.trunc <- c(event.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$eventtime)
ID.trunc <- c(ID.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$ID)
}
summary(event.trunc)
length(ID.trunc) #length ID.trunc

#proportion thrown out
(length(event.t) - length(event.trunc))/length(event.t)

#Individuals per dataset
(length(ID.trunc))/n

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

jm.fit = function(data2){


#linear model
sim.lm <- lme(resp ~ HAMA.time,
random = ~ 1| ID,
data = data2, method = "REML")

#incorrect dataset, ignoring truncation
data2.notrunc <- data2

#replace start value with 0
data2.id<- data2.notrunc %>%
group_by(ID) %>%
mutate(start = replace(start, 1, 0)) %>% as.data.frame()

#cox model w/ truncation
sim.surv <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2, x = TRUE, model = TRUE)

#cox model w/out truncation
sim.surv.nt <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2.id, x = TRUE, model = TRUE)
```

```
#joint model, accounting for truncation
sim.jm <- jointModelBayes(sim.lm, sim.surv, timeVar = "HAMA.time")
e1 <- summary(sim.jm)
#summary(sim.jm)

#joint model, ignoring truncation
sim.jm.nt <- jointModelBayes(sim.lm, sim.surv.nt, timeVar = "HAMA.time")
e2<- summary(sim.jm.nt)

estimates <- data.frame(e1$`CoefTable-Long`[1],  #intercept parameter truncated
e2$`CoefTable-Long`[1], #intercept parameter not truncated
e1$`CoefTable-Long`[3],  #intercept std error truncated
e2$`CoefTable-Long`[3], #intercept std error not truncated
e1$`CoefTable-Long`[5],  #intercept std dev truncated
e2$`CoefTable-Long`[5], #intercept std dev not truncated
e1$`CoefTable-Long`[11], #intercept p-value truncated
e2$`CoefTable-Long`[11], #intercept p-value not truncated
e1$`CoefTable-Long`[2], #hama.time parameter truncated
e2$`CoefTable-Long`[2], #hama.time parameter not truncated
e1$`CoefTable-Long`[4], #hama.time std error truncated
e2$`CoefTable-Long`[4], #hama.time std error not truncated
e1$`CoefTable-Long`[6], #hama.time std error truncated
e2$`CoefTable-Long`[6], #hama.time std error not truncated
e1$`CoefTable-Long`[12], #hama.time parameter p value truncated
e2$`CoefTable-Long`[12], #hama.time parameter p value not truncated
e1$`CoefTable-Event`[1], #sexmale parameter truncated
e2$`CoefTable-Event`[1], #sexmale parameter not truncated
e1$`CoefTable-Event`[4], #sexmale Std.Err truncated
e2$`CoefTable-Event`[4], #sexmale Std.Err not truncated
e1$`CoefTable-Event`[7], #sexmale Std.Dev truncated
e2$`CoefTable-Event`[7], #sexmale Std.Dev not truncated
e1$`CoefTable-Event`[16], #sexmale parameter p-value truncated
e2$`CoefTable-Event`[16], #sexmale parameter p-value not truncated
e1$`CoefTable-Event`[2], #assoct parameter truncated
e2$`CoefTable-Event`[2], #assoct parameter not truncated
e1$`CoefTable-Event`[5], #assoct Std.Err truncated
e2$`CoefTable-Event`[5], #assoct Std.Err not truncated
e1$`CoefTable-Event`[8], #assoct Std.Dev truncated
e2$`CoefTable-Event`[8], #assoct Std.Dev not truncated
e1$`CoefTable-Event`[17], #assoct p-value truncated
e2$`CoefTable-Event`[17], #assoct p-value not truncated
sqrt(e1$D), #stdev int truncated
sqrt(e2$D), #stdev int not truncated
e1$sigma, #stdev residuals truncated
e2$sigma, #stdev residuals not truncated
```

```
sim.jm$StDev$D, #stdev of D truncated
sim.jm.nt$StDev$D, #stdev of D not truncated
sim.jm$StDev$sigma, #stdev of sigma truncated
sim.jm.nt$StDev$sigma #stdev of sigma not truncated

)
estimates
}

#out.sim <- lapply(data2, jm.fit)
#JMmeans <- colMeans(do.call(rbind, out.sim))

############################## In Parallel #######################################

#detectCores()


nodeslist <- unlist(strsplit(Sys.getenv("NODESLIST"), split = ""))
z = as.numeric(Sys.getenv(("SLURM_ARRAY_TASK_ID")))

#cl = makeCluster(nodeslist, type = "PSOCK")
#registerDoParallel(cl)

library(parallel)
library(magrittr)
library(dplyr)

ncores=detectCores()
cl=makePSOCKcluster(ncores)

clusterExport(cl,
c("i", "v", "b0", "b1", "sdp",
"sde", "alpha", "lambda", "estgamma", "pm", "HAMA.time", "data1"))

clusterEvalQ(cl,
list(attach(loadNamespace("JMbayes"), name = "JMbayes"),
attach(loadNamespace("nlme"), name = "nlme"),
attach(loadNamespace("survival"), name = "survival"),
attach(loadNamespace("dplyr"), name = "dplyr"),
attach(loadNamespace("magrittr"), name = "magrittr"),
attach(loadNamespace("parallel"), name = "parallel")))

#set seed among clusters
#s <- set.seed(99)
#nextRNGStream(s)
```

```
clusterSetRNGStream(cl=cl, 99)

#run in parallel
out.sim <- parLapply(cl=cl,data2, jm.fit)


colMeans(do.call(rbind, out.sim))


#####################Empirical Standard Deviations############################

### truncated

#st dev intercept truncated
int.trunc <- numeric(n)
for(i in 1:n){
int.trunc[i] <- as.numeric(out.sim[[i]][1])
}
(var(int.trunc))^0.5
sd(int.trunc) #st dev intercept truncated

#st dev HAMA estimate truncated
hama.trunc <- numeric(n)
for(i in 1:n){
hama.trunc[i] <- as.numeric(out.sim[[i]][9])
}
(var(hama.trunc))^0.5
sd(hama.trunc) #st dev HAMA estimate truncated

#st dev sexmale estimate truncated
sex.trunc <- numeric(n)
for(i in 1:n){
sex.trunc[i] <- as.numeric(out.sim[[i]][17])
}
(var(sex.trunc))^0.5
sd(sex.trunc) #st dev sexmale estimate truncated

#assoct estimate truncated
assoct.trunc <- numeric(n)
for(i in 1:n){
assoct.trunc[i] <- as.numeric(out.sim[[i]][25])
}
(var(assoct.trunc))^0.5
sd(assoct.trunc) #assoct estimate truncated
```

```
#stdev int estimate truncated
sdint.trunc <- numeric(n)
for(i in 1:n){
sdint.trunc[i] <- as.numeric(out.sim[[i]][33])
}
(var(sdint.trunc))^0.5
sd(sdint.trunc) #assoct estimate truncated

#stdev res estimate truncated
sdres.trunc <- numeric(n)
for(i in 1:n){
sdres.trunc[i] <- as.numeric(out.sim[[i]][35])
}
(var(sdres.trunc))^0.5
sd(sdres.trunc) #assoct estimate truncated

############## biased

#intercept not truncated
int.bias <- numeric(n)
for(i in 1:n){
int.bias[i] <- as.numeric(out.sim[[i]][2])
}
(var(int.bias))^0.5
sd(int.bias) #intercept not truncated

#HAMA estimate not truncated
hama.bias <- numeric(n)
for(i in 1:n){
hama.bias[i] <- as.numeric(out.sim[[i]][10])
}
(var(hama.bias))^0.5
sd(hama.bias) #HAMA estimate not truncated

#sexmale estimate not truncated
sex.bias <- numeric(n)
for(i in 1:n){
sex.bias[i] <- as.numeric(out.sim[[i]][18])
}
var((sex.bias))^0.5
sd(sex.bias) #sexmale estimate not truncated


#assoct estimate not truncated
assoct.bias<- numeric(n)
```

```
for(i in 1:n){
assoct.bias[i] <- as.numeric(out.sim[[i]][26])
}
(var(assoct.bias))^0.5
sd(assoct.bias) #assoct estimate not truncated

#stdev int estimate truncated
sdint.bias <- numeric(n)
for(i in 1:n){
sdint.bias[i] <- as.numeric(out.sim[[i]][34])
}
(var(sdint.bias))^0.5
sd(sdint.bias) #assoct estimate truncated

#stdev res estimate truncated
sdres.bias <- numeric(n)
for(i in 1:n){
sdres.bias[i] <- as.numeric(out.sim[[i]][36])
}
(var(sdres.bias))^0.5
sd(sdres.bias) #assoct estimate truncated



stopCluster(cl)


#Simulation Setting 3
##############################################################################
#Sys.setenv(JAGS_HOME="C:/Program Files/JAGS/JAGS-4.3.0")
library(tidyr)
library(tidyverse)
library(dplyr)
library(JMbayes)
library(survival)
#library(broom)
#require(caTools)

#iterations
n = 5000

#individuals
i = 251

#constant baseline hazard for exp dist assume
```

```
lambda = 0.005


#observation times
a = 18.30  #mean first visit date
c = 24.71 #mean last visit date
HAMA.time = rep(seq(a, c, by = 2), i)

#mean visits per subject before discarding
 individuals
 #w/ HAMA scores taken at/after event
v = 4

#log HAM-A scores
#real data
#real.hama <- lme(loghama ~ HAMAAGE_1,
#   random = ~ 1| ID,
#  data = HAMA, method = "REML")

#summary(real.hama)$coefficients[1]
b0 = 1.08455333 #b0
b1 = 0.02331502 #b1
sdp = 0.6290531 #subject specific raneff
sde = 0.6851948 #residual
pm = 0.455 #probability male

##############################################################################
#survival model with only sex as predictor
#real.surv <- coxph(Surv(start, stop, OUTCOME1B) ~ SEX + cluster(ID) ,
#      data=HAMA, x=TRUE, model=TRUE)
#summary(real.surv)

#JM for assoct, sex parameter estimate
#real.JM <- jointModelBayes(real.hama, real.surv, timeVar = "HAMAAGE_1")
#summary(real.JM)
alpha = 0.9642 #assoct



#regression estimate for time fixed parameter
estgamma = 0.2854

#turn it into a function

sim_fun = function(x) {
```

```
bi <- rep(rnorm(i, 0, sdp), rep(v, i))
me <- rnorm(v*i, 0, sde)
sexn <- rbinom(i,1,pm)
tempeventtime <- (1 / (b1 * alpha)) *
log(1 - b1 * alpha * log(runif(i, 0, 1)) / (lambda * exp(estgamma*sexn + alpha *
(b0 + unique(bi)))))
sexn <- rep(sexn, each=4)
eventtime <- rep(tempeventtime, each=4)
ttrunctime <- rep(rlnorm(i, 2.8206476, 0.3894846))
trunctime <- rep(ttrunctime, each = 4)
simdf <- data.frame(ID = sort(rep(c(1:i), v)), HAMA.time, bi, me,
sexn, eventtime, trunctime)
simdf$sex <- factor(simdf$sexn,
levels=c(0,1),
labels=c("female","male"))
simdf$outcome = with(simdf, rep(1, rep(v*i)))

#if outcome happens, stop time = event time
#start times
simdf$start <- simdf$HAMA.time
splitID <- split(simdf[c("start", "HAMA.time")], simdf$ID)

#stop times
simdf$stop <- unlist(lapply(splitID, function(d) c(d$start[-1], d$HAMA.time[1])))

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

#remove any observation times generated before truncation time
simdf = filter(simdf, simdf$trunctime < simdf$HAMA.time)

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

simdf
}

set.seed(99)
data1 = lapply(1:n, sim_fun)


#summary statistics event times before truncation
event.t <- data1[[1]][!duplicated(data1[[1]]$ID), ]$eventtime
ID <- data1[[1]][!duplicated(data1[[1]]$ID), ]$ID
for(i in 2:n){
event.t <- c(event.t, data1[[i]][!duplicated(data1[[i]]$ID), ]$eventtime)
ID <- c(ID, data1[[i]][!duplicated(data1[[i]]$ID), ]$ID)
```

```
}
summary(event.t)
length(ID) #full set

sim_fun2 = function(data1){

#length(unique(simdf$ID)) #after truncation

#replace first start time in each ID with truncation time
data1 <- data1 %>% group_by(ID) %>%
mutate(start = replace(start, 1, trunctime))%>% as.data.frame()

#for individuals who experienced onset
#replace last stop time with age at onset
data1 <- data1 %>%
group_by(ID) %>%
mutate(stop = ifelse(outcome == 1, c(stop[-n()], tail(eventtime,1)), stop)) %>%
as.data.frame()

#for individuals who experienced onset
#replace up to last indicator with 0

data1$event <- rep(0, nrow(data1))

temp <- data1 %>%
group_by(ID) %>%
mutate(event = ifelse(outcome == 1, c(event[-n()], tail(outcome,1)), event)) %>%
as.data.frame()

data1$outcome <- temp$event

#remove any observation times generated after the event time
data1 = filter(data1, data1$start < data1$eventtime)

#generate response time with start time
data1$resp = with(data1, b0 + b1*start + bi + me )

data1$HAMA.time <- data1$start

data1
}

data2 <- lapply(data1, sim_fun2)

#Summary Statistics event times after truncation
```

```
event.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$eventtime
ID.trunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$ID
for(i in 2:n){
event.trunc <- c(event.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$eventtime)
ID.trunc <- c(ID.trunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$ID)
}
summary(event.trunc)
length(ID.trunc) #length ID.trunc

#Summary Statistics truncation times after truncation
event.ttrunc <- data2[[1]][!duplicated(data2[[1]]$ID), ]$trunctime
for(i in 2:n){
event.ttrunc <- c(event.ttrunc, data2[[i]][!duplicated(data2[[i]]$ID), ]$trunctime)
}
summary(event.ttrunc)



#proportion thrown out
(length(event.t) - length(event.trunc))/length(event.t)

#Individuals per dataset
(length(ID.trunc))/n

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

jm.fit = function(data2){


#linear model
sim.lm <- lme(resp ~ HAMA.time,
random = ~ 1| ID,
data = data2, method = "REML")

#incorrect dataset, ignoring truncation
data2.notrunc <- data2

#replace start value with 0
data2.id<- data2.notrunc %>%
group_by(ID) %>%
mutate(start = replace(start, 1, 0)) %>% as.data.frame()

#cox model w/ truncation
sim.surv <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2, x = TRUE, model = TRUE)
```

```
#cox model w/out truncation
sim.surv.nt <- coxph(Surv(start, stop, outcome) ~ sex + cluster(ID) ,
data=data2.id, x = TRUE, model = TRUE)

#joint model, accounting for truncation
sim.jm <- jointModelBayes(sim.lm, sim.surv, timeVar = "HAMA.time")
e1 <- summary(sim.jm)
#summary(sim.jm)

#joint model, ignoring truncation
sim.jm.nt <- jointModelBayes(sim.lm, sim.surv.nt, timeVar = "HAMA.time")
e2<- summary(sim.jm.nt)

estimates <- data.frame(e1$`CoefTable-Long`[1],  #intercept parameter truncated
e2$`CoefTable-Long`[1], #intercept parameter not truncated
e1$`CoefTable-Long`[3],  #intercept std error truncated
e2$`CoefTable-Long`[3], #intercept std error not truncated
e1$`CoefTable-Long`[5],  #intercept std dev truncated
e2$`CoefTable-Long`[5], #intercept std dev not truncated
e1$`CoefTable-Long`[11], #intercept p-value truncated
e2$`CoefTable-Long`[11], #intercept p-value not truncated
e1$`CoefTable-Long`[2], #hama.time parameter truncated
e2$`CoefTable-Long`[2], #hama.time parameter not truncated
e1$`CoefTable-Long`[4], #hama.time std error truncated
e2$`CoefTable-Long`[4], #hama.time std error not truncated
e1$`CoefTable-Long`[6], #hama.time std error truncated
e2$`CoefTable-Long`[6], #hama.time std error not truncated
e1$`CoefTable-Long`[12], #hama.time parameter p value truncated
e2$`CoefTable-Long`[12], #hama.time parameter p value not truncated
e1$`CoefTable-Event`[1], #sexmale parameter truncated
e2$`CoefTable-Event`[1], #sexmale parameter not truncated
e1$`CoefTable-Event`[4], #sexmale Std.Err truncated
e2$`CoefTable-Event`[4], #sexmale Std.Err not truncated
e1$`CoefTable-Event`[7], #sexmale Std.Dev truncated
e2$`CoefTable-Event`[7], #sexmale Std.Dev not truncated
e1$`CoefTable-Event`[16], #sexmale parameter p-value truncated
e2$`CoefTable-Event`[16], #sexmale parameter p-value not truncated
e1$`CoefTable-Event`[2], #assoct parameter truncated
e2$`CoefTable-Event`[2], #assoct parameter not truncated
e1$`CoefTable-Event`[5], #assoct Std.Err truncated
e2$`CoefTable-Event`[5], #assoct Std.Err not truncated
e1$`CoefTable-Event`[8], #assoct Std.Dev truncated
e2$`CoefTable-Event`[8], #assoct Std.Dev not truncated
e1$`CoefTable-Event`[17], #assoct p-value truncated
```

```
e2$`CoefTable-Event`[17], #assoct p-value not truncated
sqrt(e1$D), #stdev int truncated
sqrt(e2$D), #stdev int not truncated
e1$sigma, #stdev residuals truncated
e2$sigma, #stdev residuals not truncated
sim.jm$StDev$D, #stdev of D truncated
sim.jm.nt$StDev$D, #stdev of D not truncated
sim.jm$StDev$sigma, #stdev of sigma truncated
sim.jm.nt$StDev$sigma #stdev of sigma not truncated

)
estimates
}

#out.sim <- lapply(data2, jm.fit)
#JMmeans <- colMeans(do.call(rbind, out.sim))

############################## In Parallel #######################################

#detectCores()


nodeslist <- unlist(strsplit(Sys.getenv("NODESLIST"), split = ""))
z = as.numeric(Sys.getenv(("SLURM_ARRAY_TASK_ID")))

#cl = makeCluster(nodeslist, type = "PSOCK")
#registerDoParallel(cl)

library(parallel)
library(magrittr)
library(dplyr)

ncores=detectCores()
cl=makePSOCKcluster(ncores)

clusterExport(cl,
c("i", "v", "b0", "b1", "sdp", "sde", "alpha", "lambda",
 "estgamma", "pm", "HAMA.time", "data1"))

clusterEvalQ(cl,
list(attach(loadNamespace("JMbayes"), name = "JMbayes"),
attach(loadNamespace("nlme"), name = "nlme"),
attach(loadNamespace("survival"), name = "survival"),
attach(loadNamespace("dplyr"), name = "dplyr"),
attach(loadNamespace("magrittr"), name = "magrittr"),
```

```
attach(loadNamespace("parallel"), name = "parallel")))

#set seed among clusters
#s <- set.seed(99)
#nextRNGStream(s)
clusterSetRNGStream(cl=cl, 99)

#run in parallel
out.sim <- parLapply(cl=cl,data2, jm.fit)


colMeans(do.call(rbind, out.sim))


#####################Empirical Standard Deviations############################

### truncated

#st dev intercept truncated
int.trunc <- numeric(n)
for(i in 1:n){
int.trunc[i] <- as.numeric(out.sim[[i]][1])
}
(var(int.trunc))^0.5
sd(int.trunc) #st dev intercept truncated

#st dev HAMA estimate truncated
hama.trunc <- numeric(n)
for(i in 1:n){
hama.trunc[i] <- as.numeric(out.sim[[i]][9])
}
(var(hama.trunc))^0.5
sd(hama.trunc) #st dev HAMA estimate truncated

#st dev sexmale estimate truncated
sex.trunc <- numeric(n)
for(i in 1:n){
sex.trunc[i] <- as.numeric(out.sim[[i]][17])
}
(var(sex.trunc))^0.5
sd(sex.trunc) #st dev sexmale estimate truncated

#assoct estimate truncated
assoct.trunc <- numeric(n)
for(i in 1:n){
```

```
assoct.trunc[i] <- as.numeric(out.sim[[i]][25])
}
(var(assoct.trunc))^0.5
sd(assoct.trunc) #assoct estimate truncated

#stdev int estimate truncated
sdint.trunc <- numeric(n)
for(i in 1:n){
sdint.trunc[i] <- as.numeric(out.sim[[i]][33])
}
(var(sdint.trunc))^0.5
sd(sdint.trunc) #assoct estimate truncated

#stdev res estimate truncated
sdres.trunc <- numeric(n)
for(i in 1:n){
sdres.trunc[i] <- as.numeric(out.sim[[i]][35])
}
(var(sdres.trunc))^0.5
sd(sdres.trunc) #assoct estimate truncated


############## biased

#intercept not truncated
int.bias <- numeric(n)
for(i in 1:n){
int.bias[i] <- as.numeric(out.sim[[i]][2])
}
(var(int.bias))^0.5
sd(int.bias) #intercept not truncated

#HAMA estimate not truncated
hama.bias <- numeric(n)
for(i in 1:n){
hama.bias[i] <- as.numeric(out.sim[[i]][10])
}
(var(hama.bias))^0.5
sd(hama.bias) #HAMA estimate not truncated

#sexmale estimate not truncated
sex.bias <- numeric(n)
for(i in 1:n){
sex.bias[i] <- as.numeric(out.sim[[i]][18])
}
```

```
var((sex.bias))^0.5
sd(sex.bias) #sexmale estimate not truncated


#assoct estimate not truncated
assoct.bias<- numeric(n)
for(i in 1:n){
assoct.bias[i] <- as.numeric(out.sim[[i]][26])
}
(var(assoct.bias))^0.5
sd(assoct.bias) #assoct estimate not truncated

#stdev int estimate truncated
sdint.bias <- numeric(n)
for(i in 1:n){
sdint.bias[i] <- as.numeric(out.sim[[i]][34])
}
(var(sdint.bias))^0.5
sd(sdint.bias) #assoct estimate truncated

#stdev res estimate truncated
sdres.bias <- numeric(n)
for(i in 1:n){
sdres.bias[i] <- as.numeric(out.sim[[i]][36])
}
(var(sdres.bias))^0.5
sd(sdres.bias) #assoct estimate truncated




stopCluster(cl)
```

A.4 Bash Submission Script

```
#!/bin/bash
#SBATCH --account=def-jhorrock
#SBATCH --mem-per-cpu=10GB
#SBATCH --cpus-per-task=32
#SBATCH --time=3:00:00
#SBATCH --output=indtrunc_5000_3h.out
#SBATCH --mail-user=mcgiverl@uoguelph.ca
```

```
#SBATCH --mail-type=END
#SBATCH --mail-type=FAIL

module load nixpkgs/16.09 gcc/7.3.0 r/4.0.2 gdal/3.0.1 proj/6.0.0
module load jags/4.3.0
export R_LIBS=~/R/x86_64-pc-linux-gnu-library/4.0/
export NODESLIST=$(echo $(srun hostname | cut -f 1 -d '.'))

Rscript ind_trunc_5000.R
```