Marginal Approaches for Joint Models with Clustered Data

by
Danielle Gaudet

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

MARGINAL APPROACHES FOR
JOINT MODELS WITH CLUSTERED DATA

Danielle Gaudet
University of Guelph, 2021

Advisor(s):
Dr. Julie Horrocks
Dr. Gerarda Darlington

In the presence of clustering of individuals, standard errors (SEs) of model parameter estimates may be poorly estimated when models that assume independence of subjects are used. Marginal approaches in survival models aim to correct this, usually by estimating the model parameters under the assumption that all subjects are independent and then using a method that recognizes intra-cluster correlation to estimate the SEs. This thesis assesses the marginal approach with the application of group jackknife and group bootstrap to estimate SEs of parameter estimates in joint models with clustered data and compares them to model-based SEs that do not account for clustering. These methods were compared using a real data set. A simulation study compared the three methods of estimating the SEs against the empirical standard deviation (SD) of the parameter estimates. The results of the simulation study indicated that the group jackknife SEs and group bootstrap SEs were much closer to the empirical SDs than the model-based SEs from the misspecified joint model.

Acknowledgements

I would like to thank my advisors Dr. Julie Horrocks and Dr. Gerarda Darlington for continuing to provide constructive criticism and pushing me to improve throughout this process. I truly appreciate you dedicating your time and patience to helping me each week.

Next, I would like to thank the graduate program assistant, Susan McCormick, for being so kind in handling all the administrative work and sending reminders of important deadlines. I would also like to thank Dr. Zeny Feng, the external examiner, for taking the time out of her busy schedule to read and provide feedback on my work.

Finally, I would like to thank Matthew Lowe for providing resources and advice concerning Sharcnet and Megan French for providing a lot of support, encouragement, and advice throughout the whole process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Time-to-event data with repeated measures of a time-varying covariate arises in many study fields, for example, medicine, sociology, and epidemiology. Oftentimes, research subjects may be genetically linked, treated by the same physician, or taught in the same classroom. It is reasonable to assume that these clustered individuals might be more similar than those from different clusters. There is a need to accommodate clustering among outcomes of study subjects in statistical analyses. In particular, there exists a need to incorporate this intra-cluster correlation within the joint modelling framework.

Joint models couple the longitudinal mixed effects model for repeated measurements data with a model for time-to-event data (Rizopoulos, 2012). The longitudinal and survival sub-models are linked together through a common term (Wulfsohn & Tsiatis, 1997; Rizopoulos, 2012). This allows for inferences of the association between the two sub-models to be made especially when endogenous covariates are considered (Ibrahim et al., 2010; Rizopoulos, 2012). An endogenous covariate is usually the result of a stochastic process within the study subject and is often only observable as long as the study subject survives and is uncensored (Kalbfleisch & Prentice, 2002; Rizopoulos, 2012). As such, the covariate may contain information about the survival status of the subject (Kalbfleisch & Prentice, 2002; Rizopoulos, 2012). In contrast, an exogeneous (or external) covariate may take one of several forms: the covariate is measured prior to the study and fixed throughout (e.g. treatment/placebo), the variable path is defined so that it is known to any time t (e.g. age throughout a study), or the covariate is ancillary (Kalbfleisch & Prentice, 2002). An ancillary covariate is a stochastic process external to the study subject and is not associated with the other parameters involved in the study (e.g. the weather or ambient temperature in a lab) (Kalbfleisch & Prentice, 2002). One motivation for the joint-modelling framework is the ability to model associations

between an endogenous time-varying covariate and time-to-event data (Rizopoulos, 2012).

Model misspecification can lead to inconsistent parameter and standard error estimates (White, 1982). Using statistical models that assume independence in the presence of correlation is an example of model misspecification. Therefore, any correlation between individuals should be accounted for when fitting joint models. One method is to fit the data using a marginal approach. In survival analysis, the marginal approach is to first fit a model to estimate model parameters under the assumption that event times are independent (Xiao & Abrahamowicz, 2010). Then, a method that recognizes the cluster-level correlation is applied to estimate the SEs (Xiao & Abrahamowicz, 2010).

Marginal approaches have previously been applied to the Cox survival model in literature. Xiao and Abrahamowicz (2010) showed that the group bootstrap (sampling entire clusters with replacement) is a preferable method to the two-stage bootstrap (sampling the clusters with replacement and then randomly sampling the observations within the cluster) for estimating SEs in Cox survival models. Lipsitz et al. (1994) applied a marginal model with SEs estimated using the standard jackknife to clustered event time data and showed that it is robust to model misspecification. However, Lipsitz et al. (1994) suggest that the group jackknife may be computationally simpler when there are many clusters. Additionally, the group jackknife may preserve the correlation structure of the clusters (Ukoumunne et al., 2003).

There exist many packages to fit joint models in the statistical software `R` (R Core Team, 2021) including `JM` (Rizopoulos, 2010), `joineR` (Philipson et al., 2018) and `JMBayes` (Rizopoulos, 2016). However, to my knowledge, none are designed to account for clustering of individuals. Since SE under-estimation may result in an inflated type 1 error, it is important that SEs are estimated appropriately (Xiao & Abrahamowicz, 2010). Therefore, the aim of this thesis is to extend the method proposed by (Lipsitz et al., 1994) to joint models. A marginal approach with SEs estimated by the group jackknife and group bootstrap are proposed and evaluated with respect to estimation of the SEs of the model parameter estimates for joint models with clustered data. Joint models will be fit using the `jointModel` function in the `JM` (Rizopoulos, 2010) package and then, to account for clustering, the SEs will be estimated using the group jackknife (Busing et al., 1999; Efron, 1979) and the group bootstrap (Efron & Stein, 1981; Sherman & le Cessie, 1997).

A general background of longitudinal analysis, survival analysis and joint models, as well as an introduction to the jackknife and bootstrap variance estimation techniques are explained in Chapter 2. Chapter 3 describes the application of the jackknife and bootstrap

estimates of variance for clustered joint models to data from a study of bipolar disorder. Chapter 4 outlines the methodology, modelling and results from simulated data. Finally, conclusions and suggestions for future research are discussed in Chapter 5.

Chapter 2

Background

This chapter introduces the basics of longitudinal analyses, survival models, and joint models. The group jackknife (Efron, 1979), and the group bootstrap (Davison & Hinkley, 1999; Field & Welsh, 2007) will be described. All notation used in this chapter and throughout the thesis can be referred to in Table D.1 of Appendix D.

2.1 Introduction to Longitudinal Analysis

Oftentimes in research it is important to measure a variable of interest over time. Consider studies over time of changes in human microbiome (Luna et al., 2020), periodontal disease status over time (Wang et al., 2011), and quality of life changes in cancer patients (Anota et al., 2014), all of which require repeated measurements at various time points t. In these examples, there are some elements that can be thought of as averaged over the population (i.e. fixed effects) and other effects that can be treated as unique to each study subject (i.e. random effects) (Rizopoulos, 2012).

A useful approach for modelling both fixed and random effects is the linear mixed effects (LME) model. The mean response is modeled as a function of population characteristics shared by all subjects and the effects specific to each subject are accounted for (Fitzmaurice et al., 2004). Consider a study with G clusters with n_g individuals in the g^{th} cluster. There are n_{gi} repeated measurements for the i^{th} individual within cluster g where $i = 1, 2, \ldots, n_g$ and $g = 1, 2, \ldots, G$. Let p represent the number of fixed effects and q represent the number of random effects. Using vector and matrix notation similar to Rizopoulos (2012), the general LME model for the response of interest, the $n_{gi} \times 1$ vector $\boldsymbol{y}_{gi}$ for subject i in cluster g, can be expressed as

$$\begin{cases} \boldsymbol{y}_{gi} &= \mathbf{X}_{gi}\boldsymbol{\beta} + \mathbf{Z}_{gi}\boldsymbol{b}_{gi} + \boldsymbol{\varepsilon}_{gi} \\ \boldsymbol{b}_{gi} &\sim \mathrm{N}(0, \mathbf{D}) \\ \boldsymbol{\varepsilon}_{gi} &\sim \mathrm{N}(0, \sigma^2\mathbf{I}_{n_{gi}}) \end{cases} \tag{2.1}$$

where $\mathbf{X}_{gi}$ and $\mathbf{Z}_{gi}$ are known design matrices of size $n_{gi} \times p$ and $n_{gi} \times q$, respectively. The $n_{gi} \times n_{gi}$ identity matrix is denoted by $\mathbf{I}_{n_{gi}}$. The fixed-effects are represented by a $p \times 1$ vector $\boldsymbol{\beta}$ of regression parameters and the random-effects $\boldsymbol{b_{gi}}$ are represented by a $q \times 1$ vector (Rizopoulos, 2012). The random effects are assumed to be normally distributed with mean 0 and $q \times q$ variance-covariance matrix $\mathbf{D}$ (Rizopoulos, 2012). The $n_{gi} \times 1$ vector of errors $\boldsymbol{\varepsilon}_{gi}$ is assumed to be normally distributed with mean 0 and variance-covariance matrix $\sigma^2\mathbf{I}_{n_{gi}}$ and assumed to be independent of the random effects (Rizopoulos, 2012).

LME models not only provide easily interpretable estimates of the regression coefficients, but are also able to predict how individual response trajectories change over time (Rizopoulos, 2012). This feature is especially advantageous for joint modelling.

In order to estimate the parameters, maximum likelihood (ML) estimation methods are applied. The marginal density of the observed response $\boldsymbol{y_{gi}}$ for subject i in cluster g is given by

$$p(\boldsymbol{y}_{gi}) = \int p(\boldsymbol{y}_{gi} \mid \boldsymbol{b}_{gi}) p(\boldsymbol{b}_{gi}) d\boldsymbol{b}_{gi} \tag{2.2}$$

where $p(\cdot)$ represents a probability density function (Rizopoulos, 2012). Given the normality assumptions of $\boldsymbol{b}_{gi}$ and $\boldsymbol{\varepsilon}_{gi}$ detailed in Equation 2.1, the integral in Equation 2.2 has a closed-form solution leading to a multivariate normal distribution with mean $\boldsymbol{X_{gi}\beta}$ and variance-covariance matrix $\boldsymbol{V_{gi} = Z_{gi} D Z_{gi}^\top} + \sigma^2 \boldsymbol{I}_{n_{gi}}$ (Rizopoulos, 2012), such that,

$$p(\boldsymbol{y_{gi}}) = (2\pi)^{-n_{gi}/2} |\boldsymbol{V}_{gi}|^{-1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{y}_{gi} - \boldsymbol{X_{gi}\beta})^\top \boldsymbol{V}_{gi}^{-1} (\boldsymbol{y}_{gi} - \boldsymbol{X_{gi}\beta}) \right\}.$$

Then, the log-likelihood is

$$\ell(\boldsymbol{\theta}) = \sum_{g=1}^{G} \sum_{i=1}^{n_g} \log \int p(\boldsymbol{y_{gi}} \mid \boldsymbol{b_{gi}}; \boldsymbol{\beta}, \sigma^2)\; p(\boldsymbol{b_{gi}}; \boldsymbol{\theta_b})\; d\boldsymbol{b_{gi}} \tag{2.3}$$

where $\boldsymbol{\theta}^\top = (\boldsymbol{\beta}^\top, \sigma^2, \boldsymbol{\theta_b}^\top)^\top$ and $\boldsymbol{\theta_b}$ is the vectorization of $\mathbf{D}$ (Rizopoulos, 2012). When $\boldsymbol{V_{gi}}$ is known, then the maximization of Equation 2.3, conditional on the parameters in $\boldsymbol{V_{gi}}$, results in the closed form for the fixed effects regression coefficient generalized least squares

estimator:

$$\hat{\boldsymbol{\beta}} = \left(\sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{\boldsymbol{gi}}^{\top} \boldsymbol{V}_{\boldsymbol{gi}}^{-\mathbf{1}} \boldsymbol{X}_{\boldsymbol{gi}} \right)^{-1} \sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{\boldsymbol{gi}}^{\top} \boldsymbol{V}_{\boldsymbol{gi}}^{-\mathbf{1}} \boldsymbol{y}_{\boldsymbol{gi}} \tag{2.4}$$

(Rizopoulos, 2012). Further, the SE for the fixed effects regression coefficients can be directly obtained by calculating the estimated variance-covariance matrix of the least squares estimator as shown by Rizopoulos (2012),

$$\hat{\mathbf{var}}(\hat{\boldsymbol{\beta}}) = \left(\sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{\boldsymbol{gi}}^{\top} \hat{\boldsymbol{V}}_{gi}^{-1} \boldsymbol{X}_{gi} \right)^{-1} \left(\sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{gi}^{\top} \hat{\boldsymbol{V}}_{gi}^{-1} \hat{\mathbf{var}}(\boldsymbol{y}_{gi}) \hat{\boldsymbol{V}}_{gi}^{-1} \boldsymbol{X}_{gi} \right) \times \left(\sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{gi}^{\top} \hat{\boldsymbol{V}}_{gi}^{-1} \boldsymbol{X}_{gi} \right)^{-1} \tag{2.5}$$

where $\hat{\mathrm{SE}}(\hat{\beta}_k) = \sqrt{\hat{\mathrm{var}}(\hat{\beta}_k)}$, $k = 0, 1, \ldots, p-1$. The variance-covariance matrix estimate from Equation 2.5 can be simplified when the model is specified correctly, that is, $\hat{\mathbf{var}}(\boldsymbol{y}_{\boldsymbol{gi}}) = \hat{\boldsymbol{V}}_{gi}$ (Rizopoulos, 2012). Then, the estimate in Equation 2.5 can be expressed as,

$$\hat{\mathbf{var}}(\hat{\boldsymbol{\beta}}) = \left(\sum_{g=1}^{G} \sum_{i=1}^{n_g} \boldsymbol{X}_{gi}^{\top} \hat{\boldsymbol{V}}_{gi}^{-1} \boldsymbol{X}_{gi} \right)^{-1}. \tag{2.6}$$

In instances where the model is misspecified, the sandwich estimator proposed by White (1982) can be applied to make the SE estimates more robust. This is obtained by setting $\hat{\mathbf{var}}(\boldsymbol{y}_{\boldsymbol{gi}})$ in Equation 2.5 to $(\boldsymbol{y}_{\boldsymbol{gi}} - \boldsymbol{X}_{\boldsymbol{gi}}\hat{\boldsymbol{\beta}})(\boldsymbol{y}_{\boldsymbol{gi}} - \boldsymbol{X}_{\boldsymbol{gi}}\hat{\boldsymbol{\beta}})^{\top}$ (White, 1982; Rizopoulos, 2012).

2.2 Introduction to Survival Analysis

The main goal of survival analysis is to fit a model of the time until some defined event, such as death or failure. Applicable to many fields, such as medicine (e.g. tumor development), demography (e.g. divorce) or econometrics (e.g. loan defaults), the analysis of survival data is widely used as a technique to assess associations between covariates and the time to an event of interest. Unlike many other statistical methods that rely on the assumption of normality, survival analysis does not assume normality as times to an event are generally skewed (Rizopoulos, 2012). Further, survival data usually involves censoring (Rizopoulos, 2012). For the purposes of this thesis, only right censoring will be considered. Right censoring

occurs when a subject leaves the study prematurely or is lost to follow-up or when the study ends prior to the subject experiencing the event of interest (Aalen et al., 2008).

Let T^*_{gi} denote the random variable representing the time to the event of interest and C_{gi} denote the random variable representing the time to a censoring event such that what is observed is $T_{gi} = \min(T^*_{gi}, C_{gi})$. Then the censoring indicator δ_{gi} for the i^{th} subject in cluster g is

$$\delta_{gi} = \begin{cases} 0, & T^*_{gi} > T_{gi} \\ 1, & T^*_{gi} = T_{gi} \end{cases} \tag{2.7}$$

such that 0 indicates the subject is right censored and 1 indicates the subject's event time was observed (Rizopoulos, 2012; Moore, 2016). To express the probability that the event of interest has not yet happened at time t, the survival function is defined as

$$S(t) = \mathrm{P}(T^* > t) = \int_t^\infty p(s)\, ds \tag{2.8}$$

where $p(\cdot)$ represents the probability density function (Rizopoulos, 2012; Aalen et al., 2008). The survival function must be non-increasing as t increases (Aalen et al., 2008; Rizopoulos, 2012). Assuming that T^* is continuous with some probability density $p(\cdot)$, then the hazard function is defined as

$$h(t) = \lim_{dt \to 0} \frac{\mathrm{P}(t \leq T^* < t + dt \mid T^* \geq t)}{dt}, \quad t > 0 \tag{2.9}$$

(Aalen et al., 2008; Rizopoulos, 2012). The hazard function describes the instantaneous rate or risk of experiencing the event within the small time interval $[t, t + dt)$ (Aalen et al., 2008; Rizopoulos, 2012). The survival function and the hazard function are related through the cumulative hazard function, $H(t)$ (Aalen et al., 2008), that is the accumulated risk up until time t,

$$S(t) = \exp\{-H(t)\} = \exp\left\{-\int_0^t h(s)\, ds\right\}. \tag{2.10}$$

In order to model the effect of covariates on survival, the Cox regression model is often fit (Cox, 1972). Then, the hazard rate for an individual i in cluster g with covariates $\boldsymbol{w}_{gi} = (w_{gi1}, w_{gi2}, \ldots, w_{gih})$ is

$$h(t|\boldsymbol{w_{gi}}) = h_0(t) \exp\{\boldsymbol{w}_{gi}^\top \boldsymbol{\gamma}\} \tag{2.11}$$

where $h_0(t)$ represents the baseline hazard and $\boldsymbol{\gamma}$ denotes the $h \times 1$ vector of regression coefficients corresponding to $\boldsymbol{w}_{gi}$ (Cox, 1972). The hazard ratio is the ratio between the hazard of two subjects, such that the hazard ratio between subject i with covariate vector $\boldsymbol{w_{gi}}$ and subject j with covariate vector $\boldsymbol{w_{gj}}$ is expressed as $\exp\{\boldsymbol{\gamma}^\top(\boldsymbol{w_{gi}} - \boldsymbol{w_{gj}}\}$. Then, $\exp\{\gamma_h\}$ denotes the hazard ratio for a one-unit change in covariate w_{gih} at any time t (Rizopoulos, 2012). Cox (1972) showed that $\boldsymbol{\gamma}$ can be estimated by maximizing the partial log-likelihood ℓ with no need to make any assumptions about the baseline hazard $h_0(\cdot)$,

$$\ell(\boldsymbol{\gamma}) = \sum_{g=1}^{G} \sum_{i=1}^{n_g} \delta_{gi} \left[\boldsymbol{w}_{gi}^\top \boldsymbol{\gamma} - \log \left\{ \sum_{T_j \geq T_i} \exp\{\boldsymbol{w}_{gi}^\top \boldsymbol{\gamma}\} \right\} \right]. \tag{2.12}$$

The model in Equation 2.11 can be further extended to accommodate exogenous time-dependent covariates. This modification is referred to as the extended Cox model (or the Andersen-Gill model) (Andersen & Gill, 1982; Rizopoulos, 2012). The extended Cox model requires treating the events as a slow Poisson process, also known as a counting process (Rizopoulos, 2012). Let $N_{gi}(t)$ represent the number of events for subject i in cluster g at time t and $R_{gi}(t)$ represent a left continuous at-risk process with $R_{gi}(t) = 1$ if the subject i in cluster g is at risk at time t, $R_{gi}(t) = 0$ otherwise (Rizopoulos, 2012). Then the event process for a subject i in cluster g is given by $\{N_{gi}(t), R_{gi}(t)\}$. This leads to the modified version of Equation 2.11:

$$h_{gi}(t \mid \mathcal{Y}_{gi}(t), \boldsymbol{w}_{gi}) = h_0(t) R_{gi}(t) \exp\{\boldsymbol{\gamma}^\top \boldsymbol{w}_{gi} + \alpha y_{gi}(t)\} \tag{2.13}$$

where $y_{gi}(t)$ denotes the time-dependent covariate at time t for individual i in cluster g with regression coefficient α and $\mathcal{Y}_{gi}(t)$ denotes the covariate history up to time t (Rizopoulos, 2012). The hazard ratio is now represented by $\exp\{\gamma^\top \boldsymbol{w}_{gi} + \alpha y_{gi}(t)\}$ and is now time-dependent (Rizopoulos, 2012).

Unfortunately, the extended Cox model is not able to handle endogenous covariates since this model requires the time-dependent covariates to be measured without error, predictable, and have a fully specified path (Rizopoulos, 2012). Therefore, more sophisticated statistical methods, such as the joint model, should be employed.

2.3 Introduction to the Joint Model

The joint model is a common alternative when the extended Cox model is not appropriate as is the case when handling endogenous variables (Rizopoulos, 2012). Joint models are comprised of both longitudinal and survival sub-models, linked together by a common term (Wulfsohn & Tsiatis, 1997). Joint models have applications in fields such as medicine. For example, in the identification of optimal drug dose (Altzerinakou & Paoletti, 2021) or assessing predictors for progression of Alzheimer's disease (Li & Luo, 2017).

2.3.1 Longitudinal Sub-model

The longitudinal sub-model allows for modeling the history of the covariate of interest (Rizopoulos, 2012). The longitudinal covariate is measured intermittently and with error at time t_{gij} where i represents the individual ($i = 1, 2, \ldots, n_g$) within cluster g ($g = 1, 2, \ldots, G$) and j represents the time point ($j = 1, 2, \ldots, n_{gi}$).

Let the observed longitudinal covariate be denoted as $y_{gi}(t)$, and the smoothed covariate be denoted as $m_{gi}(t)$ (Rizopoulos, 2012). In order to produce estimates of the smoothed covariate's longitudinal history, $M_{gi}(t) = \{m_{gi}(s),\ 0 \leq s < t\}$, the LME model is applied, denoted by

$$\begin{cases} y_{gi}(t) & = & m_{gi}(t) + \varepsilon_{gi}(t) \\ m_{gi}(t) & = & \boldsymbol{x}_{\boldsymbol{gi}}^{\top}(t)\boldsymbol{\beta} + \boldsymbol{z}_{\boldsymbol{gi}}^{\top}(t)\boldsymbol{b}_{gi} \\ \boldsymbol{b}_{gi} & \sim & \mathrm{N}(0, \mathbf{D}) \\ \varepsilon_{gi}(t) & \sim & \mathrm{N}(0, \sigma^2) \end{cases} \tag{2.14}$$

where, at a fixed time t, $\boldsymbol{x_{gi}}(t)$ and $\boldsymbol{z_{gi}}(t)$ are design vectors for the fixed effects and random effects of size $p \times 1$ and $q \times 1$, respectively (Rizopoulos, 2012). The $q \times 1$ $\boldsymbol{b}_{gi}$ vector is assumed to be normally distributed with mean 0 and variance-covariance matrix $\boldsymbol{D}$ and the errors $\varepsilon_{gi}(t)$ are assumed to be normally distributed with mean 0 and variance σ^2 (Rizopoulos, 2012).

The LME model produces estimates of the smoothed covariate $m_{gi}(t)$, which are applied to the survival sub-model to estimate $S_{gi}(t)$ (Rizopoulos, 2012). It is used as it is able to account for any measurement error surrounding $m_{gi}(t)$.

2.3.2 Survival Sub-model

To assess the association, α, between the smoothed covariate, $m_i(t)$, and the risk for an event, we consider the relative risk model of the form:

$$h_{gi}(t|M_{gi}(t), \boldsymbol{w_{gi}}) = h_0(t) \exp\left\{\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}} + \alpha m_{gi}(t)\right\}, \quad t > 0, \tag{2.15}$$

where $M_{gi}(t) = \{m_{gi}(s), 0 \leq s < t\}$ denotes the smoothed longitudinal covariate history up to time point t, $h_0(\cdot)$ denotes the baseline hazard function, and $\boldsymbol{w_{gi}}$ is a $(h \times 1)$ vector of h baseline covariates with the corresponding $(h \times 1)$ vector of regression coefficients $\boldsymbol{\gamma}$ (Rizopoulos, 2012). Similarly, α represents the regression coefficient of the smoothed unobserved longitudinal covariate $m_{gi}(t)$ (Rizopoulos, 2012). The hazard function depends on time t only through $m_{gi}(t)$; however this is not the same for the survival function (Rizopoulos, 2012). Using the relationship between the survival function and the cumulative hazard function, then the survival function is:

$$S_{gi}(t \mid M_{gi}(t), \boldsymbol{w_{gi}}) = \exp\left(-\int_0^t h_0(s) \exp\left\{\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}} + \alpha m_{gi}(s)\right\} ds\right)$$

(Rizopoulos, 2012). It can be seen that the survival function depends on the entire history of the longitudinal covariate from 0 to time t, $M_{gi}(t)$ (Rizopoulos, 2012).

In the Cox model, an assumption for the functional form of the baseline hazard, $h_0(\cdot)$, is not required. However, within the joint modelling framework, leaving $h_0(\cdot)$ unspecified may lead to the underestimation the SEs of the parameter estimates (Hsieh et al., 2006; Rizopoulos, 2012). Generally, flexible functions such as the fully parametric Weibull or gamma hazard functions or a piecewise-constant function are used as the baseline hazard function (Papageorgiou et al., 2019; Rizopoulos, 2012).

This thesis will specify the baseline hazard as the Weibull proportional hazards (PH) function. The Weibull baseline hazard has been found to be a good descriptor of many types of survival data and is also quite flexible (Lawless, 2003). The Weibull-PH hazard function takes the form

$$h_0(t) = \lambda \eta t^{\eta - 1} \tag{2.16}$$

where $\eta > 0$ and $\lambda > 0$ are the shape and scale parameters, respectively (Rizopoulos, 2012; Lawless, 2003). The `JM` package incorporates an intercept term in the $\boldsymbol{\gamma}$ regression coefficient term (see Equation 2.15) (Rizopoulos, 2010). To avoid identifiability problems between the

intercept term γ_0 and the shape parameter η, the `JM` package sets λ in Equation 2.16 to 1 (Rizopoulos, 2012; Kalbfleisch & Prentice, 2002). In the special case where $\eta = 1$, the hazard function is the exponential distribution (Rizopoulos, 2010). When $\eta > 1$, then the Weibull-PH hazard function is monotone increasing and monotone decreasing when $\eta < 1$.

2.3.3 The Joint Likelihood

The joint model estimates the parameters of the longitudinal and survival sub-models jointly, instead of separately (Hsieh et al., 2006). For the purposes of this thesis, the `R` package `JM` will be used to provide model parameter estimates. This package estimates the parameters using the maximum likelihood (ML) approach (Rizopoulos, 2012).

The ML estimates are derived from the log-likelihood function corresponding to the joint distribution of the observed outcomes $\{T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}\}$. Assume that the vector of time-independent random effects $\boldsymbol{b_{gi}}$ underlies both the longitudinal and survival processes. This implies that the random effects account for both the association between the longitudinal and survival outcomes, as well as the correlation between the repeated measurements (Rizopoulos, 2012). Assuming that $\{T_{gi}, \delta_{gi}\}$ and $\boldsymbol{y_{gi}}$ are independent given the random effects, the probability density of the joint distribution given the random effects is

$$\begin{aligned} p(T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}} \mid \boldsymbol{b_{gi}}) &= p(T_{gi}, \delta_{gi} \mid \boldsymbol{b_{gi}})\, p(\boldsymbol{y_{gi}} \mid \boldsymbol{b_{gi}}), \quad \text{and} \\ p(\boldsymbol{y_{gi}} \mid \boldsymbol{b_{gi}}) &= \prod_j p\left\{\boldsymbol{y_{gij}} \mid \boldsymbol{b_{gi}}\right)\} \end{aligned}$$

(Rizopoulos, 2012). We also assume that, given the observed longitudinal history, the censoring mechanism and the process at which the time points for observations are taken are independent of the true event times and future longitudinal measurements (Rizopoulos, 2012). Under these assumptions, the log-likelihood for the i^{th} subject in cluster g is

$$\begin{aligned} \log p(T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}) &= \log \int p(T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}, \boldsymbol{b_{gi}})\, d\boldsymbol{b_{gi}} \\ &= \log \int p(T_{gi}, \delta_{gi} \mid \boldsymbol{b_{gi}}) \left[\prod_j p\left\{\boldsymbol{y_{gij}} \mid \boldsymbol{b_{gi}}\right)\}\right] p(\boldsymbol{b_{gi}})\, d\boldsymbol{b_{gi}} \end{aligned} \tag{2.17}$$

(Rizopoulos, 2012). In this thesis, a relative risk survival sub-model used will be of the form

$$\begin{aligned} p(T_{gi}, \delta_{gi} \mid \boldsymbol{b_{gi}}) &= h_i(T_{gi})^{\delta_{gi}} S_{gi}(T_{gi}) \\ &= \left[h_0(T_{gi}) \exp\left\{\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}} + \alpha m_{gi}(T_{gi})\right\}\right]^{\delta_{gi}} \\ &\quad \times \exp\left(-\int_0^{T_{gi}} h_0(s) \exp\left\{\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}} + \alpha m_{gi}(s)\right\} d\boldsymbol{s}\right) \end{aligned} \tag{2.18}$$

where $h_0(\cdot)$ is the baseline hazard function (Rizopoulos, 2012; Wulfsohn & Tsiatis, 1997). Assuming normality for both the longitudinal covariate and the random effects gives the joint density for the longitudinal process with the random effects as

$$\begin{aligned} p(\boldsymbol{y_{gi}} \mid \boldsymbol{b_{gi}})\, p(\boldsymbol{b_{gi}}) &= \prod_j p\left\{\boldsymbol{y_{gij}} \mid \boldsymbol{b_{gi}}\right\}\ p(\boldsymbol{b_{gi}}) \\ &= \frac{1}{(2\pi\sigma^2)^{n_{gi}/2}} \exp\left[-\frac{\| \boldsymbol{y_{gi}} - \boldsymbol{X_{gi}}\boldsymbol{\beta} - \boldsymbol{Z_{gi}}\boldsymbol{b_{gi}} \|^2}{2\sigma^2}\right] \\ &\quad \times \frac{1}{(2\pi)^{q/2}} \det(\boldsymbol{D})^{-1/2} \exp(-\boldsymbol{b_{gi}^\top}\boldsymbol{D^{-1}}\boldsymbol{b_{gi}}/2) \end{aligned} \tag{2.19}$$

where q denotes the dimensionality of the random-effects vector $\boldsymbol{b}_{gi}$ and where $\|x\| = \{\sum_i x_i^2\}^{1/2}$ denotes the Euclidean vector norm (Rizopoulos, 2012; Tsiatis & Davidian, 2004).

The maximization of the log-likelihood $\ell(\boldsymbol{\theta}) = \sum_g \sum_i \log p(T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be done using standard algorithms such as the Expectation-Maximization (EM) (Dempster et al., 1977) or the Newton-Raphson algorithms (Lange, 2013; Rizopoulos, 2012; Wulfsohn & Tsiatis, 1997). Specifically, the `JM` package (Rizopoulos, 2010) starts with the EM algorithm, treating the random effects as missing data, for a fixed number of iterations and in the case of non-convergence, switches to quasi-Newton methods until convergence (Rizopoulos, 2010).

A source of difficulty for fitting joint models is the integral with respect to time in the survival function and the integral with respect to the random effects in the longitudinal function. To alleviate the computational burden, the package `JM` applies the standard or adaptive Gauss-Hermite (GH) rule to approximate the integral over the random effects and Gauss-Kronrod (GK) points to approximate the integral in the survival function (Rizopoulos, 2010). This thesis will use the standard GH rule with the default 15 GH-quadrature points and the default 15 Gauss-Kronrod (GK) points, exclusively.

The score vector can be defined as the first derivative of the log-likelihood, $\mathcal{S}(\boldsymbol{\theta}) = \frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$.

Then, the score vector can be written as

$$\mathcal{S}(\boldsymbol{\theta}) = \sum_g \sum_i \int A(\boldsymbol{\theta}, \boldsymbol{b_{gi}})\, p(\boldsymbol{b_{gi}} \mid T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}; \boldsymbol{\theta})\, d\boldsymbol{b_{gi}} \tag{2.20}$$

where $A(\cdot)$ denotes the complete data score vector, given by $A(\boldsymbol{\theta}, \boldsymbol{b_{gi}}) = \partial\{\log g(T_{gi}, \delta_{gi}|\boldsymbol{b_{gi}}; \boldsymbol{\theta}) + \log p(\boldsymbol{y_{gi}}|\boldsymbol{b_{gi}}; \boldsymbol{\theta}) + \log p(\boldsymbol{b_{gi}}; \boldsymbol{\theta})\}/\partial\boldsymbol{\theta}^\top$ (Rizopoulos, 2012). If the score equations (Equation 2.20) are solved with respect to $\boldsymbol{\theta}$, where $p(\boldsymbol{b_{gi}} \mid T_{gi}, \delta_{gi}, \boldsymbol{y_{gi}}; \boldsymbol{\theta})$ is fixed at the $\boldsymbol{\theta}$ value of the previous iteration, then this represents an EM algorithm (Rizopoulos, 2012). For the complete outline of the EM algorithm for joint models see Rizopoulos (2012, Appendix B).

The JM package calculates the Hessian using only the function that computes the score vector using a numerical derivative routine, for example the forward or central difference approximation. This thesis uses the default forward difference approximation. The estimated observed information matrix, $\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}})$, is the negative of the inverse Hessian matrix and is used to get an estimate of the standard errors:

$$\hat{\mathbf{var}}(\hat{\boldsymbol{\theta}}) = \left\{\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}})\right\}^{-1}, \quad \text{with } \boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}}) = -\left.\frac{\partial \boldsymbol{\mathcal{S}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\theta=\hat{\theta}}$$

(Rizopoulos, 2012). The SEs of the parameter estimates can be found as the square-root of the diagonal elements of $\hat{\mathbf{var}}(\hat{\boldsymbol{\theta}})$. When the baseline hazard is left unspecified, the estimation of the SE becomes complicated and may be underestimated (Hsieh et al., 2006; Papageorgiou et al., 2019). In this thesis, the baseline hazard will be specified as the Weibull proportional hazards function.

2.4 Introduction to Resampling Methods

The `R` package `JM` estimates the joint model parameters and their associated standard errors (SEs). Since the `jointModel` function in `JM` (Rizopoulos, 2010) treats all subjects as independent, the SEs are not adjusted for clustering. In order to adjust for intra-cluster correlation, the jackknife and the bootstrap will be used to estimate the SEs. The basic background of these techniques is outlined in this section.

The key feature of clustered data is that observations within a cluster are correlated (e.g. family members, patients treated at the same hospital) (Galbraith et al., 2010). In classical statistics, subjects are often treated as independent (Galbraith et al., 2010). In instances where intra-cluster correlation is present and the response is continuous, but clas-

sical statistical methods are used, parameter estimates will be consistent (Diggle et al., 2013; Fitzmaurice et al., 2004). However standard errors have been found to be underestimated (Hsieh et al., 2006; Sherman & le Cessie, 1997).

Resampling methods, such as the jackknife and bootstrap, are able to estimate the properties of parameter estimates such as SE and bias (Efron & Tibshirani, 1993). However, these methods also require independence among sampled units (Efron & Tibshirani, 1993). With clustered data, it is not reasonable to assume that subjects are independent within a cluster (Galbraith et al., 2010; Sherman & le Cessie, 1997). As such, it is inappropriate to resample individual observations to estimate SE. In addition, the individual-level jackknife and bootstrap do not preserve the correlation structure within a cluster (Du & Lee, 2019; Ukoumunne et al., 2003). A grouped resampling method that samples independent clusters rather than individuals can be used to preserve the correlation structure (Sherman & le Cessie, 1997). Overall, there are two strategies: (1) resample clusters with replacement (one stage) or (2) resample clusters and then resample individuals within the selected clusters (two-stage) (Davison & Hinkley, 1999; Ukoumunne et al., 2003). Davison & Hinkley (1999) have shown that the one-stage (i.e. group) resampling method is preferable (Davison & Hinkley, 1999). Additionally, resampling imposes the assumption that all sampled units are independent. Since the two-stage method involves resampling within a cluster, the independence assumption is violated. Therefore, only the one-stage grouped resampling method will be considered in this thesis.

Since clusters may vary in size, the jackknife samples may also vary in size. The same can be said for the bootstrap. In order to adjust for this, the bootstrap and jackknife estimators will be weighted for sample size (Busing et al., 1999; Sherman & le Cessie, 1997).

2.4.1 Jackknife

Suppose we have a random sample of size n and we calculate the estimator $\hat{\theta}$. The i^{th} jackknife sample can be used to estimate the SE of $\hat{\theta}$. A jackknife sample takes the form,

$$\boldsymbol{x}_{(-i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1} \ldots, x_n)$$

where $(-i)$ indicates that individual i is left out of the sample, $i = 1, 2, \ldots, n$ (Efron & Tibshirani, 1993). Therefore, there are a total of n jackknife samples, all of size $n - 1$. For each jackknife sample $\boldsymbol{x}_{(-i)}$, calculate the estimator of interest and denote it as $\hat{\theta}_{(-i)}$. Then

the jackknife estimate of standard error is defined by

$$\hat{\text{SE}}_{Jack}(\hat{\theta}) = \left[\frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}_{(-i)} - \hat{\theta}_{(\cdot)} \right)^2 \right]^{1/2} \tag{2.21}$$

where $\hat{\theta}_{(\cdot)} = \sum_{i=1}^{n} \hat{\theta}_{(-i)}/n$ (Efron & Tibshirani, 1993).

This technique can be extended to data where there are multiple clusters. Here we outline the group jackknife (Du & Lee, 2019). Using the same notation as previously established, suppose we have G clusters where we assume that there is intra-cluster correlation and let y_{gij} be the response for individual i ($i = 1, 2, \ldots, n_g$) in cluster g ($g = 1, 2, \ldots, G$) at time point j ($j = 1, 2, \ldots, n_{gi}$). Instead of deleting one observation, leave out an entire cluster to preserve the correlation structure within the cluster (Du & Lee, 2019). The estimator of interest is calculated based on each sample $\boldsymbol{x}_{(-g)}$ where the subscript $(-g)$ denotes that cluster g is being left out. The parameter estimate is calculated based on the g^{th} sample is denoted as $\hat{\theta}_{(-g)}$.

In the case where each cluster g is of equal size, then the SE computation for the group jackknife is analogous to the regular jackknife (see Equation 2.21), replacing n with the number of clusters G,

$$\hat{\text{SE}}_{EJK}(\hat{\theta}) = \left[\frac{G-1}{G} \sum_{g=1}^{G} \left(\hat{\theta}_{(-g)} - \hat{\theta}_{(\cdot)} \right)^2 \right]^{1/2} \tag{2.22}$$

where $\hat{\theta}_{(\cdot)} = \sum_{g=1}^{G} \hat{\theta}_{(-g)}/G$ (Busing et al., 1999). Hereafter, this will be referred to as the equal jackknife (EJK).

However, it cannot always be assumed that cluster sizes are equal. A method proposed by Busing et al. (1999) suggests weighting by the proportion of subjects in cluster g. Let $u_g = \frac{n_g}{n}$ be the proportion of subjects in cluster g. The SE estimator for the $\hat{\theta}$ is based on the pseudo-values,

$$\tilde{\theta}_{(-g)} = \frac{1}{u_g} \hat{\theta}_n - \left(\frac{1}{u_g} - 1 \right) \hat{\theta}_{(-g)} \tag{2.23}$$

where $\hat{\theta}_n$ is the estimate of θ based on the full sample (Busing et al., 1999). Then the SE

computation for the jackknife with unequal cluster sizes is

$$\hat{\text{SE}}_{UJK}(\hat{\theta}) = \left[\frac{1}{G}\sum_{g=1}^{G}\frac{u_g}{1-u_g}\left(\tilde{\theta}_{(-g)} - \bar{\theta}_{(\cdot)}\right)^2\right]^{1/2} \tag{2.24}$$

where $\bar{\theta}_{(\cdot)}$ is the weighted average of the G estimators $\hat{\theta}_{(-g)}$ and is equal to $G\hat{\theta}_n - \sum_{g=1}^{G}(1-u_g)\hat{\theta}_{(-g)}$ (Busing et al., 1999). This will be referred to as the unequal jackknife (UJK).

2.4.2 Bootstrap

The bootstrap is similar to the jackknife. In fact, the jackknife is shown to be a linear approximation of the bootstrap (Efron, 1979). However, the bootstrap samples are drawn from the original sample with replacement and the samples are of the same size (i.e. n) (Efron, 1979).

Suppose we have a random sample $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ of size n and we calculate some estimator $\hat{\theta}$. To estimate the SE of $\hat{\theta}$, B independent bootstrap samples are required. Let the bootstrap samples be denoted as $\boldsymbol{x^{(b)}}$ for $b = 1, 2, \ldots, B$. The B bootstrap samples are drawn with replacement from the original sample $\boldsymbol{x}$ and are usually of the same size n. The estimator calculated on the b^{th} bootstrap sample is denoted $\hat{\theta}^{(b)}$. These bootstrap estimates can then be used to estimate the SE of $\hat{\theta}$:

$$\hat{\text{SE}}_{Boot}(\hat{\theta}) = \left[\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^{(b)} - \hat{\theta}^{(\cdot)}\right)^2\right]^{1/2} \tag{2.25}$$

where $\hat{\theta}^{(\cdot)} = \sum_{b=1}^{B}\hat{\theta}^{(b)}/B$ (Efron & Tibshirani, 1993).

Much like the individual-level jackknife, the individual-level bootstrap does not preserve the correlation structure within clusters (Ukoumunne et al., 2003). The bootstrap can also be extended to accommodate correlated outcomes in the form of multiple clusters. Again, we will consider only the group bootstrap (Ukoumunne et al., 2003). Suppose we have n observations arranged in G clusters where we assume that there is intra-cluster correlation. Instead of sampling individuals, G clusters are randomly sampled in their entirety with replacement to preserve the correlation structure within the cluster (Ukoumunne et al., 2003). In the case where each cluster is of equal size, then $n = Gk$ where k is the number of individuals in each cluster. Then, the SE computation for the group bootstrap is the same as that for independent data (see Equation 2.25) (Xiao & Abrahamowicz, 2010).

If clusters are not all of equal size, each bootstrap sample may vary in size. Therefore, the bootstrap estimates can be weighted based on bootstrap sample size (Efron & Tibshirani, 1993; Sherman & le Cessie, 1997). The weighted bootstrap estimator from the b^{th} sample is $\hat{\theta}_W^{(b)} = \left(\frac{n_b^*}{n}\right)^{1/2} \hat{\theta}^{(b)}$ where n_b^* is the size of the b^{th} bootstrap sample and n is the original sample size (Sherman & le Cessie, 1997). Then the weighted estimate of the bootstrap standard error is

$$\hat{\mathrm{SE}}_{Boot}^{W}(\hat{\theta}) = \left[\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_W^{(b)} - \hat{\theta}_W^{(\cdot)}\right)^2\right]^{1/2}$$

where $\hat{\theta}_W^{(\cdot)} = \sum_{i=1}^{B} \hat{\theta}_W^{(b)}/B$.

Chapter 3

Data Analysis

This chapter will discuss the application of the marginal approaches to fit joint models in the presence of clusters. Models are first fit under the assumption that all subjects are independent and then the group jackknife and group bootstrap are used to estimate the model parameter estimates' SEs. The first data set is concerned with the onset of bipolar disorder (BD) and the second data set concerns the survival of female baboons. All source code for these analyses can be found in Appendices C.1 and C.2.

3.1 Bipolar Data Set

The bipolar data set is collected as part of an ongoing prospective study where offspring from a parent diagnosed with bipolar disorder (BD) are assessed and followed over time. Note that each offspring has exactly one parent affected with BD. A description of the data collection can be found in Duffy et al. (2014). A key feature of this data set is that some of the study participants are identified as siblings or cousins. Therefore, clustering based on families is present.

The data set contains measured covariates and the event or censoring times from 305 individuals. The baseline characteristics of all individuals measured upon enrollment in the study are sex, socioeconomic status of affected parent as determined by the Hollingshead scale (Hollingshead, 1957), lithium response of the affected parent, and the parental age of onset of BD. Participants were intermittently assessed using the Hamilton anxiety (HAM-A) scale (Hamilton, 1959). The HAM-A is designed to assess the severity of anxiety symptoms and is scored on a scale of 0 to 56 (Hamilton, 1959). The HAM-A scores serve as the time-varying covariate of interest for this analysis.

The time to event outcome of interest is the age at which participants are diagnosed with either BD (Type I, II, or not-otherwise specified (NOS)), major depressive disorder (MDD) or schizoaffective disorder (SchD) (depressive or bipolar type). Participants were considered diagnosed with BD/MDD/SchD when they first met the criteria for full Diagnostic and Statistical Manual of Mental Disorders (DSM) diagnosis.

3.1.1 Exploratory Data Analysis

The original data comprises 305 patients from 121 different families. Of the 305 individuals, only 207 met the criterion of having at least one HAM-A value prior to diagnosis of BD or the censoring time. After dropping the individuals not meeting the criterion, a total of 98 families remained from the previous 121.

Since genetic similarity may contribute to correlated outcomes for diseases that are highly heritable such as BD (Wilde et al., 2014), the participants are clustered based on family. A histogram of the cluster sizes can be seen in Figure 3.1 for the 98 clusters used in this thesis. The cluster sizes range from 1 to 9, with singletons being the most common. Table 3.1 further outlines the summary statistics of the cluster sizes.

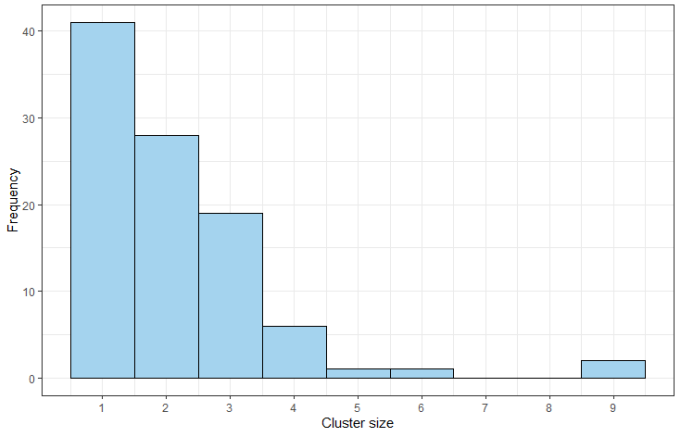

Figure 3.1: Frequency of cluster sizes ($G = 98$).

Table 3.1: Summary statistics of the clusters ($G = 98$). SD = standard deviation.

	Mean (SD)	Median	Minimum	Maximum
Cluster size	2.11 (1.46)	2.00	1.00	9.00

Of the 207 participants, there are more females (54.6%) than males (45.4%). At the

time of enrollment in the study, most of the affected parents of the study subjects are classed within the highest socioeconomic status (SES) according to the Hollingshead scale (Hollingshead, 1957). The SES values of 1, 2 and 3 were merged together due to the low frequency of subjects within those categories. Looking at the event outcome, there are more participants who did not (90.8%) get diagnosed with BD/MDD/SchD than those that did (9.2%). Table 3.2 provides a summary of these baseline and event outcome characteristics and Table 3.3 describes the summary statistics of the continuous measurements.

Table 3.2: Summary statistics of baseline and response variables of bipolar study subjects ($n = 207$). The socioeconomic status (SES) is measured by the Hollingshead scale and values of 1, 2, and 3 were grouped.

	Frequency (%)
Event outcome (BD/MDD/SchD diagnosis)	
No	188 (90.8)
Yes	19 (9.2)
Sex	
Male	94 (45.4)
Female	113 (54.6)
Hollingshead scale for SES of affected parent	
1, 2, 3	30 (14.5)
4	81 (39.1)
5	96 (46.4)
Lithium response of affected parent	
Negative (LiNR/LiNR-profile)	121 (58.5)
Positive (LiR/LiR-profile)	86 (41.5)

Table 3.3: Summary statistics for continuous measurements for the bipolar data set ($n = 207$). SD = standard deviation.

	Mean (SD)	Median	Minimum	Maximum
Parental age of onset	26.07 (9.97)	24.19	5.12	49.63
Age at first interview	16.34 (6.78)	14.98	3.27	39.71
Age at last interview	24.02 (7.89)	24.51	7.24	45.81

The HAM-A scores of the subjects were taken intermittently over the course of the study and the number of measured scores varies from person to person. Since the HAM-A scores

will be fit in an LME model as part of the joint model, the normality of the residuals must be checked. The normality of the model residuals with HAM-A as the response was compared to that of the transformed HAM-A where the natural logarithm transformation was applied. The HAM-A scores can possibly range from 0 to 56, therefore 1 point was added to all measurements prior to taking the log. It can be seen in Figure 3.2 that the transformed HAM-A scores result in a more normal distribution of the residuals. To visualize the relationship between the HAM-A scores over the course of the subjects lives, the HAM-A scores were plotted against age of the subject (see Figure 3.3).

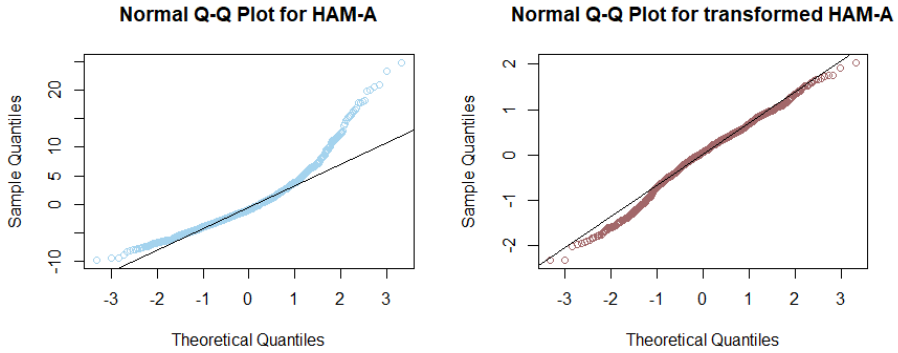

Figure 3.2: Normal Q-Q plots for LME models with HAM-A (A) and transformed HAM-A (B) as the response to check the normality of the errors assumption. Sex, lithium response, parental age of onset, and SES status are accounted for.

As shown in Table 3.4, the mean number of HAM-A scores taken per participant is 2.75. The measured scores range from 0 to 37, with the average being 5.93. The overall distribution of the HAM-A and the transformed HAM-A scores can be seen in Figure 3.4.

Table 3.4: Summary statistics of HAM-A and transformed HAM-A scores ($n = 207$). SD = standard deviation.

	Mean (SD)	Median	Minimum	Maximum
HAM-A scores	5.93 (5.74)	4.00	0.00	37.00
Transformed HAM-A scores	1.59 (0.88)	1.61	0.00	3.64
Number of HAM-A scores per subject	2.75 (2.31)	2.00	1.00	12.00

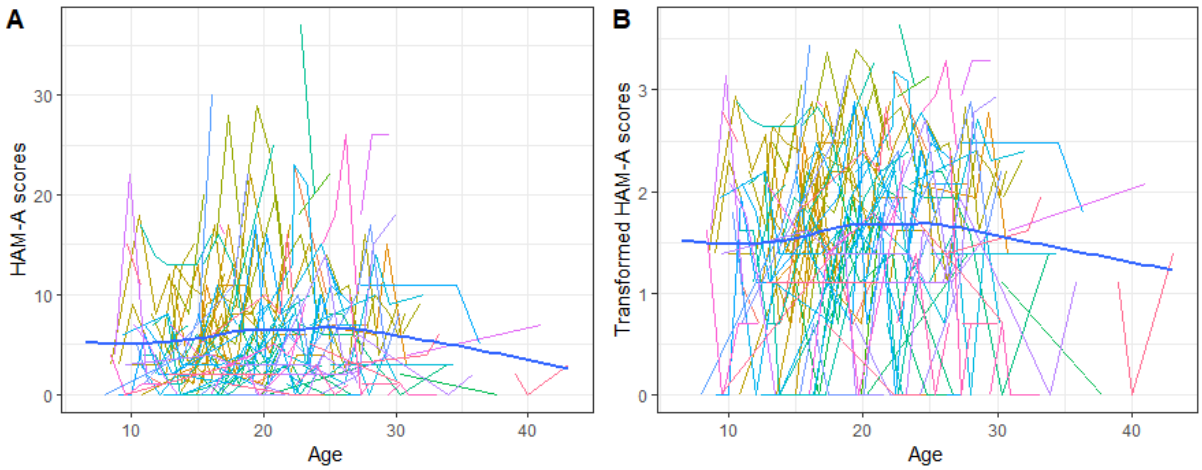

Figure 3.3: (A) Individual HAM-A scores plotted over time ($n = 207$). (B) Individual transformed HAM-A scores plotted over time ($n = 207$). The thicker blue line represents a locally estimated scatterplot smoother.

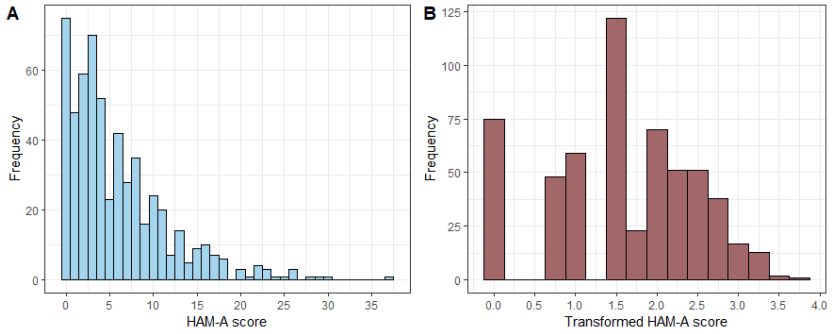

Figure 3.4: (A) Frequency of HAM-A scores from the 207 individuals. (B) Frequency of transformed HAM-A scores.

3.1.2 Data Set Analysis and Discussion

As previously discussed, in many statistical models parameter estimates are consistent regardless of correlation structure, but the SE estimates may be incorrect (Diggle et al., 2013). This section outlines the fitted joint model as well as the jackknife and bootstrap samples to estimate SE. Since the purpose of this investigation is to assess methods for estimating SE for use in marginal approaches, only simple models are considered. Therefore, a simple joint model was fit using the transformed HAM-A as the longitudinal covariate. Further, SEs were estimated using jackknife and bootstrap. All joint models are fit with the `JM` package (Rizopoulos, 2010) in `R` (R Core Team, 2021) as well as using the Shared Hierarchi-

cal Academic Research Computing Network (SHARCNET: www.sharcnet.ca) supported by Compute/Calcul Canada (www.computecanada.ca).

The time-independent variables used as part of the simple model are defined as

$$\text{Sex} = \begin{cases} 1, & \text{Female} \\ 0, & \text{Male} \end{cases},$$

$$\text{SES}_4 = \begin{cases} 1, & \text{SES score of 4} \\ 0, & \text{otherwise} \end{cases},$$

$$\text{SES}_{123} = \begin{cases} 1, & \text{SES score of 1,2, or 3} \\ 0, & \text{otherwise} \end{cases}.$$

The longitudinal sub-model follows a random intercept mixed-effects model. So that the transformed HAM-A score is modelled by,

$$m_{gi}(t_{gij}) = (\beta_0 + b_{0gi}) + \beta_1 t_{gij} + \beta_2 \text{Sex}_{gi} \tag{3.1}$$

where $m_{gi}(t)$ is the expected value of the transformed HAM-A scores for individual i in cluster g at time t, and t_{gij} is the time at which the HAM-A score is assessed for individual i in cluster g at timepoint j ($g = 1, 2, \ldots, 98; i = 1, 2, \ldots, n_g; j = 1, 2, \ldots, n_{gi}$). The random intercept term is denoted by b_{0gi}. The survival sub-model is of the form

$$h_{gi}(t_{gij}) = h_0(t_{gij}) \exp\left\{\gamma_1 \text{SES}_{4gi} + \gamma_2 \text{SES}_{123gi} + \alpha m_{gi}(t_{gij})\right\} \tag{3.2}$$

where $h_0(t)$ represents the Weibull-PH baseline hazard.

The results of the fitted joint model are summarized in Table 3.5. The longitudinal sub-model indicates that the transformed HAM-A scores increase by 0.023 (95% CI: (0.009, 0.036)) units for a one year increase in age at assessment given that sex remains constant. Meanwhile, there is weak evidence that the transformed HAM-A scores of female participants are different than the male participants at a given time t.

The intercept term in the survival sub-model corresponds to the log of the baseline hazard scale parameter λ from Equation 2.16 and the log(Shape) corresponds to the Weibull-PH η. The association parameter represents the relationship between the survival time and the smoothed longitudinal transformed HAM-A scores, $m_{gi}(t)$. There is strong evidence to suggest the log-hazard ratio of the smoothed transformed HAM-A scores is not 0. The log-hazard

Table 3.5: Summary of the estimated model parameters from the fitted bipolar joint model. The SE estimates do not account for clustering within the data.

Sub-model	Variable	Coefficient	Standard error	95% CI	p-value
Longitudinal	Intercept	1.009	0.166	(0.684, 1.333)	< 0.001
	Sex	0.201	0.111	(-0.017, 0.418)	0.070
	Age	0.023	0.007	(0.009, 0.036)	0.001
Survival	Intercept	-15.024	2.412	(-19.751, -10.296)	< 0.001
	SES_4	0.054	0.579	(-1.081, 1.189)	0.925
	SES_{123}	0.901	0.586	(-0.247, 2.048)	0.124
	Association	1.224	0.565	(0.118, 2.330)	0.030
	log(Shape)	1.154	0.197		< 0.001
Variance	σ	0.683	0.038		
	D	0.354	0.181		

ratio of the transformed HAM-A score is 1.224 (95% CI: (0.118, 2.330)). Further, there is no evidence that SES_4 (95% CI: $(-1.081, 1.189)$) and SES_{123} (95% CI: $(-0.247, 2.048)$) has an effect on the time until onset of BD/MDD/SchD relative to SES_5.

Table 3.6: Comparison of the bipolar data set joint model SE estimates. There were 98 jackknife samples and 200 bootstrap resamples. Sixteen of the bootstrap resamples resulted in non-convergent models and were removed and replaced with other bootstrap samples. The unadjusted SE estimates from the joint model with no adjustments for clustering, equal jackknife (EJK), unequal jackknife (UJK), bootstrap (BS) and the weighted bootstrap (WBS) estimates of the model parameter estimates' SEs are included.

			Estimated SE				
Sub-model	Variable	Coefficient	Unadjusted	EJK	UJK	BS	WBS
Longitudinal	Intercept	1.009	0.166	0.202	0.186	0.199	0.196
	Sex	0.201	0.111	0.122	0.122	0.111	0.111
	Age	0.023	0.007	0.010	0.009	0.010	0.010
Survival	Intercept	-15.024	2.412	2.201	2.171	2.314	2.462
	SES_4	0.054	0.579	0.664	0.575	0.671	0.671
	SES_{123}	0.901	0.586	1.083	0.894	0.799	0.802
	Association	1.224	0.516	0.744	0.773	0.729	0.728
Variance	σ	0.683	0.038	0.032	0.028	0.032	0.043
	D	0.354	0.181	0.072	0.070	0.071	0.070

The SEs are estimated using the group jackknife and the group bootstrap (see Table

3.6), resulting in 98 jackknife samples and 200 bootstrap resamples. The EJK was included for comparison. However, since the clusters from the bipolar data set are not of equal size, the UJK is a more appropriate estimator. Similarly, both the weighted and unweighted bootstrap estimates were calculated. All of the jackknife samples provided convergent joint models. However, of the 200 bootstrap samples, 16 resulted in a non-convergent joint model. Therefore, the 16 bootstrap samples were removed and replaced with other bootstrap samples.

The UJK SE estimates are generally higher than the estimates unadjusted for clustering, except for the survival sub-model intercept SE, SES_4 and σ which showed a 10%, 0.7%, and 26.3% decrease, respectively. Additionally, the UJK SE estimates for D are 61.3% lower than the unadjusted SE. The EJK estimates display a similar pattern to that of the UJK estimates, however the EJK estimates are higher than those from the UJK. The BS and WBS SEs provide comparable estimates to the EJK and UJK. The BS and WBS SE estimates are also greater than the unadjusted SE except for the SE of D, which are approximately 60% less than the unadjusted SE. Notably, the jackknife and bootstrap SE estimates for D are much lower than the one generated by the model that does not account for clustering.

Across both the jackknife and the bootstrap, estimated SEs are relatively similar. Broadly speaking, the jackknife and bootstrap SE estimates demonstrate that the unadjusted SEs may be poorly estimated when the effect of clustering of families is not taken into account for the bipolar joint models.

3.2 Baboon Data Set

A similar analysis was performed on a data set concerning the survival of female baboons and their levels of fecal glucocorticoid. However, since there are only 13 clusters, the data proved to be not suitable for the cluster-level jackknife and bootstrap. Most of the jackknife samples (61.5%) and almost half of the bootstrap samples (48.0%) resulted in non-convergent joint models. Further description of the data and analysis of the baboon data set can be found in Appendix A.

Chapter 4

Simulation Study

This chapter will focus on the methodologies and results of a simulation study. In order to assess the use of marginal models for joint models with clustered data, longitudinal covariate and time-to-event data is generated with an added family-level random effect and fit with the joint models implemented in the `JM` package (Rizopoulos, 2010). Then, the group jackknife and group bootstrap are applied to estimate the SEs of the model parameter estimates. The marginal model approaches will be compared to one another as well as the joint model fit without accounting for family-level clustering.

The simulation study data was generated based on the results from the bipolar data analysis presented in Section 3.1. The event times were generated based on work by Bender et al. (2005) and Austin (2012). Bender et al. (2005) derived and demonstrated methods for generating event times using the exponential, Weibull or Gompertz distributions with time-fixed covariates and Austin (2012) extended Bender et al.'s (2005) methods to include a continuous time-varying covariate. This chapter expands the method of Austin (2012) to incorporate a family-level random effect when generating event times using the exponential distribution.

The results of the simulations are presented as the average of the estimates across the iterations. Consider some arbitrary parameter ϕ, then the mean of the parameter estimates $\overline{\phi}_{sim}$ is represented as

$$\overline{\phi}_{sim} = \frac{1}{n_{sim}} \sum_{m=1}^{n_{sim}} \hat{\phi}_m \tag{4.1}$$

where n_{sim} represents the number of simulation iterations, m indexes the iteration number ($m = 1, 2, \ldots, n_{sim}$), and $\hat{\phi}_m$ is the parameter estimate from the m^{th} iteration. The empirical standard deviation (SD) is an estimate of the standard deviation of the sampling distribution

of an estimator. The empirical SD is calculated as

$$\frac{1}{n_{sim}-1}\sum_{m=1}^{n_{sim}}\left(\hat{\phi}_m-\overline{\phi}_{sim}\right)^2 \tag{4.2}$$

where $\hat{\phi}_m$ represents the parameter estimate from the m^{th} iteration. In order to compare the results of the simulation study, the percent relative difference (% RD) between the mean SE estimates and the SD will be calculated using

$$\left(\frac{\text{Mean SE}-\text{SD}}{\text{SD}}\right)\times 100\% \tag{4.3}$$

where mean SE is the average of the estimated SEs of the model parameter estimates from each iteration. These metrics will be used to compare the SEs estimated by the jackknife and bootstrap to SEs estimated by the joint model that does not account for family-level clustering.

The simulation was conducted using the statistical software `R` (R Core Team, 2021) as well as the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) supported by Compute/Calcul Canada (www.computecanada.ca). All the source code for the simulation can be found in Appendix C.3.

4.1 Method for Generating Event Times

Recall the Cox model with time-fixed covariates of the form $h_{gi}(t)=h_0(t)\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})$ where $h_0(t)$ is the baseline hazard, $\boldsymbol{\gamma}$ is a $h\times 1$ vector of regression coefficients, and $\boldsymbol{w_{gi}}$ is a vector of time-independent covariates (Cox, 1972). As shown in Equation 2.10, the corresponding survival function of the Cox model is $S_{gi}(t)=\exp\{-H_{gi}(t)\}=\exp\left\{-H_0(t)\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})\right\}$ where $H_0(t)$ is the cumulative baseline hazard function defined as $H_0(t)=\int_0^t h_0(u)\,du$.

Bender et al. (2005) express the distribution function of the event times under the Cox model as

$$F_{gi}(t)=1-\exp(-H_0(t)\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})). \tag{4.4}$$

Let T be a random variable with distribution function F, then $U=F(T)$ follows the uniform distribution in the interval $[0,1]$ (Bender et al., 2005). It follows that if $U\sim \text{Unif}[0,1]$, then $1-U\sim \text{Unif}[0,1]$ (Bender et al., 2005). Let T_{gi} be the survival time of the Cox model,

then Bender et al. (2005) show that it follows from Equation 4.4 that

$$U_{gi} = \exp\left[-H_0(T_{gi})\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})\right] \sim \text{Unif}[0,1]. \tag{4.5}$$

Bender et al. (2005) further demonstrate that if $h_0(t) > 0$ for all t, then H_0 can be inverted and the survival time T_{gi} of the Cox model is of the form

$$T_{gi} = H_0^{-1}\left(\frac{-\log(U_{gi})}{\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})}\right) \tag{4.6}$$

where $U_{gi} \sim \text{Unif}[0,1]$. In a similar vein, then

$$T_{gi} = H_{gi}^{-1}(-\log(U_{gi})) \tag{4.7}$$

where now it is a function of the cumulative hazard instead of the baseline cumulative hazard.

Austin (2012) uses a similar method to include a single time-varying covariate in the Cox model expressed as

$$h_{gi}(t) = h_0(t)\exp\{\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}} + \tau\zeta_{gi}(t)\} \tag{4.8}$$

where $\zeta_{gi}(t)$ is a time-varying covariate and τ is the corresponding regression coefficient. Austin (2012) assumes that the time-varying covariate $\zeta_{gi}(t)$ is proportional to t where $\zeta_{gi}(t) = kt$ and $k > 0$. Further, Austin (2012) lets the event times be from a Cox model with $h_0(t) = \lambda$. Then by inverting the Cox model cumulative hazard function $H(t)$ and exploiting the relationship $T_{gi} = H^{-1}(-\log(U_{gi}))$, the survival times can be generated by

$$T_{gi} = \frac{1}{\tau k}\log\left(1 + \frac{\tau k(-\log(U_{gi}))}{\lambda\exp(\boldsymbol{\gamma}^\top \boldsymbol{w_{gi}})}\right) \tag{4.9}$$

where $U_{gi} \sim \text{Unif}[0,1]$ (Austin, 2012).

Stefan (2019) and Lowe (2020) extended Austin's method to allow a subject-specific random effect in the Cox model. In this thesis, the method was further extended to allow an additional family-level random effect, as follows.

Let $m_{gi}^F(t) = \beta_0 + \beta_1 t + b_{gi} + f_g$ where $b_{gi} \sim N(0, D)$ and f_g denotes the cluster-level random effect. Suppose the survival data follows a Cox model with hazard

$$h_{gi}(t) = h_0(t)\exp\left\{\alpha(m_{gi}^F(t))\right\} = h_0(t)\exp\left\{\alpha(m_{gi}(t) + f_g)\right\} \tag{4.10}$$

where $f_g \sim N(0, \sigma_f^2)$.

To derive an expression to simulate event times based on the joint model, consider the cumulative hazard function for Equation 4.10:

$$H_{gi}(t, m_{gi}(t)) = \int_0^t h_0(u) \exp\{\alpha(m_{gi}(u) + f_g)\}\, du. \tag{4.11}$$

Let $h_0(t) = \lambda$ and substitute in $m_{gi}(t) = \beta_0 + \beta_1 t + b_{gi}$. Then,

$$\begin{aligned} H_{gi}(t, m_{gi}(t)) &= \int_0^t \lambda \exp\{\alpha(\beta_0 + \beta_1 u + b_{gi} + f_g)\}\, du \\ &= \lambda \exp\{\alpha(\beta_0 + b_{gi} + f_g)\} \int_0^t \exp\{\alpha\beta_1 u\}\, du \\ &= \lambda \exp\{\alpha(\beta_0 + b_{gi} + f_g)\} \left[\frac{1}{\alpha\beta_1} \exp\{\alpha\beta_1 u\}\right]_0^t \\ &= \frac{\lambda}{\alpha\beta_1} \exp\{\alpha(\beta_0 + b_{gi} + f_g)\} \big[\exp\{\alpha\beta_1 t\} - 1\big]. \end{aligned} \tag{4.12}$$

Finding the inverse of Equation 4.12 yields

$$H_{gi}^{-1}(u) = \frac{1}{\alpha\beta_1} \log\left(1 + \frac{\alpha\beta_1 u}{\lambda \exp\{\alpha(\beta_0 + b_{gi} + f_g)\}}\right). \tag{4.13}$$

It follows from $T_{gi} = H^{-1}(-\log(U_{gi}))$ that

$$T_{gi} = \frac{1}{\alpha\beta_1} \log\left(1 - \frac{\alpha\beta_1 \log(U_{gi})}{\lambda \exp\{\alpha(\beta_0 + b_{gi} + f_g)\}}\right). \tag{4.14}$$

Then, Gn_g event times are generated using Equation 4.14 where U_{gi} is a pseudo-generated Unif[0,1] number, the G family random effects f_g are generated from $N(0, \sigma_f^2)$ and Gn_g subject-specific random intercept terms b_{gi} are generated from $N(0, D)$.

4.2 Simulation and Results

To fit joint models, data for both the observed longitudinal covariate and time-to-event outcome must be generated. In order to generate the data, the longitudinal covariate was generated based on a linear mixed-effects (LME) model. To simplify the simulation, no time-fixed covariates and only one time-varying covariate were generated. Additionally, the

baseline hazard is kept constant, that is $h_0(t) = \lambda$.

Recall the longitudinal sub-model and notation from Equation 2.14,

$$y_{gi}(t) = m_{gi}(t) + \varepsilon_{gi}(t) = \boldsymbol{x}_{\boldsymbol{gi}}^{\top}(t)\boldsymbol{\beta} + \boldsymbol{z}_{\boldsymbol{gi}}^{\top}(t)\boldsymbol{b}_{\boldsymbol{gi}} + \varepsilon_{gi}(t) \tag{4.15}$$

where $g = 1, \ldots, G$ indexes the clusters, $i = 1, \ldots, n_g$ indexes the subjects in cluster g, and $j = 1, \ldots, n_{gi}$ indexes the observation for each subject within the g^{th} cluster. It is assumed that $b_{gi} \sim N(0, D)$ and $\varepsilon_{gi}(t) \sim N(0, \sigma_{\varepsilon}^2)$.

For the purposes of this simulation, we will add the cluster-level effect to both the longitudinal and survival sub-models. Let f_g denote the cluster-level random effect such that the longitudinal sub-model for the observed time-varying covariate $y_{gi}^F(t)$ with a family random effect is expressed as

$$y_{gi}^F(t) = m_{gi}^F(t) + \varepsilon_{gi}(t) = \beta_0 + \beta_1 t + f_g + b_{gi} + \varepsilon_{gi}(t) \tag{4.16}$$

where $m_{gi}^F(t)$ represents the smoothed time-varying covariate with an added family (i.e. cluster) random effect, t represents time, β_0 is the common intercept term, β_1 is the slope term, b_{gi} is the random intercept term for individual i in cluster g, and $\varepsilon_{gi}(t)$ is the random error term for individual i in cluster g. The random intercept term $b_{gi} \sim N(0, D)$. Similarly, $\varepsilon_{gi}(t) \sim N(0, \sigma_{\varepsilon}^2)$ where σ_{ε}^2 is a constant variance term.

To simulate the observed time-varying covariate $y_{gi}^F(t)$, the intercept β_0, the slope term β_1, the number of clusters (or families) G, the number of individuals n_g per cluster, and the number of observations per individual n_{gi} are specified. Let the observed times of the n_{gi} longitudinal measurements for individual i in cluster g be $t_{gi} = 0, 1, \ldots, n_{gi} - 1$. We then generate Gn_g subject-specific random intercept terms b_{gi} from $N(0, D)$ and $Gn_g n_{gi}$ measurement-specific error terms from $N(0, \sigma^2)$ where G, n_g, and n_{gi} are all specified. Additionally, G family-specific random effect terms are generated from $N(0, \sigma_f)$. Then, the observed time-varying covariate, $y_{gi}^F(t)$, can be calculated using Equation 4.16.

The Gn_g survival times are generated from Equation 4.14 in Section 4.1 where U_{gi} is a pseudo-random Unif[0,1] number. Note that the values of the random effects b_{gi} and f_g are the same in both the survival and longitudinal sub-models.

Lastly, the generated event times are used to filter the observed longitudinal measurements, $y_{gi}^F(t)$, to exclude any that were observed after the event time for each individual. That is, for some individual i within the g^{th} cluster, any simulated longitudinal measurements taken at any time greater than the event time were removed from the simulated data set.

Since all event times are greater than 0 and $t_{gi1} = 0$ for all $g = 1, \ldots, G$ and $i = 1, \ldots, n_g$, all individuals within each cluster remained; however the number of longitudinal measures, n_{gi}, may vary.

Overall, the simulation study was conducted by generating 1000 data sets, subsequently referred to as iterations, according to a joint model with an added family-level random. The simulation data was generated from the following joint model:

$$\begin{cases} h_{gi}(t) = \lambda \exp\{\alpha(\beta_0 + \beta_1 t + f_g + b_{gi})\} \\ y^F_{gi}(t) = \beta_0 + \beta_1 t + f_g + b_{gi} + \varepsilon_{gi}(t) \\ b_{gi} \sim N(0, \boldsymbol{D}), \quad f_g \sim N(0, \sigma^2_f), \quad \varepsilon_{gi}(t) \sim N(0, \sigma^2). \end{cases} \tag{4.17}$$

Note that since only the random-intercept model is considered for this simulation, the variance-covariance matrix D is of size 1 by 1 wherein the only element represents the variance of the subject-specific random effects.

The simulation is based on the fitted joint model parameters from the bipolar data set (Duffy et al., 2014). The true parameters used for the simulation are summarized in Table 4.1 where N is the total sample size, G is the number of family clusters, n_g is the number of subjects in cluster g, and n_{gi} is the number of observations per subject i in cluster g. Three different values of σ_f are considered in order to assess the parameter and SE estimates at varying levels of intra-cluster correlation. As σ_f increases, the intra-cluster correlation increases. The higher the intra-cluster correlation, the higher the between-cluster variability. For example, when $\sigma_f = 0.001$, there is very little intra-cluster correlation, indicating there is very small effect of clustering. Recall that the longitudinal measurements are filtered so that all observed longitudinal measurements of individual i in cluster g are less than the event time for that individual. The average summary statistics for the number of longitudinal covariate measurements per subject of the 1000 generated data sets after filtering is provided in Table 4.2.

Table 4.1: True parameter values used to generate event time and longitudinal covariate data for the simulation studies before filtering. These parameters are held constant for all values of $\sigma_f \in \{0.001, 0.5, 1\}$

N	G	n_g	n_{gi}	β_0	β_1	α	λ	D	σ
500	100	5	4	1	0.02	1.2	0.1	0.410	0.6

Joint models were fit using the `jointModel` function (Rizopoulos, 2010) from the `JM`

Table 4.2: Average summary statistics for the number of longitudinal covariate measurements per subject, n_{gi}, for 1000 iterations of generated data after data was filtered to remove longitudinal measurements taken after event time for three values of σ_f used in the simulation.

σ_f	Mean	Median	Minimum	Maximum
0.001	2.531	2.287	1.000	4.000
0.500	2.513	2.243	1.000	4.000
1.000	2.490	2.247	1.000	4.000

(Rizopoulos, 2010) package. The models were fit using SHARCNET (www.sharcnet.ca). The joint model does not account for any family-level random effects. Thus the fitted models are misspecified. The fitted joint model parameter estimates are summarized in Table 4.3.

Table 4.3 shows that the `jointModel` coefficient estimates appear to be relatively unbiased, even as σ_f increases. The only exception is that the estimate for D tends to overestimate the parameter as σ_f increases. This overestimation is due to the family-level variation which is not accounted for by the `jointModel` function. In fact, it can be seen that the mean estimate for D is approximately equal to the sum of the true D and σ_f^2. This is true for all values of σ_f.

When looking at the percent relative difference (% RD), as σ_f increases, the SDs for β_0, β_1, $\log(\lambda)$ and α are all underestimated by the SEs generated by the model (JM SE). The underestimation is more pronounced as σ_f gets larger for β_0. The σ and D SDs are the only ones to be overestimated by the JM SE. The overestimation for σ gets more pronounced as σ_f increases, however, the JM SE estimate for the SD of D goes from overestimating to greatly underestimating the SD as σ_f increases.

Overall, the JM SEs and the SDs of the parameters do not align. Even in the case where there is little intra-cluster correlation, that is, when $\sigma_f = 0.001$, most of the SE estimates underestimate the SDs. This is true except for D SD, which is severely overestimated. Only the JM SE estimate of the SD for σ appears to have a % RD of 0. This indicates that the SEs produced by the misspecified `JM` model are poor estimates for the true SEs in the presence of a family-level random effect. Additionally, except for D, all model parameter estimates are apparently unbiased for varying values of σ_f. Therefore, the joint model data generated with a family-level random effect in both the longitudinal and survival sub-models is a good candidate for marginal modeling approaches with SEs estimated by the jackknife or bootstrap.

Table 4.3: Mean parameter estimates (Mean Est.) obtained from the `JM` joint model fitted to data simulated with a family-level term in both the longitudinal & survival sub-model (100% convergence) with varying values of $\sigma_f \in \{0.001, 0.5, 1\}$ and based on 1000 iterations of the simulation study. The empirical standard deviation (SD) of the estimates and mean of the standard errors from `JM` (JM SE) are also provided and the percent relative difference (% RD) between the two is calculated.

		True value	Mean Est.	SD	JM SE	% RD
$\sigma_f = 0.001$	β_0	1.000	0.998	0.048	0.036	−25.0
	β_1	0.020	0.021	0.013	0.011	−15.4
	$\log(\lambda)$	−2.303	−2.326	0.197	0.178	− 9.6
	α	1.200	1.218	0.147	0.130	−11.6
	σ	0.600	0.598	0.025	0.025	0.0
	D	0.410	0.409	0.040	0.093	132.5
$\sigma_f = 0.5$	β_0	1.000	1.002	0.070	0.042	−40.0
	β_1	0.020	0.021	0.010	0.009	−10.0
	$\log(\lambda)$	−2.303	−2.318	0.172	0.156	− 9.3
	α	1.200	1.210	0.115	0.104	− 9.6
	σ	0.600	0.599	0.024	0.025	4.2
	D	0.410	0.660	0.064	0.083	29.7
$\sigma_f = 1$	β_0	1.000	1.003	0.136	0.054	−60.3
	β_1	0.020	0.021	0.006	0.005	−16.7
	$\log(\lambda)$	−2.303	−2.285	0.151	0.135	−10.6
	α	1.200	1.187	0.088	0.078	−11.4
	σ	0.600	0.612	0.023	0.025	8.7
	D	0.410	1.415	0.164	0.073	−55.5

The group jackknife and group bootstrap methods were applied to the 1000 data sets using SHARCNET (www.sharcnet.ca). Each data set resulted in 100 jackknife samples and 200 bootstrap resamples. Each of the samples were fit with the misspecified `JM` joint model that does not account for family-level clustering. Since all family-level clusters are of equal size, only the equal jackknife (EJK) and the unweighted bootstrap (BS) estimates for SE were calculated. Table 4.4 summarizes the results of the jackknife and bootstrap.

It should be noted that there are some issues when applying the jackknife and bootstrap methods, especially as σ_f increases. All 100 jackknife and 200 bootstrap samples provided joint models that converged for each of the 1000 iterations. However, upon convergence some

models resulted in a non-positive definite Hessian matrix as provided by a warning from the `jointModel` function. Any such iterations were removed from the final average calculation. There were 21 iterations removed from the EJK calculations when $\sigma_f = 1$, and 1 and 145 iterations removed from the BS calculations when $\sigma_f = 0.5$ and $\sigma_f = 1$, respectively.

Table 4.4: Average SE estimates accounting for family-level clustering based on 1000 iterations. The SE estimates from the misspecified joint model (JM SE), the group equal jackknife (EJK) and the group bootstrap (BS) are provided. The empirical standard deviation (SD) and the percent relative differences (%RD) between the SE of the coefficient estimates and the SD are also provided.

			Mean estimated SE			
σ_f	Parameter	True value	SD	JM SE (% RD)	EJK* (% RD)	BS† (% RD)
0.001	β_0	1.000	0.048	0.036 (−25.0)	0.035 (−27.1)	0.035 (−27.1)
	β_1	0.020	0.013	0.011 (−15.4)	0.011 (−15.4)	0.011 (−15.4)
	$\log(\lambda)$	-2.303	0.197	0.178 (− 9.6)	0.180 (− 8.6)	0.183 (− 7.1)
	α	1.200	0.147	0.130 (−11.6)	0.132 (− 10.2)	0.135 (− 8.2)
	σ	0.600	0.025	0.025 (0.0)	0.015 (−40.0)	0.015 (−40.0)
	D	0.410	0.040	0.093 (132.5)	0.038 (− 5.0)	0.038 (− 5.0)
0.5	β_0	1.000	0.070	0.042 (−40.0)	0.063 (−10.0)	0.063 (−10.0)
	β_1	0.020	0.010	0.009 (−10.0)	0.009 (−10.0)	0.009 (−10.0)
	$\log(\lambda)$	-2.303	0.172	0.156 (− 9.3)	0.159 (− 7.6)	0.161 (− 6.4)
	α	1.200	0.115	0.104 (− 9.6)	0.106 (− 7.8)	0.108 (− 6.1)
	σ	0.600	0.024	0.025 (4.2)	0.015 (−37.5)	0.015 (−37.5)
	D	0.410	0.064	0.083 (29.7)	0.063 (− 1.6)	0.062 (− 3.1)
1	β_0	1.000	0.136	0.054 (−60.3)	0.131 (− 3.7)	0.132 (− 2.9)
	β_1	0.020	0.006	0.005 (−16.7)	0.006 (0.0)	0.006 (0.0)
	$\log(\lambda)$	-2.303	0.151	0.135 (−10.6)	0.147 (− 2.6)	0.143 (− 5.3)
	α	1.200	0.088	0.078 (−11.4)	0.088 (0.0)	0.085 (− 3.4)
	σ	0.600	0.023	0.025 (8.7)	0.016 (−30.4)	0.015 (−34.8)
	D	0.410	0.164	0.073 (−55.5)	0.169 (3.0)	0.166 (1.2)

* The average EJK calculations for $\sigma_f = 1$ are based on 979 iterations.

† The average BS calculations for $\sigma_f = 0.5$ and $\sigma_f = 1$ are based on 999 and 855 iterations, respectively.

Generally, for all values of σ_f, the EJK and BS estimates are on average much closer to the SD than those generated by the model (JM SE). Further, the EJK and the BS estimates are similar. Recall that % RD is the percent relative difference between the mean estimated

SE and the SD. The range of the % RD for the EJK and BS estimates are $(-40.0, 3.0)$ and $(-40.0, 1.2)$ respectively. The range of the % RDs of the JM SEs not accounting for clustering is $(-60.3, 132.5)$. When there is very little family-level variance, that is $\sigma_f = 0.001$, both the EJK and BS estimates underestimate the SD for the intercept term β_0, the slope term β_1 and σ by about 27%, 15% and 40%, respectively. As σ_f increases, the EJK and BS estimates for the SD improve for β_0 and β_1; however they continue to considerably underestimate the SD for σ. The EJK and BS estimates for $\log(\lambda)$ and D appear to consistently have % RDs closer to 0 than the JM SE estimates. Additionally, for all values of σ_f, the % RD for the EJK and BS SE estimates of $\log(\lambda)$ and D are relatively close to 0.

Figure 4.1 shows the average SE estimates from the three estimation methods compared to the SD. Notably, all methods of SE estimation tend to underestimate the SDs, except for JM SE in the case of σ and D. For β_0, β_1, $\log(\lambda)$, α, and D, the EJK and BS have less empirical bias than the JM SE for all values of σ_f. The JM SE performs very poorly for the SD of D, especially when compared to the EJK and BS which have relatively small % RDs for all values of σ_f. Overall, as σ_f increases, the EJK and BS tend to perform increasingly better than the JM SE.

Of particular interest are the estimates for the SD of the σ model parameter estimates. For all values of σ_f, the model-based SE either estimates the SD well or slightly overestimates it. However, the EJK and the BS methods severely underestimate the SD for σ by at least 30%. This is an unexpected result and further investigation may be required.

Overall, the simulation study suggests that the EJK and BS estimates of the β_0, β_1, $\log(\lambda)$, α, and D SEs are preferable to the model-based SEs. Computationally, the jackknife approach requires less computer resources and provides fewer models with non-positive definite Hessian matrices than the bootstrap. However, both methods perform similarly in terms of SE estimation. Therefore, the marginal modelling approaches with SEs estimated by the jackknife or bootstrap seem to be acceptable methods to account for intra-cluster correlation.

Recall that as expected, the subject-level variance, D, is increasingly over-estimated as the family-level standard deviation term σ_f increases as shown in Table 4.3. Additionally, the simulation study shows that regardless of the estimation method, the σ SE tends to be poorly estimated in the presence of intra-cluster correlation.

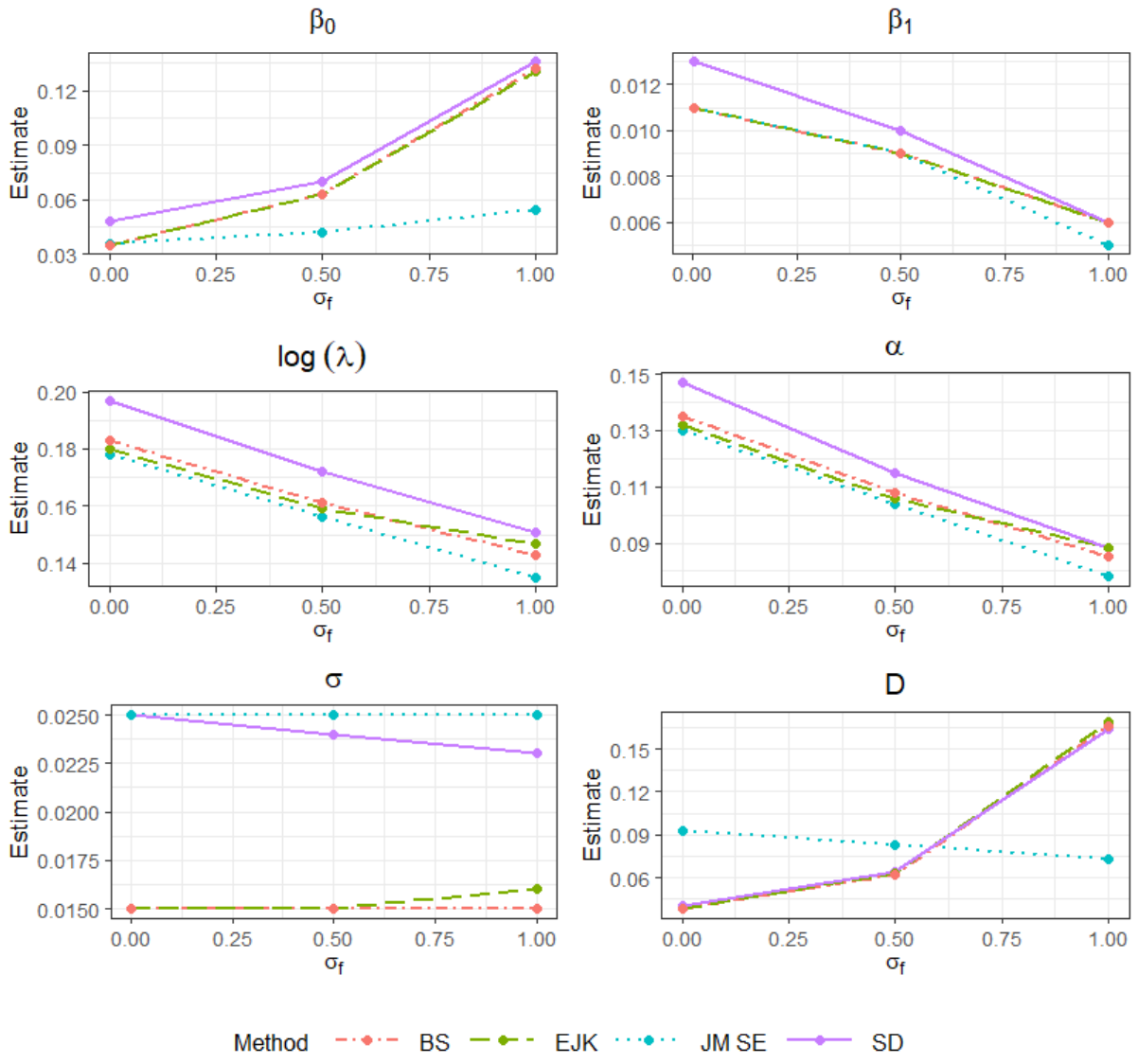

Figure 4.1: Average standard error estimates for the joint model parameter estimates across 1000 iterations from the joint model not accounting for family-level clustering: model-based (JM SE), group jackknife (EJK), and group bootstrap (BS) compared to the empirical standard deviation (SD) for varying values of $\sigma_f \in \{0.001, 0.5, 1\}$.

Chapter 5

Conclusion and Further Work

A form of marginal models used in survival analysis is to first estimate model parameters assuming that individuals are independent and then estimate the SE of the model parameter estimates using a method that recognizes the family-level intra-cluster correlation (Xiao & Abrahamowicz, 2010). The grouped jackknife and grouped bootstrap-based marginal approaches have successfully been applied to Cox models (Lipsitz et al., 1994; Xiao & Abrahamowicz, 2010); therefore, the aim of this thesis was to explore these marginal modelling approaches for joint models with clustered data.

The marginal modelling approach was applied to a real life data set concerning the onset of bipolar disorder (BD), major depressive disorder, or schizoaffective disorder in children with at least one parent diagnosed with BD. The bipolar data set analysis found that the jackknife and bootstrap methods produced slightly higher estimates of the model parameter estimates' SEs than that of the standard joint model. This indicates that intra-cluster correlation may be present which may lead to underestimation of the the model parameter estimates' SEs. A second data set was analyzed concerning the survival of female baboons and fecal glucocorticoid levels; however, too few clusters in the data set proved to be unsuitable for the group jackknife and group bootstrap methods.

To generate the time-to-event data, the method formulated by Austin (2012), Stefan (2019) and Lowe (2020) was extended to incorporate the family-level term f_g. Data sets were generated based on varying levels of σ_f to assess the quality of the estimators at various levels of intra-cluster correlation. The simulation study showed that parameter estimates were unbiased regardless of the value of the family-level variance, except for the parameter D. The parameter D, which represents the variance of the subject-level random effects, was generally over-estimated, likely a result of the family-level random effect not being taken

into consideration when fitting the model. Further, the simulation study showed that the % RD from the empirical SD is reduced for the jackknife and bootstrap SE estimates when compared to the model-based SEs which ignored the family-level correlation.

A notable issue with the EJK and BS estimation methods was the significant underestimation of the SD of $\hat{\sigma}$. Further investigation of this is required. Another issue is that some of resamples required for the EJK and BS methods resulted in non-positive definite Hessian matrices. This issue was more prominent when the family-level standard deviation σ_f was large. Applying the jackknife and bootstrap to the baboon data set proved unworkable since so many resamples had non-positive definite Hessians. However, this can likely be attributed to the small number of clusters.

To my knowledge, there exists no literature applying marginal methods with jackknife and bootstrap to joint models. This thesis has investigated marginal modelling approaches for the simplest joint model wherein there is a random intercept term with no fixed covariates in either sub-model. In the absence of `R` software (R Core Team, 2021) that accounts for any family-level clustering, we have shown that marginal modelling approaches to joint models are appropriate methods for incorporating intra-cluster correlation into the model. However, further research should be conducted to ensure that this holds in the case of unequal family-level clusters or when the model becomes increasingly complex. Future work should prioritize developing `R` packages that incorporate a family-level clustering term in both the longitudinal and survival joint model sub-models to avoid having to apply the more computationally intensive jackknife and bootstrap.

Situations may arise where individuals have no intra-cluster correlation with respect to the time-varying covariate, but may have correlated survival outcomes. This may occur in cases where genetic or environmental factors affect survival but not the time-varying covariate. Using the notation defined in this thesis, a joint frailty model can be written as

$$\begin{cases} h_{gi}(t) = \lambda \exp\{\alpha(\beta_0 + \beta_1 t + b_{gi} + f_g)\} \\ y_{gi}(t) = \beta_0 + \beta_1 t + b_{gi} + \varepsilon_{gi}(t) \\ b_{gi} \sim N(0, \sigma_b^2), \quad f_g \sim N(0, \sigma_f^2), \quad \varepsilon_{gi}(t) \sim N(0, \sigma^2) \end{cases} \tag{5.1}$$

where the family-level random effect is incorporated into only the survival sub-model. The index g represents the g^{th} family for $g = 1, \ldots, G$ and the index i represents the i^{th} subject for $i = 1, \ldots, n_g$ (Horrocks, n.d.). Ancillary to the aim of this thesis, data was generated according to the model in Equation 5.1 and was fit using the `jointModel` function (Rizopoulos,

2010). The summarized results can be found in Appendix B. The results of this indicate that there is also a need for R software to incorporate frailty terms into modelling joint longitudinal and survival data.

In conclusion, this thesis proposed applying marginal approaches to joint models with clustered data. This research showed that both the group jackknife and group bootstrap-based marginal approaches are preferable to the model fit without considering intra-cluster correlation. It should be noted that the marginal approach does not correct the biased estimate of D and alternative approaches, such as fitting a model with an added family-level random effect term may prove preferable in the future.

References

Aalen, O. O., Borgan, Ø., & Gjessing, H. K. (2008). *Survival and event history analysis: A process point of view* (1st ed. 2008. ed.). New York, NY: Springer.

Altzerinakou, M., & Paoletti, X. (2021). Change-point joint model for identification of plateau of activity in early phase trials. *Statistics in Medicine*, *40*(9), 2113-2138.

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. doi: 10.1214/aos/1176345976

Anota, A., Barbieri, A., Savina, M., Pam, A., Gourgou-Bourgade, S., Bonnetain, F., & Bascoul-Mollevi, C. (2014). Comparison of three longitudinal analysis models for the health-related quality of life in oncology: a simulation study. *Health and Quality of Life Outcomes*, *12*, 192. doi: https://doi.org/10.1186/s12955-014-0192-2

Austin, P. C. (2012). Generating survival times to simulate Cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, *31*(29), 3946-3958.

Bender, R., Augustin, T., & Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, *25*(11), 1978-1979.

Busing, F. M. T. A., Meijer, E., & Leeden, R. V. D. (1999). Delete-m jackknife for unequal m. *Statistics and Computing*, *9*(1), 3-8.

Campos, F. A., Archie, E. A., Gesquiere, L. R., Tung, J., Altmann, J., & Alberts, S. C. (2021). Glucocorticoid exposure predicts survival in female baboons. *Science Advances*, *7*(17). doi: 10.1126/sciadv.abf6759

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B, Methodological*, *34*(2), 187-220.

Davison, A. C., & Hinkley, D. V. (1999). *Bootstrap methods and their application.* (Vol. 1 (Reprinted with corrections)). Cambridge University Press.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1-38.

Diggle, P. J., Heagerty, P., Liang, K.-Y., & Zeger, S. (2013). *Analysis of longitudinal data* (Second [paperback] edition. ed.). Oxford: Oxford University Press.

Du, R., & Lee, J.-H. (2019). A weighted jackknife method for clustered data. *Communications in Statistics. Theory and Methods*, *48*(8), 1963-1980.

Duffy, A., Horrocks, J., Doucette, S., Keown-Stoneman, C., McCloskey, S., & Grof, P. (2014). The developmental trajectory of bipolar disorder. *The British Journal of*

Psychiatry, *204*(2), 122-128.
Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1 – 26. doi: 10.1214/aos/1176344552
Efron, B., & Stein, C. (1981). The jackknife estimate of variance. *The Annals of Statistics*, *9*(3), 586 – 596. doi: 10.1214/aos/1176345462
Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.
Field, C. A., & Welsh, A. H. (2007). Bootstrapping clustered data. *Journal of the Royal Statistical Society: Series B*, *69*, 369-390.
Fitzmaurice, G. M., Laird, N. M., & Ware, J. H. (2004). *Applied longitudinal analysis*. Hoboken, N.J: Wiley-Interscience.
Galbraith, S., Daniel, J. A., & Vissel, B. (2010). A study of clustered data and approaches to its analysis. *Journal of Neuroscience*, *30*(32), 10601–10608. doi: 10.1523/JNEUROSCI.0362-10.2010
Hamilton, M. (1959). The assessment of anxiety states by rating. *British Journal of Medical Psychology*, *32*, 50-55. doi: https://doi.org/10.1111/j.2044-8341.1959.tb00467.x
Hollingshead, A. B. (1957). Two factor index of social position. *Yale University*.
Horrocks, J. (n.d.). *Joint frailty model.* (Preprint)
Hsieh, F., Tseng, Y.-K., & Wang, J.-L. (2006). Joint modeling of survival and longitudinal data: Likelihood approach revisited. *Biometrics*, *62*(4), 1037-1043. doi: https://doi.org/10.1111/j.1541-0420.2006.00570.x
Ibrahim, J. G., Chu, H., & Chen, L. M. (2010). Basic concepts and methods for joint models of longitudinal and survival data. *Journal of Clinical Oncology*, *28*(16), 2796-2801. doi: https://doi.org/10.1200/JCO.2009.25.0654
Kalbfleisch, J. D., & Prentice, R. L. (2002). *The statistical analysis of failure time data*. Hoboken: John Wiley & Sons, Incorporated.
Lange, K. (2013). *Optimization* (2nd ed. 2013. ed.). New York, NY: Springer.
Lawless, J. F. (2003). *Statistical models and methods for lifetime data* (Vol. 2). Hoboken, N.J: Wiley-Interscience.
Li, K., & Luo, S. (2017). Functional joint model for longitudinal and time-to-event data: an application to Alzheimer's disease. *Statistics in Medicine*, *36*(22), 3560-3572.
Lipsitz, S. R., Dear, K. B. G., & Zhao, L. (1994). Jackknife estimators of variance for parameter estimates from estimating equations with applications to clustered survival data. *Biometrics*, *50*(3), 842-846.
Lowe, M. (2020). *The cumulative effects of time-varying covariates in survival analysis.* University of Guelph. (Master's thesis)
Luna, P. N., Mansbach, J. M., & Shaw, C. A. (2020). A joint modeling approach for longitudinal microbiome data improves ability to detect microbiome associations with disease. *PLoS Computational Biology*, *16*(12), 1-45. doi: https://doi.org/10.1371/journal.pcbi.1008473
Moore, D. F. (2016). *Applied survival analysis using R*. Hoboken: Springer.

Papageorgiou, G., Mauff, K., Tomer, A., & Rizopoulos, D. (2019). An overview of joint modeling of time-to-event and longitudinal outcomes. *Annual Review of Statistics and its Application*, *6*(1), 223-240.

Philipson, P., Sousa, I., Diggle, P. J., Williamson, P., Kolamunnage-Dona, R., Henderson, R., & Hickey, G. L. (2018). joineR: Joint modelling of repeated measurements and time-to-event data [Computer software manual]. Retrieved from `https://github.com/graemeleehickey/joineR/` (R package version 1.2.5)

R Core Team. (2021). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `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*, *35*(9), 1-33. Retrieved from `http://www.jstatsoft.org/v35/i09/`

Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data : with applications in R.* Boca Raton: 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. doi: 10.18637/jss.v072.i07

Runcie, D. E., Wiedmann, R. T., Archie, E. A., Altmann, J., Wray, G. A., Alberts, S. C., & Tung, J. (2013). Social environment influences the relationship between genotype and gene expression in wild baboons. *Philosophical Transaction of the Royal Society B: Biological*, *368*, 1618.

Sapolsky, R. M. (2004). Social status and health in humans and other animals. *Annual Review in Anthropology*, *33*, 393-418.

Sherman, M., & le Cessie, S. (1997). A comparison between bootstrap methods and generalized estimating equations for correlated outcomes in generalized linear models. *Communications in Statistics. Simulation and Computation*, *26*(3), 901-925.

Stefan, G. (2019). *A comparison of Cox and joint models for time-to-event data.* University of Guelph. (Master's thesis)

Tsiatis, A. A., & Davidian, M. (2004). Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica*, *14*(3), 809-834.

Ukoumunne, O. C., Davison, A. C., Gulliford, M. C., & Chinn, S. (2003). Non-parametric bootstrap confidence intervals for the intraclass correlation coefficient. *Statistics in Medicine*, *22*(24), 3805-3821.

Wang, M., Kong, M., & Datta, S. (2011). Inference for marginal linear models for clustered longitudinal data with potentially informative cluster sizes. *Statistical Methods in Medical Research*, *20*(4), 347-367. doi: https://doi-org.subzero.lib.uoguelph.ca/10.1177/0962280209347043

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, *50*, 1-25. doi: https://doi.org/10.2307/1912526

Wilde, A., Chan, H. N., Rahman, B., Meiser, B., Mitchell, P. B., Schofield, P. R., & Green, M. J. (2014). A meta-analysis of the risk of major affective disorder in relatives of individuals affected by major depressive disorder or bipolar disorder. *Journal of*

Affective Disorders, *158*, 37-47. Retrieved from https://doi.org/10.1016/j.jad.2014.01.014 doi: 10.1016/j.jad.2014.01.014

Wulfsohn, M. S., & Tsiatis, A. A. (1997). A joint model for survival and longitudinal data measured with error. *Biometrics*, *53*(1), 330-339. Retrieved from http://www.jstor.org/stable/2533118

Xiao, Y., & Abrahamowicz, M. (2010). Bootstrap-based methods for estimating standard errors in Cox's regression analyses of clustered event times. *Statistics in Medicine*, *29*(7-8), 915-923.

Yaribeygi, H., Panahi, Y., Sahraei, H., Johnston, T. P., & Sahebkar, A. (2017). The impact of stress on body function: a review. *EXCLI Journal*, *16*, 1057-1072. doi: 10.17179/excli2017-480

Appendix A

Baboon Data

Stressors can be an event, experience or environmental stimulus that evokes a biological response in an individual (Yaribeygi et al., 2017). In humans, it is generally expected that a chronic stress response leads to reduced survival (Campos et al., 2021). Yet, less is known about the effect of stress on animals such as the baboon (*Papio cynocephalus*). It has been observed that lower ranked baboons within social groups have elevated levels of stress (Sapolsky, 2004). Stress induces a normal release of glucocorticoid (Campos et al., 2021). However, long-term exposure to stress may result in glucocorticoid dysregulation (Campos et al., 2021).

The data for this analysis was collected as part of a longitudinal study of 242 adult female baboons over the span of 1634 female years (Campos et al., 2021). A baboon is considered an adult at the age of 5 in order to be considered for this analysis (Campos et al., 2021). In order to investigate the relationship between survival and glucocorticoid exposure, 14,173 fecal glucocorticoid (fGC) measurements were collected from multiple study groups living in the Amboseli basin in southern Kenya (Campos et al., 2021). In addition to the fGC measurements, the baboons were observed till their death. Their deaths were noted as from the time since adulthood (Campos et al., 2021). Since female baboons are highly sociable creatures who often remain in their social group till death, once a baboon permanently leaves the social group, it is assumed that the female has died (Campos et al., 2021). Further observations were taken of the baboons social rank and relationships with other female and male baboons within the social group (Campos et al., 2021).

A.1 Exploratory Data Analysis

The original data comprises 14,173 fGC measurements from 242 adult female baboons from 13 different social groups. Some fGC measurements in the data from the same subject appear to be taken at the same time, therefore only those first listed within the data were kept for the analysis, dropping 253 measurements. Overall, all 242 subjects were kept in the analysis with a total of 13,920 fGC measurements.

The social structure and group of an individual baboon have been shown to have profound effects on a baboon's gene expression, and survival (Runcie et al., 2013). Therefore, for the purposes of this analysis, the social groups are treated as clusters. There are a total of 13 clusters with a mean size of 18.32 baboons. Table A.1 further outlines the summary statistics of the cluster sizes.

Table A.1: Summary statistics of the baboon clusters ($G = 13$). SD = standard deviation.

	Minimum	Mean (SD)	Median	Maximum
Cluster size	2.00	18.62 (13.28)	16.00	41.00

Table A.2: Summary of the event outcome (i.e. death) of the adult female baboon study subjects ($n = 242$).

Event outcome	Frequency (%)
No	147 (60.7)
Yes	95 (39.3)

Of the 242 adult females, 95 (39.3%) individuals died before the censoring time (see Table A.2). Measurements of social rank were also observed. The baseline dyadic sociality index (DSI) of the baboons was measured with the other females as well as the other males in the social group. The DSI is a measure of the social bond strength with the female's top three social bonds with females (DSI_F) and males (DSI_M) (Campos et al., 2021). Overall, the bond strengths with other females was observed to be stronger than the social bonds with other males in the social group. Another measure of social rank is the baseline proportional rank (PR). The PR refers to the proportion of adult females that the study subject dominates (Campos et al., 2021). That is, the higher the PR, the more dominant the female. Table A.3 further outlines the summary statistics of these covariates.

The fGC measures were measured intermittently over the course of the study and range widely from 7.51 to 983.87. Since the fGC measures are fit to an LME model, the normality

Table A.3: Summary statistics of the female baboons ($n = 242$). The baseline dyadic sociality index with females (DSI_F) and males (DSI_M) as well as the baseline proportional rank (PR) over the the baboon's lifetime were reported. SD = standard deviation.

	Minimum	Mean (SD)	Median	Maximum
Baseline DSI_F	−0.50	0.96 (0.53)	0.98	2.94
Baseline DSI_M	−1.48	0.55 (0.66)	0.59	2.56
Baseline PR	0.00	0.48 (0.31)	0.50	1.00
Survival/censoring time	0.07	7.90 (5.49)	7.00	22.68
fGC measures	7.51	78.39 (40.97)	70.01	982.87
log(fGC) measures	2.02	4.26 (0.45)	4.25	6.89
Number of measures per subject	1.00	57.52 (55.88)	39.00	284.00

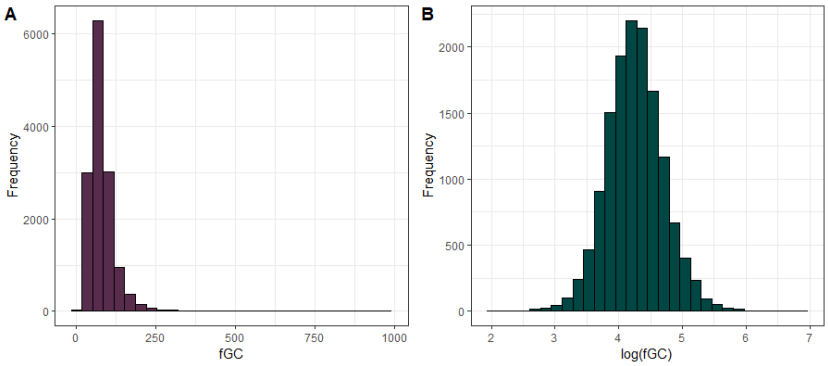

Figure A.1: (A) Frequency of baboon fecal glucocorticoid amounts from 242 baboons. (B) Frequency of baboon log(fecal glucocorticoid) amounts.

assumption of the residuals must be checked. A normal Q-Q plot was fit for the fGC as well as the log(fGC) measures to ensure that the assumption holds. As can be seen in Figure A.2, the log(fGC) measures are more suitable for the model. Therefore, only the log(fGC) measures will be considered for this analysis. To further visualize the relationship between log(fGC) and time, the log(fGC) was plotted over the adulthood of the baboons (see Figure A.3).

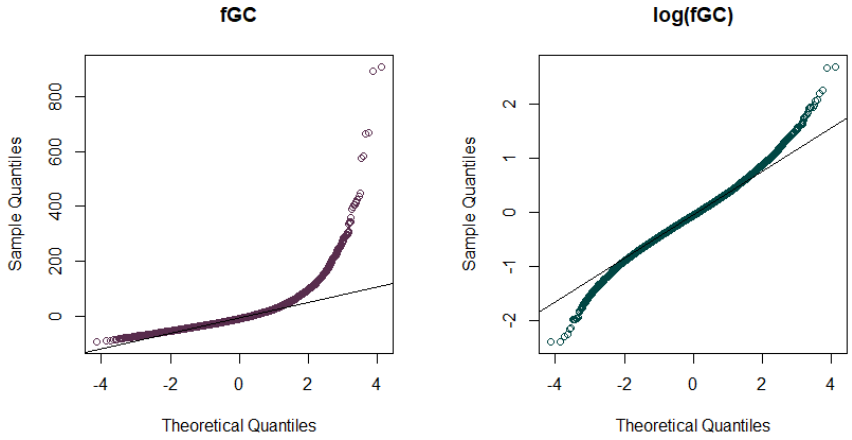

Figure A.2: Normal Q-Q plots for LME models with fGC (A) and log(fGC) (B) as the response to check the normality of the errors.

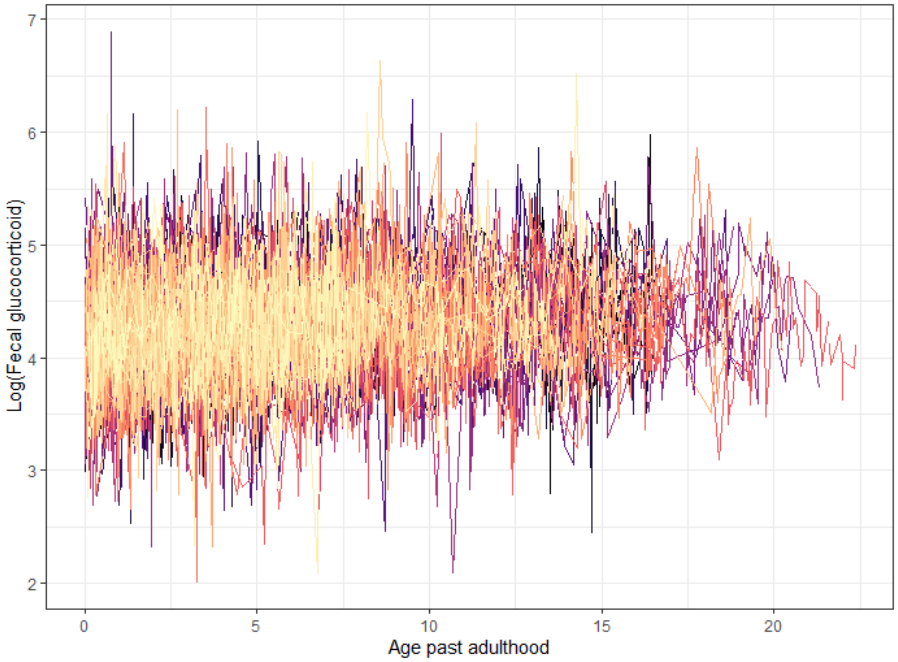

Figure A.3: Individual log(fGC) measures plotted over time for 242 baboons. Age is represented as the time since the beginning of adulthood (5 years of age).

A.2 Data Set Analysis and Discussion

This section outlines the fitted joint model as well as the application of the group jackknife and the group bootstrap. Only a simple model is considered, where fGC is the smoothed

longitudinal covariate. All joint models are fit with the `JM` package (Rizopoulos, 2010) in `R` (R Core Team, 2021) as well as using the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca).

The time-independent covariate used as part of the simple model is DSI_F which is the measure of social bond with other females in the social group upon enrollment in the study. The longitudinal sub-model follows a random intercept and slope mixed-effects model. Thus, the smoothed longitudinal covariate, $m_{gi}(t_{gij})$, is modelled by

$$m_{gi}(t_{gij}) = (\beta_0 + b_{0gi}) + (\beta_1 + b_{igi})t_{gij} + \beta_2 \text{DSI}_{Fgi} \tag{A.1}$$

where $m_{gi}(t_{gij})$ is the expected value of the log(fGC) scores for individual i in cluster g at time t, and t_{gij} is the time at which the response log(fGC) is measured for individual i in the g^{th} cluster at timepoint j ($g = 1, 2, \ldots, 13;\ i = 1, 2, \ldots, 242;\ j = 1, 2, \ldots, n_{gi}$). The random intercept term is denoted by b_{0gi} and b_{1gi} denotes the random slope term. The survival sub-model is of the form

$$h_{gi}(t_{gij}) = h_0(t_{gij}) \exp\{\gamma_1 \text{DSI}_{Fgi} + \alpha m_{gi}(t_{gij})\}) \tag{A.2}$$

where $h_0(t_{gij})$ represents the Weibull-PH baseline hazard.

The results of the fitted joint model are summarized in Table A.4. The longitudinal sub-model indicates that for a one-unit increase in DSI_F, the smoothed log(fGC) measure increases by 0.086 (95% CI: ((0.065, 0.107)) given that age remains constant. Further, as the age of a baboon increases by one year, the smoothed log(fGC) measure increases by 0.024 (95% CI: (0.021, 0.026)) given the social bond strength remains constant.

Table A.4: Joint model summary for the primary baboon sample. SE estimates do not account for clustering within data.

Sub-model	Variable	Coefficient	Standard error	95% CI	p-value
Longitudinal	Intercept	4.047	0.017	(4.013, 4.080)	< 0.001
	DSI_F	0.086	0.011	(0.065, 0.107)	< 0.001
	Age	0.024	0.001	(0.021, 0.026)	< 0.001
Survival	Intercept	-9.711	2.829	$(-15.255, -4.168)$	0.001
	DSI_F	-0.274	0.197	$(-0.665, 0.117)$	0.170
	Association	1.232	0.679	$(-0.098, 2.563)$	0.069
	log(Shape)	0.517	0.087		< 0.001

The survival sub-model suggests that there is some evidence to suggest a relationship between the smoothed log(fGC) scores and the survival time (95% CI: $(-0.098, 2.563)$). However, there is no evidence that the DSI_F (95% CI: $(-0.665, 0.117)$) log-hazard ratio is not 0. Therefore, DSI_F may not have an effect on survival time.

In order to estimate SE to adjust for clustering 13 jackknife and 200 bootstrap samples were taken. Of the 13 jackknife samples, only 5 (38.5%) resulted in viable joint models. Given the nature of the jackknife, it is not possible to replace these jackknife samples. Similar issues arose in the bootstrap samples - 96 of the 200 bootstrap resamples were removed and replaced with other bootstrap resamples. The number of non-viable jackknife and bootstrap samples results in estimates that are not reliable. Therefore, this form of analysis is problematic for this data set and perhaps the cluster-level jackknife and bootstrap is not suitable for data sets with a small number of clusters.

Appendix B

The Joint Frailty Model

The `jointModel` function from the `R` package `JM` (Rizopoulos, 2010) does not account for any shared family effects. In the joint frailty model for correlated outcomes, the random subject effects b_{gi} are shared between the two sub-models (Horrocks, n.d.). However, the random family effects f_g only occur in the survival sub-model (Horrocks, n.d.). For the purposes of this simulation, there are no time-fixed covariates and the baseline hazard is constant, $h_0(t) = \lambda$. Then, using the notation defined in this thesis, the joint frailty model can be written as

$$\begin{cases} h_{gi}(t) = \lambda \exp\{\alpha(\beta_0 + \beta_1 t + b_{gi} + f_g)\} \\ y_{gi}(t) = \beta_0 + \beta_1 t + b_{gi} + \varepsilon_{gi}(t) \\ b_{gi} \sim N(0, \sigma_b^2), \quad f_g \sim N(0, \sigma_f^2), \quad \varepsilon_{gi}(t) \sim N(0, \sigma^2) \end{cases} \tag{B.1}$$

where g represents the g^{th} family for $g = 1, \ldots, G$ and i represents the i^{th} subject for $i = 1, \ldots, n_g$. (Horrocks, n.d.).

To investigate the effect of only including the family-level effect in the survival sub-model data was generated based on Equation B.1 where $g = 1, 2, \ldots, 100$ clusters, $i = 1, 2, \ldots, 5$ individuals per cluster g, and $j = 1, 2, \ldots, 4$ observations per individual i in cluster g. The true model parameters are the same as used in the simulation study from Chapter 4 that are summarized in Table 4.1. That is, $\beta_0 = 1$, $\beta_1 = 0.02$, $\alpha = 1.2$, $\lambda = 0.1$, $\sigma_f = 0.5$, $\sigma_b = 0.64$ and $\sigma = 0.6$. Joint models that do not account for any family-level effects were fit and the results are summarized in Table B.1.

Table B.1: Mean parameter estimates obtained from the `JM` joint model fitted to data simulated with a family-level term in only the survival sub-model with varying values of family-level variance, $\sigma_f \in \{0.001, 0.5, 1\}$ and true parameters based on the simulation study. The empirical standard deviation (SD) and mean of the standard errors (Mean SE) are also provided.

	Parameter	True value	Est.	SD	Mean SE
$\sigma_f = 0.001$	β_0	1.000	0.998	0.048	0.036
	β_1	0.020	0.021	0.013	0.011
	$\log(\lambda)$	-2.303	-2.326	0.197	0.011
	α	1.200	1.218	0.147	0.011
	σ	0.600	0.598	0.025	0.045
	σ_b	0.640	0.639	0.031	0.025
$\sigma_f = 0.5$	β_0	1.000	1.005	0.048	0.035
	β_1	0.020	0.009	0.012	0.010
	$\log(\lambda)$	-2.303	-2.008	0.186	0.010
	α	1.200	1.005	0.137	0.010
	σ	0.600	0.598	0.025	0.045
	σ_b	0.640	0.637	0.032	0.025
$\sigma_f = 1$	β_0	1.000	1.007	0.048	0.035
	β_1	0.020	0.003	0.010	0.009
	$\log(\lambda)$	-2.303	-1.549	0.179	0.009
	α	1.200	0.705	0.124	0.009
	σ	0.600	0.598	0.025	0.045
	σ_b	0.640	0.637	0.032	0.026

Appendix C

Source Code

C.1 Analysing the Bipolar Data Set

```
#Bipolar Analysis Code

library(magrittr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(survival)
library(survminer)
library(cowplot)
library(JM)
library(nlme)


### Data clean up code ------------------------------------------------------------

data = read.csv(file.choose())
data = data %>% filter(!is.na(HAMAAGE_1))  # Remove 32 subjects that do not have any HAM-A
    scores (305 to 273 subjects).
dataO1B = data[,c(1:47)] # Remove unnecessary columns not relevant to outcome 1B and HAM-A
    scores.

# Make censoring time the same as the last interview if outcome was not experienced.
for(i in 1:nrow(dataO1B)){
  if(is.na(dataO1B$OUTCOME1BAGE[i])){
    dataO1B$OUTCOME1BAGE[i] <- dataO1B$AGELASTINT[i]
  }
}

# Remove all HAMA values observed past the age of Outcome1B
dataO1B$boolean = dataO1B$OUTCOME1BAGE >= dataO1B$HAMAAGE_1 #If false, then earliest HAMAAGE
    _1 is measured after experienced event.
```

```
dataO1B$recode = ifelse(dataO1B$boolean=="FALSE",0,1) #If true, assigned to be 1.

# Filter data to only include the 1 which indicates that at least one HAM-A measurement is
    less than the event time.
dataO1B = dataO1B %>% filter(recode == 1) #From 273 subjects to 207.

# Clean up HAM-A values that are greater than the age that patient experienced event.
cols = seq(19, 47, by=2) #Columns with HAMAAGE
for (j in cols){
  dataO1B$booltemp = dataO1B[,10] > dataO1B[,j] #Compares the HAMAAGE with the event
      /censoring time.
  dataO1B$retemp = ifelse(dataO1B$booltemp =="FALSE",0,1) #If False, then jth HAMAAGE is
      greater than the event/censoring time.
  dataO1B$retemp = replace_na(dataO1B$retemp, 1)
  for (i in 1:nrow(dataO1B)){ #Make HAMAAGE and HAMATOT NA
    if (dataO1B$retemp[i]==0){
      dataO1B[i,j] <- NA #Nullify age of HAM-A NA since it is past the event time.
      dataO1B[i, j-1] <- NA #Nullify value of HAM-A
    }
  }
}


### Summary statistics ------------------------------------------------------------------

cluscount = dataO1B %>% count(FAMILYID) #Count of clusters.
summary(cluscount$n)

histcluster = ggplot(cluscount, aes(x=n)) +
  geom_histogram(color="black", fill="lightskyblue2", binwidth=1) +
  labs(x="Cluster size", y="Frequency") + theme_bw()
histcluster + scale_x_continuous(breaks = seq(0,9,1)) # Histogram of number of kids per
    family

dataO1B %>% count(SEX) #Males=113, Females=94
dataO1B %>% count(PARENTSES_1) #Hollingshead scale for socioeconomic status
dataO1B %>% count(LITHRESP) #0=LiNR/LiNR-profile, 1=LiR/LiR-profile
dataO1B %>% count(OUTCOME1B) #1=Yes, 0=No, was the patient diagnosed with BD, MDD

summary(dataO1B$PARONSAGE); sd(dataO1B$PARONSAGE, na.rm=TRUE) #Summary of parental onset age
summary(dataO1B$AGEFIRSTINT); sd(dataO1B$AGEFIRSTINT) #Summary of age of first interview
summary(dataO1B$AGELASTINT); sd(dataO1B$AGELASTINT) #Summary of age of last interview
summary(dataO1B$AGELASTINT-dataO1B$AGEFIRSTINT); sd(dataO1B$AGELASTINT-dataO1B$AGEFIRSTINT)
    #Summary of length of time in study

HAMAages = dataO1B[, c(5, 16, seq(19, 47, by=2))] #Columns of HAMAAGE
HAMAvals = dataO1B[, c(5, 17, seq(18, 46, by=2))] #Columns of HAMA scores

HAMAvals$na_count <- apply(HAMAvals[,c(2:17)], 1, function(x) sum(is.na(x))) #Number of NA
    scores for each participant
HAMAvals$no_values <- 16-HAMAvals$na_count #Count of HAMA scores observed per subject
summary(HAMAvals$no_values)
```

```
sd(HAMAvals$no_values)

HAMAvalues.long = HAMAvals %>% #Data frame with ID, HAMA score number, and HAMA score as
    columns.
  pivot_longer(cols = starts_with("HAMATOT_"), names_to = "HAMA_score_number",
               names_prefix = "HAMATOT_", values_to = "score", values_drop_na = TRUE)
summary(HAMAvalues.long$score); sd(HAMAvalues.long$score)

# Add the transformed HAMA score as a column
HAMAvalues.long = HAMAvalues.long %>% mutate(logHAMAscore = log(score+1)) #Add one since the
    lowest possible score is 0.
summary(HAMAvalues.long$logHAMAscore); sd(HAMAvalues.long$logHAMAscore)

# HAMA and Transformed HAMA histogram
histHAMA = ggplot(HAMAvalues.long, aes(x=score)) +
  geom_histogram(color="black", fill="lightskyblue2", binwidth=1) +
  labs(x="HAM-A_score", y="Frequency") + theme_bw()
p1 = histHAMA + scale_x_continuous(breaks = seq(0,40,5))

histlogHAMA = ggplot(HAMAvalues.long, aes(x=logHAMAscore)) +
  geom_histogram(color="black", fill="#a26769", binwidth=0.25) +
  labs(x="Transformed_HAM-A_score", y="Frequency") + theme_bw()
p2 = histlogHAMA + scale_x_continuous(breaks = seq(0,4,0.5))

plot_grid(p1, p2, labels="AUTO")

HAMAages.long = HAMAages %>%
  pivot_longer(cols = starts_with("HAMAAGE_"), names_to = "HAMA_age",
               names_prefix = "HAMAAGE_", values_to = "age", values_drop_na = TRUE)
HAMAages.long$ID == HAMAvalues.long$ID

HAMAlong <- cbind(HAMAvalues.long$ID, HAMAvalues.long$logHAMAscore, HAMAages.long$age,
    HAMAvalues.long$score)
colnames(HAMAlong) <- c("ID", "logHAMA", "Age", "Score")
HAMAlong = data.frame(HAMAlong)

# Spaghetti plots
spagplot = ggplot(HAMAlong, aes(x=Age, y=logHAMA, color=factor(ID))) +
  geom_line(show.legend = FALSE) + theme_bw() +
  labs(y="Transformed_HAM-A_scores") + theme(legend.position = "none")
spagplot + stat_smooth(aes(group=1), se=FALSE)

spagplot2 = ggplot(HAMAlong, aes(x=Age, y=Score, color=factor(ID))) +
  geom_line(show.legend = FALSE) + theme_bw() +
  labs(y="HAM-A_scores") + theme(legend.position = "none")
spagplot2 + stat_smooth(aes(group=1), se=FALSE)

plot_grid(spagplot2 + stat_smooth(aes(group=1), se=FALSE), spagplot + stat_smooth(aes(group
    =1), se=FALSE), labels="AUTO")

# Making a long version of data frame with the 207 subjects
t = dataO1B[,c(1:5, 7, 10, 12:47)]
```

```
t2 = t %>%
  pivot_longer(
    cols = starts_with("HAMATOT_"),
    names_to = "Measure_num",
    names_prefix = "HAMATOT_",
    values_to = "Score",
    values_drop_na = TRUE
  )
t2$Age = HAMAlong$Age
t2$logScore = HAMAlong$logHAMA

# Fully cleaned data ready to be fit
bd.data = t2[,-c(12:27)]

bd.data$PARENTSES_1 = recode(bd.data$PARENTSES_1, "5"=1, "4"=2, "3"=3, "2"=3, "1"=3) #Recode
    SES
bd.data.id = bd.data[!duplicated(bd.data$ID),]


### Fitting the joint model ----------------------------------------------------

fitLME.int.SEX = lme(logScore ~ factor(SEX) + Age, random = ~1|ID, data=bd.data)
fitSURV.SES = coxph(Surv(OUTCOME1BAGE, OUTCOME1B) ~ factor(PARENTSES_1), data = bd.data.id,
    x = TRUE)
fitJM.int.SES = jointModel(fitLME.int.SEX, fitSURV.SES, timeVar = "Age",
                           method = "weibull-PH-GH")

### Defining a function to fit the joint model ----
theta = function(data){
  ctrl.lme <- lmeControl(opt="optim", maxIter=100, msMaxIter=100)
  LMEmod = lme(logScore ~ factor(SEX) + Age, random=~1|ID, data=data, control=ctrl.lme)
  data.id = data[!duplicated(data$ID),]
  SURVmod = coxph(Surv(OUTCOME1BAGE, OUTCOME1B) ~ factor(PARENTSES_1), data=data.id, x=TRUE)
  ctrl.JM <- list(tol1 = 0.002, tol2 = 0.0002)
  JMmod = jointModel(LMEmod, SURVmod, timeVar="Age", method="weibull-PH-GH", control=ctrl.JM
      )
  return(JMmod)
}

### Jackknife ------------------------------------------------------------------

bd.data = read.table("/home/gaudetd/Data/bipolar_cleaned.csv", sep=",", header=TRUE)[,-1] ##
    Bring in the bipolar_cleaned data.

bd.data$FamilyNum <- as.integer(x = factor(x = bd.data$FAMILYID)) # Coding the family ID to
    numbers

# Defining a function to calculate the EJK estimate.
sejack = function(vector){
  g = length(vector) #number of jackknife resamples
  theta.mean = mean(vector)
  return(sqrt((g-1)*sum((vector-theta.mean)^2)/g))
```

```
}

# Applying the group jackknife
g=length(unique(bd.data$FamilyNum)) #there will be 98 jackknife leave out a cluster samples
theta.g = theta.g.pseudo = matrix(NA, nrow=g, ncol=9) #B
dimnames(theta.g) = list(c(1:g), c("Alpha", "Intercept", "SEX", "AGE", "Intercept", "SES4",
    "SES123", "Sigma", "D"))
dimnames(theta.g.pseudo) = list(c(1:g), c("Alpha", "Intercept", "SEX", "AGE", "Intercept", "
    SES4", "SES123", "Sigma", "D"))
nvec = numeric() #Empty vector to store the total number of subjects n for the jackknife
    resample.
hvec = numeric() #u_g^-1, the inverse of the proportion of subjects in cluster g
for(i in 1:g){
  bd.data.g = bd.data %>% filter(!FamilyNum == i) #Leave out a cluster sample g
  JM.temp = theta(bd.data.g) #Fit joint model
  nvec[i] = length(unique(bd.data.g$ID)) #jackknife sample size.
  hvec[i] = 207/(207-nvec[i]) #n/removed cluster size.
  theta.g[i, 1] = JM.temp$coefficients$alpha[1]
  theta.g[i, 2] = JM.temp$coefficients$betas[1]
  theta.g[i, 3] = JM.temp$coefficients$betas[2]
  theta.g[i, 4] = JM.temp$coefficients$betas[3]
  theta.g[i, 5] = JM.temp$coefficients$gammas[1]
  theta.g[i, 6] = JM.temp$coefficients$gammas[2]
  theta.g[i, 7] = JM.temp$coefficients$gammas[3]
  theta.g[i, 8] = JM.temp$coefficients$sigma
  theta.g[i, 9] = JM.temp$coefficients$D[1,1]
  #Pseudo-values
  theta.g.pseudo[i, 1] = hvec[i]*1.223869 - (hvec[i]-1)*JM.temp$coefficients$alpha[1]
  theta.g.pseudo[i, 2] = hvec[i]*1.00845653 - (hvec[i]-1)*JM.temp$coefficients$betas[1]
  theta.g.pseudo[i, 3] = hvec[i]*0.20055802 - (hvec[i]-1)*JM.temp$coefficients$betas[2]
  theta.g.pseudo[i, 4] = hvec[i]*0.02238974 - (hvec[i]-1)*JM.temp$coefficients$betas[3]
  theta.g.pseudo[i, 5] = hvec[i]*(-15.02352986) - (hvec[i]-1)*JM.temp$coefficients$gammas[1]
  theta.g.pseudo[i, 6] = hvec[i]*0.05426266 - (hvec[i]-1)*JM.temp$coefficients$gammas[2]
  theta.g.pseudo[i, 7] = hvec[i]*0.90049267 - (hvec[i]-1)*JM.temp$coefficients$gammas[3]
  theta.g.pseudo[i, 8] = hvec[i]*0.6827196 - (hvec[i]-1)*JM.temp$coefficients$sigma
  theta.g.pseudo[i, 9] = hvec[i]*0.353888 - (hvec[i]-1)*JM.temp$coefficients$D[1,1]
}

# Calculate the EJK
se.JK = apply(theta.g, 2, sejack) #Unweighted estimated JK SE.
se.JK

## Calculate the UJK
# Calculating the weighted average, theta.hat_(.) for each theta.
pseudo.jack.alpha = g*1.223869 - sum((1-hvec^(-1))*theta.g[,1])
pseudo.jack.beta1 = g*1.00845653 - sum((1-hvec^(-1))*theta.g[,2])
pseudo.jack.beta2 = g*0.20055802 - sum((1-hvec^(-1))*theta.g[,3])
pseudo.jack.beta3 = g*0.02238974 - sum((1-hvec^(-1))*theta.g[,4])
pseudo.jack.gamma1 = g*(-15.02352986) - sum((1-hvec^(-1))*theta.g[,5])
pseudo.jack.gamma2 = g*0.05426266 - sum((1-hvec^(-1))*theta.g[,6])
pseudo.jack.gamma3 = g*0.90049267 - sum((1-hvec^(-1))*theta.g[,7])
pseudo.jack.sigma = g*0.6827196 - sum((1-hvec^(-1))*theta.g[,8])
```

```
pseudo.jack.D = g*0.353888 - sum((1-hvec^(-1))*theta.g[,9])

# Delete mj jackknife using the pseudo values
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,1]-pseudo.jack.alpha)^2)))) #Association
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,2]-pseudo.jack.beta1)^2)))) #Long intercept
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,3]-pseudo.jack.beta2)^2)))) #SEX
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,4]-pseudo.jack.beta3)^2)))) #AGE
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,5]-pseudo.jack.gamma1)^2)))) #Surv intercept
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,6]-pseudo.jack.gamma2)^2)))) #SES4
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,7]-pseudo.jack.gamma3)^2)))) #SES123
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,8]-pseudo.jack.sigma)^2)))) #Sigma
sqrt((1/g)*(sum((1/((hvec)-1))*(theta.g.pseudo[,9]-pseudo.jack.D)^2)))) #D


### Bootstrap ------------------------------------------------------------------

bd.data = read.table("/home/gaudetd/Data/bipolar_cleaned.csv", sep=",", header=TRUE)[,-1]

bd.data$FamilyNum <- as.integer(x = factor(x = bd.data$FAMILYID)) # Coding the family ID to
    numbers

# Defining a function to calculate the BS and WBS estimate.
seboot = function(vector){
  b = length(vector) #number of bootstrap samples, should be 200
  theta.mean = mean(vector)
  return(sqrt((1)*sum((vector-theta.mean)^2)/(b-1)))
}

# Applying the group bootstrap
set.seed(123)
B = 200 #Number of samples required to estimate the standard error.
G = length(unique(bd.data$FamilyNum))
count = 0 #Counting the number of total bootstrap samples
b = 0 #Count of the convergent bootstrap samples
theta.b = theta.b.weighted = matrix(NA, nrow=B, ncol=9)
dimnames(theta.b) = list(c(1:B), c("Alpha", "Intercept", "SEX", "AGE", "Intercept", "SES4",
    "SES123", "Sigma", "D"))
dimnames(theta.b.weighted) = list(c(1:B), c("Alpha", "Intercept", "SEX", "AGE", "Intercept",
     "SES4", "SES123", "Sigma", "D"))
while(b < 200){
  count = count + 1
  cat(paste("iteration",count,"\n"))
  index = sample(1:G, size=G, replace=TRUE) #Randomly selects G clusters with replacement
  aa = table(index) #Makes a contingency table of the randomly selected clusters
  boot.data = NULL
  for(i in 1:max(aa)){
    bb = bd.data[bd.data$FamilyNum %in% names(aa[aa %in% i]),] #Finds which clusters were
        sampled i times
    for(j in 1:i){ #Adds to data.frame i times. First time keeps normal ID, 2nd time adds
        1.9/2, so forth till hits i.
      cc = bb
      if(j >=2){
```

```
        cc$ID = bb$ID + 1.9/j
      }
      boot.data = rbind(boot.data, cc)
    }
  }
  nstar = length(unique(boot.data$ID)) #Bootstrap sample size.
  possibleError = tryCatch(theta(boot.data), error=function(e) e) #Fit joint model and
      stores error if present.
  if(!inherits(possibleError, "error")){ #If there is no error generated, fit and record
      parameter estimates.
    b = b + 1
    JM.temp = theta(boot.data)
    theta.b[b, 1] = JM.temp$coefficients$alpha[1]
    theta.b[b, 2] = JM.temp$coefficients$betas[1]
    theta.b[b, 3] = JM.temp$coefficients$betas[2]
    theta.b[b, 4] = JM.temp$coefficients$betas[3]
    theta.b[b, 5] = JM.temp$coefficients$gammas[1]
    theta.b[b, 6] = JM.temp$coefficients$gammas[2]
    theta.b[b, 7] = JM.temp$coefficients$gammas[3]
    theta.b[b, 8] = JM.temp$coefficients$sigma #sigma
    theta.b[b, 9] = JM.temp$coefficients$D[1,1] #sigma_b^2
    #Weighted estimates. Estimates are weighted by the sqrt(bootstrap sample size/original
        sample size)
    theta.b.weighted[b, 1] = sqrt(nstar/207)*JM.temp$coefficients$alpha[1]
    theta.b.weighted[b, 2] = sqrt(nstar/207)*JM.temp$coefficients$betas[1]
    theta.b.weighted[b, 3] = sqrt(nstar/207)*JM.temp$coefficients$betas[2]
    theta.b.weighted[b, 4] = sqrt(nstar/207)*JM.temp$coefficients$betas[3]
    theta.b.weighted[b, 5] = sqrt(nstar/207)*JM.temp$coefficients$gammas[1]
    theta.b.weighted[b, 6] = sqrt(nstar/207)*JM.temp$coefficients$gammas[2]
    theta.b.weighted[b, 7] = sqrt(nstar/207)*JM.temp$coefficients$gammas[3]
    theta.b.weighted[b, 8] = sqrt(nstar/207)*JM.temp$coefficients$sigma
    theta.b.weighted[b, 9] = sqrt(nstar/207)*JM.temp$coefficients$D[1,1]
  }
}

count #Print the total number of bootstrap resamples.
theta.b

#BS estimates
se.BS = apply(theta.b, 2, seboot) #Unweighted estimated BS SE.
se.BS

#WBS estimates
se.BS.weighted = apply(theta.b.weighted, 2, seboot) #Weighted method.
se.BS.weighted
```

C.2 Analysing the Baboon Data Set

```
library(magrittr)
```

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(survival)
library(cowplot)
library(JM)
library(nlme)
library(viridis)

### Cleaning the data ------------------------------------------------------------
longdata = read.csv(file.choose())
longdata$number = c(1:nrow(longdata))
dup = longdata %>% group_by(id_code, age_ad) %>% filter(n()>1) #Returns all duplicated age
    _ads
dup.half = dup %>% slice_min(number) #Takes only the first measurement.
rm.numbers = dup$number
dup.rm = rm.numbers[-which(rm.numbers %in% dup.half$number)]
df.long = longdata %>% slice(-c(dup.rm)) #Removes the measurements that are at the same time
     point except for this first one.

n = length(unique(df.long$id_code)) #Sample size is unchanged = 242

#Survival data
eventdata = read.csv(file.choose())

#survival events from 242 baboons
survdata = eventdata %>%
  group_by(id_code) %>%
  slice_max(age_end_ad) #End of life

#The highest age at which gc was measured.
highagemeasure = df.long %>%
  group_by(id_code) %>%
  slice_max(age_ad)

#The lowest age at which gc was measured.
lowagemeasure = df.long %>%
  group_by(id_code) %>%
  slice_min(age_ad)  #Some baboons have alpha status upon enrollment into the study.

#Verify that all measurements are prior to survival/censoring time.
logic = survdata$age_end_ad > highagemeasure$age_ad #All survival/censoring times greater
    than measurements

#Clustered based on initial group.
lowagemeasure$GrpNum <- as.integer(x = factor(x = lowagemeasure$grp))
temp = cbind(lowagemeasure$id_code, lowagemeasure$GrpNum)
dimnames(temp) = list(c(1:242), c("id_code", "GrpNum"))


### Summary statistics -----------------------------------------------------------
```

```
#Summary of the event/censoring times
survdata %>% ungroup() %>% count(death)
summary(survdata$age_end_ad)

#Baseline DSI_M and DSI_F,
earlysurvmeasure = eventdata %>%
  group_by(id_code) %>%
  slice_min(age_end_ad)
summary(earlysurvmeasure)

#Baseline proportional rank, make a data frame of their baseline measurements and their
    survival times and outcome (0 vs. 1)
dat.mat = cbind(c(1:242), earlysurvmeasure$DSI_F, earlysurvmeasure$DSI_M, lowagemeasure
    $proportional_rank, survdata$age_end_ad, survdata$death)
dimnames(dat.mat) = list(c(1:242), c("id_code", "DSI_F", "DSI_M", "prop_rank", "surv_time",
    "death"))

df.full = merge(df.long, dat.mat, by="id_code")

#Clustered to the initial group
lowagemeasure$GrpNum <- as.integer(x = factor(x = lowagemeasure$grp))
temp = cbind(lowagemeasure$id_code, lowagemeasure$GrpNum)
dimnames(temp) = list(c(1:242), c("id_code", "GrpNum"))

data.full = merge(df.full, temp, by="id_code") #Added the code for cluster.


### Figure source code ---------------------------------------------------------

### Histograms
histGC = ggplot(df.long, aes(x=gc)) +
  geom_histogram(color="black", fill="#582c4d", bins = 30) +
  labs(x="fGC", y="Frequency") + theme_bw()
histGC

histlogGC = ggplot(df.long, aes(x=ln_gc)) +
  geom_histogram(color="black", fill="#004643", bins=30) +
  labs(x="log(fGC)", y="Frequency") + theme_bw()
histlogGC

plot_grid(histGC, histlogGC, labels="AUTO")

### Spaghetti plot
SpagPlotLn = ggplot(df.long, aes(x=age_ad, y=ln_gc, color=factor(id_code))) +   #group=id
    _code,
  geom_line(show.legend = FALSE) + labs(y="Log(Fecal glucocorticoid)", x="Age past adulthood
      ")

SpagPlotLn + scale_color_viridis(discrete=TRUE, option="magma") + theme_bw()


### Fitting the joint model
```

```
LMEbab.slope = lme(ln_gc ~ DSI_F + age_ad, random = ~age_ad|id_code, data=df.full)
SURVbab.fem = coxph(Surv(surv_time, death) ~ DSI_F, data = df.full.id, x = TRUE)
JMbab.fem = jointModel(LMEbab.slope, SURVbab.fem, timeVar = "age_ad",
                       method = "weibull-PH-GH", verbose=TRUE)

## Defining  a function to fit this joint model
theta = function(data){
  ctrl.lme <- lmeControl(opt="optim", maxIter=100, msMaxIter=100)
  LMEmod = lme(ln_gc ~ DSI_F + age_ad, random=~age_ad|id_code, data=data, control=ctrl.lme)
  data.id = data[!duplicated(data$id_code),]
  SURVmod = coxph(Surv(surv_time, death) ~ DSI_F, data=data.id, x=TRUE)
  ctrl.JM <- list(tol1 = 0.002, tol2 = 0.0002)
  JMmod = jointModel(LMEmod, SURVmod, timeVar="age_ad", method="weibull-PH-GH", control=ctrl
      .JM)
  return(JMmod)
}

### Jackknife ------------------------------------------------------------------
bab.data = read.table("/home/gaudetd/Data/baboon_cleaned.csv", sep=",", header=TRUE)[,-1]

#Equal cluster size jackknife (EJK)
sejack = function(vector){
  g = length(vector) #number of jackknife samples
  theta.mean = mean(vector)
  return(sqrt((g-1)*sum((vector-theta.mean)^2)/g))
}

g=length(unique(bab.data$GrpNum)) #there will be 13 jackknife samples
theta.g = theta.g.pseudo = matrix(NA, nrow=g, ncol=6)
dimnames(theta.g) = list(c(1:g), c("Alpha", "Intercept", "L_DSI_F", "AGE", "Intercept", "S
    _DSI_F"))
dimnames(theta.g.pseudo) = list(c(1:g), c("Alpha", "Intercept", "L_DSI_F", "AGE", "Intercept
    ", "S_DSI_F"))
nvec = numeric()
hvec = numeric()
for(i in 1:g){
  bab.data.g = bab.data %>% filter(!GrpNum == i) #Leave out a cluster sample g
  possibleError = tryCatch(theta(bab.data.g), error=function(e) e)
  if(!inherits(possibleError, "error")){ #If the model fits correctly, then save parameter
      estimates
    JM.temp = theta(bab.data.g) #Fit joint model
    nvec[i] = length(unique(bab.data.g$id_code)) #temp variable for jackknife asmple size.
    hvec[i] = 242/(242-nvec[i]) #n/removed cluster size.
    theta.g[i, 1] = JM.temp$coefficients$alpha[1]
    theta.g[i, 2] = JM.temp$coefficients$betas[1]
    theta.g[i, 3] = JM.temp$coefficients$betas[2]
    theta.g[i, 4] = JM.temp$coefficients$betas[3]
    theta.g[i, 5] = JM.temp$coefficients$gammas[1]
    theta.g[i, 6] = JM.temp$coefficients$gammas[2]
    #Pseudo-values
    theta.g.pseudo[i, 1] = hvec[i]*1.232295 - (hvec[i]-1)*JM.temp$coefficients$alpha[1]
```

```
    theta.g.pseudo[i, 2] = hvec[i]*4.04656223 - (hvec[i]-1)*JM.temp$coefficients$betas[1]
    theta.g.pseudo[i, 3] = hvec[i]*0.08603379 - (hvec[i]-1)*JM.temp$coefficients$betas[2]
    theta.g.pseudo[i, 4] = hvec[i]*0.02366949 - (hvec[i]-1)*JM.temp$coefficients$betas[3]
    theta.g.pseudo[i, 5] = hvec[i]*(-9.7113541) - (hvec[i]-1)*JM.temp$coefficients$gammas[1]
    theta.g.pseudo[i, 6] = hvec[i]*(-0.2737551) - (hvec[i]-1)*JM.temp$coefficients$gammas[2]
  }
  else { #if error is thrown, save estimates as 0.
    theta.g[i, 1] = 0
    theta.g[i, 2] = 0
    theta.g[i, 3] = 0
    theta.g[i, 4] = 0
    theta.g[i, 5] = 0
    theta.g[i, 6] = 0
    #Pseudo-values
    theta.g.pseudo[i, 1] = 0
    theta.g.pseudo[i, 2] = 0
    theta.g.pseudo[i, 3] = 0
    theta.g.pseudo[i, 4] = 0
    theta.g.pseudo[i, 5] = 0
    theta.g.pseudo[i, 6] = 0
  }
}

#EJK
se.JK = apply(theta.g, 2, sejack) #Unweighted estimated JK SE.
se.JK

# Average pseudo-values.
pseudo.jack.alpha = g*1.232295 - sum((1-hvec^(-1))*theta.g[,1])
pseudo.jack.beta1 = g*4.04656223 - sum((1-hvec^(-1))*theta.g[,2])
pseudo.jack.beta2 = g*0.08603370 - sum((1-hvec^(-1))*theta.g[,3])
pseudo.jack.beta3 = g*0.02366949 - sum((1-hvec^(-1))*theta.g[,4])
pseudo.jack.gamma1 = g*(-9.7113541) - sum((1-hvec^(-1))*theta.g[,5])
pseudo.jack.gamma2 = g*(-0.2737551) - sum((1-hvec^(-1))*theta.g[,6])

# Delete mj jackknife using the pseudo values
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,1]-pseudo.jack.alpha)^2)))) #Association
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,2]-pseudo.jack.beta1)^2)))) #Long intercept
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,3]-pseudo.jack.beta2)^2)))) #DSIF
    longitudinal
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,4]-pseudo.jack.beta3)^2)))) #AGE
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,5]-pseudo.jack.gamma1)^2)))) #Surv intercept
sqrt((1/g)*(sum((1/((hvec)-1)*(theta.g.pseudo[,6]-pseudo.jack.gamma2)^2)))) #DSIF survival

### Bootstrap ------------------------------------------------------------------

bab.data = read.table("/home/gaudetd/Data/baboon_cleaned.csv", sep=",", header=TRUE)[,-1]

# Define function to calculate BS and WBS estimates - same as used for bipolar data set
seboot = function(vector){
  b = length(vector) #number of bootstrap samples, b=200
  theta.mean = mean(vector)
```

```
  return(sqrt((1)*sum((vector-theta.mean)^2)/(b-1)))
}

set.seed(123)
B = 200 #Number of samples required to estimate the standard error.
G = length(unique(bab.data$GrpNum)) #13 clusters
count = 0
b = 0
theta.b = theta.b.weighted = matrix(NA, nrow=B, ncol=6)
dimnames(theta.b) = list(c(1:B), c("Alpha", "Intercept", "L_DSI_F", "AGE", "Intercept", "S
    _DSI_F"))
dimnames(theta.b.weighted) = list(c(1:B), c("Alpha", "Intercept", "L_DSI_F", "AGE", "
    Intercept", "S_DSI_F"))
while(b < 200){
  cat(paste("iteration",b,"\n"))
  count = count + 1
  index = sample(1:G, size=G, replace=TRUE)
  aa = table(index)
  boot.data = NULL
  for(i in 1:max(aa)){
    bb = bab.data[bab.data$GrpNum %in% names(aa[aa %in% i]),] #Finds which clusters were
        sampled i times
    for(j in 1:i){ #Adds to data.frame i times. First time keeps normal ID, 2nd time adds
        1.9/2, so forth till hits i.
      cc = bb
      if(j >=2){
        cc$id_code = bb$id_code + 1.9/j
      }
      boot.data = rbind(boot.data, cc)
    }
  }
  nstar = length(unique(boot.data$id_code)) #Bootstrap sample size.
  possibleError = tryCatch(theta(boot.data), error=function(e) e) #Fit joint model, this
      might make the
  if(!inherits(possibleError, "error")){
    JM.temp = theta(boot.data)
    b = b + 1
    theta.b[b, 1] = JM.temp$coefficients$alpha[1]
    theta.b[b, 2] = JM.temp$coefficients$betas[1]
    theta.b[b, 3] = JM.temp$coefficients$betas[2]
    theta.b[b, 4] = JM.temp$coefficients$betas[3]
    theta.b[b, 5] = JM.temp$coefficients$gammas[1]
    theta.b[b, 6] = JM.temp$coefficients$gammas[2]
    #Weighted estimates
    theta.b.weighted[b, 1] = sqrt(nstar/242)*JM.temp$coefficients$alpha[1]
    theta.b.weighted[b, 2] = sqrt(nstar/242)*JM.temp$coefficients$betas[1]
    theta.b.weighted[b, 3] = sqrt(nstar/242)*JM.temp$coefficients$betas[2]
    theta.b.weighted[b, 4] = sqrt(nstar/242)*JM.temp$coefficients$betas[3]
    theta.b.weighted[b, 5] = sqrt(nstar/242)*JM.temp$coefficients$gammas[1]
    theta.b.weighted[b, 6] = sqrt(nstar/242)*JM.temp$coefficients$gammas[2]
  }
}
```

```
theta.b
theta.b.weighted
count

#Unweighted bootstrap.
se.BS = apply(theta.b, 2, seboot) #Unweighted estimated BS SE.
se.BS

#Weighted bootstrap
se.BS.weighted = apply(theta.b.weighted, 2, seboot)
se.BS.weighted
```

C.3 Simulation Code

Some of this source code derives itself from the source code of the analyses from Stefan (2019) and Lowe (2020).

```
library(magrittr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(survival)
library(survminer)
library(cowplot)
library(JM)
library(nlme)


### Initializing parameters ----------------------------------------------------
ITER = 100 #Generate 100 data sets
g = 100 #Number of clusters
ng = 5 #Number of individuals per cluster (500 individuals in study)
ngi = 4 #Number of observations per subject in cluster g
beta0 = 1 #longitudinal intercept
beta1 = 0.02 #slope
sigf = 1 #SD for cluster effects, sigf = (0.001, 0.5, 1)
sigb = 0.64 #SD for random effect b
sig = 0.6 #SD for individual subject
alpha = 1.2 #association parameter
lambda = 0.1 #baseline hazard, gamma = log(lambda) = -2.302585

fg = rep(0, g) #Cluster-specific random effects (1 x no. clusters)
bgi = rep(0, g*ng) #subject specific random effects (1 x no. individuals)
egij = rep(0, g*ng*ngi) #Measurement specific effect (1 x no. measurements)
ygij = rep(0, g*ng*ngi) #observed longitudinal covariate (1 x no. measurements)


### Define a function to generate joint data with clustered outcomes -----------
JMsimdata = function(i) {
```

```
  time = rep(seq(0,ngi-1, by=1), g*ng) #set up the timepoints for observed measures
  df = data.frame(subject = sort(rep(c(1:(g*ng)), ngi)), time = time, Family = sort(rep(c(1:
      g), ng*ngi))) #set up data frame
  fg = rep(rnorm(g, 0, sigf), rep(ng*ngi, g)) #Cluster specific random effects
  bgi = rep(rnorm(g*ng, 0, sigb), rep(ngi, g*ng)) #subj-specific random effects
  egij = rnorm(g*ng, 0, sig) #Measurement-spec random effects
  ygij = beta0 + beta1*time + bgi + egij + fg
  df$response = ygij
  ### Generate event times. Include family??? ###
  eventtime = (1/(beta1*alpha)) * log(1-beta1*alpha*log(runif(g*ng,0,1))/(lambda*exp(alpha*(
      beta0+unique(bgi)+rep(unique(fg), rep(ng, g))))))
  df$eventtime = rep(eventtime, rep(ngi, g*ng))
  df$event = rep(1, g*ng*ngi)
  df = filter(df, time < eventtime) #removes all longitudinal times taken after the event
      time.
  #df.id = df[!duplicated(df$subject),]
  #simdata = list(df)
}

### Generate the data ------------------------------------------------------------------
set.seed(123) # 10 lists of 100 data frames for a total of 1000 datasets
simdata1 = lapply(1:ITER, JMsimdata)
simdata2 = lapply(1:ITER, JMsimdata)
simdata3 = lapply(1:ITER, JMsimdata)
simdata4 = lapply(1:ITER, JMsimdata)
simdata5 = lapply(1:ITER, JMsimdata)
simdata6 = lapply(1:ITER, JMsimdata)
simdata7 = lapply(1:ITER, JMsimdata)
simdata8 = lapply(1:ITER, JMsimdata)
simdata9 = lapply(1:ITER, JMsimdata)
simdata10 = lapply(1:ITER, JMsimdata)


### Define function to fit a joint model -----------------------------------------------
## jointFit returns JMmod for the jackknife and bootstrap resamples, however returns est
    _parm when fitting the joint models for the full data set
jointFit = function(data){
  LMEmod = lme(response ~ time, random=~1|subject, data=data) #control=ctrl.lme
  data.id = data[!duplicated(data$subject),]
  SURVmod = coxph(Surv(eventtime, event) ~ 1, data=data.id, x=TRUE)
  JMmod = jointModel(LMEmod, SURVmod, timeVar="time", method="weibull-PH-GH") #, init=start)
      #, control=ctrl.JM
  #est_parm = list(JMmod$coefficients$betas[1], sqrt(solve(JMmod$Hessian))[1,1], #beta0
  #                JMmod$coefficients$betas[2], sqrt(solve(JMmod$Hessian))[2,2], #beta1
  #                JMmod$coefficients$sigma, sqrt(solve(JMmod$Hessian))[3,3], #Sigma
  #                sqrt(JMmod$coefficients$D[1,1]), sqrt(solve(JMmod$Hessian))[7,7], #Random
      -intercept sigmab
  #                JMmod$coefficients$alpha, sqrt(solve(JMmod$Hessian))[5,5], #alpha
  #                JMmod$coefficients$gammas, sqrt(solve(JMmod$Hessian))[4,4], as.integer(
      JMmod$convergence))
  return(JMmod)
}
```

```
### Define a function to jackknife the iteration ------------------------------------
jointJK = function(data) {
  g = length(unique(data$Family)) #this is the number of samples that will be taken --
      should be 100.
  est_parm = list()
  for(j in 1:g){
    data.g = data %>% filter(!Family == j) #Leave out a cluster sample g
    JM.temp = jointFit(data.g) #Fit joint model and save variables.
    est_parm[[j]] = c(JM.temp$coefficients$betas[1], sqrt(solve(JM.temp$Hessian))[1,1],
                      JM.temp$coefficients$betas[2], sqrt(solve(JM.temp$Hessian))[2,2],
                      JM.temp$coefficients$sigma, sqrt(solve(JM.temp$Hessian))[3,3],
                      sqrt(JM.temp$coefficients$D[1,1]), sqrt(solve(JM.temp$Hessian))[7,7],
                      JM.temp$coefficients$alpha, sqrt(solve(JM.temp$Hessian))[5,5],
                      JM.temp$coefficients$gammas, sqrt(solve(JM.temp$Hessian))[4,4],
                      length(unique(data.g$subject)), as.integer(JM.temp$convergence))
  }
  return(est_parm)
}

### Define a function to bootstrap the iteration ------------------------------------
jointBS = function(data) {
  #Number of samples required to estimate the standard error is 200 bootstrap samples.
  G = length(unique(data$Family)) #Number of clusters chosen for each bootstrap sample. I.e.
       100 clusters are sampled with replacement.
  est_parm = list()
  count = 0
  b = 0
  while(b < 200){
    count = count + 1
    index = sample(1:G, size=G, replace=TRUE)
    aa = table(index)
    boot.data = NULL
    for(i in 1:max(aa)){
      bb = data[data$Family %in% names(aa[aa %in% i]),] #Finds which clusters were sampled i
           times
      for(j in 1:i){ #Adds to data.frame i times. First time keeps normal ID, 2nd time adds
          1.9/2, so forth till hits i.
        cc = bb
        if(j >=2){
          cc$subject = bb$subject + 1.9/j
        }
        boot.data = rbind(boot.data, cc)
      }
    }
    nstar = length(unique(boot.data$subject)) #Bootstrap sample size.
    possibleError = tryCatch(jointFit(boot.data), error=function(e) e) #Fit joint model,
        this might make the
    if(!inherits(possibleError, "error")){
      b = b + 1
      JM.temp = jointFit(boot.data)
      #Saving the coefficient and standard error estimates and the overall sample size.
```

```
      est_parm[[b]] = c(JM.temp$coefficients$betas[1], sqrt(solve(JM.temp$Hessian))[1,1],
                        JM.temp$coefficients$betas[2], sqrt(solve(JM.temp$Hessian))[2,2],
                        JM.temp$coefficients$sigma, sqrt(solve(JM.temp$Hessian))[3,3],
                        sqrt(JM.temp$coefficients$D[1,1]), sqrt(solve(JM.temp$Hessian))
                            [7,7],
                        JM.temp$coefficients$alpha, sqrt(solve(JM.temp$Hessian))[5,5],
                        JM.temp$coefficients$gammas, sqrt(solve(JM.temp$Hessian))[4,4],
                        length(unique(boot.data$subject)), as.integer(JM.temp$convergence))
    }
  }
  return(list(est_parm, count))
}

### Example of a script to run the functions ------------------------------------------
## Note: this is just an example of fitting joint models, however the jackknife and
    bootstrap follow a similar pattern, replacing the parLapply() argument for function with
     the respective jackknife and bootstrap functions previously defined.

nodeslist <- unlist(strsplit(Sys.getenv("NODESLIST"), split=" "))

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

i = as.numeric(Sys.getenv("SLURM_ARRAY_TASK_ID")) #

#Define the connection to the Graham environment
ncores = detectCores()
cl = makePSOCKcluster(ncores)

#Note: include the code to the functions that will be used for the script. To fit only the
    joint model, we include the jointFit() function.
jointFit = function(data){
  LMEmod = lme(response ~ time, random=~1|subject, data=data) #control=ctrl.lme
  data.id = data[!duplicated(data$subject),]
  SURVmod = coxph(Surv(eventtime, event) ~ 1, data=data.id, x=TRUE)
  JMmod = jointModel(LMEmod, SURVmod, timeVar="time", method="weibull-PH-GH")
  est_parm = list(JMmod$coefficients$betas[1], sqrt(solve(JMmod$Hessian))[1,1], #beta0
                  JMmod$coefficients$betas[2], sqrt(solve(JMmod$Hessian))[2,2], #beta1
                  JMmod$coefficients$sigma, sqrt(solve(JMmod$Hessian))[3,3], #Sigma
                  sqrt(JMmod$coefficients$D[1,1]), sqrt(solve(JMmod$Hessian))[7,7], #Random-
                      intercept sigmab
                  JMmod$coefficients$alpha, sqrt(solve(JMmod$Hessian))[5,5], #alpha
                  JMmod$coefficients$gammas, sqrt(solve(JMmod$Hessian))[4,4], as.integer(
                      JMmod$convergence))
}

#Import data
Simulations = readRDS(paste("/home/gaudetd/scratch/data/simdata", i, ".RDS", sep=""))

#Export all variables to the cluster. Include all functions, variables needed within each
    cluster.
```

```
clusterExport(cl, varlist=c("jointFit", "Simulations"))

#Export all necessary packages to the cluster
clusterEvalQ(cl, list(attach(loadNamespace("JM"), name = "JM"),
                      attach(loadNamespace("nlme"), name = "nlme"),
                      attach(loadNamespace("survival"), name = "survival"),
                      attach(loadNamespace("parallel"), name = "parallel"),
                      attach(loadNamespace("dplyr"), name = "dplyr"),
                      attach(loadNamespace("magrittr"), name = "magrittr")))

#Set seed at cluster level.
clusterSetRNGStream(cl=cl, 123)

#Fit joint model over all 1000 iterations.
JMfits = parLapply(cl=cl, Simulations, jointFit)

#Free the memory reserved on the cores
stopCluster(cl)

#Save output to own file.
saveRDS(JMfits, file = paste("/scratch/gaudetd/output/JMfits", i, ".RDS", sep=""))


### Example of a clean up code script ------------------------------------------
## This is the code to clean up the fitted joint model results, the jackknife and bootstrap
    use similar code

library(rlist); library(here); library(xtable)

# Get the RDS files holding the results.
here()
results = list()
for(i in 1:10){
  results = c(results, readRDS(here(paste("JMfits",i,".RDS",sep=""))))
}

#Define empty data frame
ests = matrix(NA, nrow=length(results), ncol=13)
colnames(ests) = c("Int", "IntSE", "Time", "TimeSE", "Sig", "SigSE", "SigB", "SigBSE", "
    Alpha", "AlphaSE", "logLambda", "logLambdaSE", "Convergence")

ests[,1] = list.rbind(lapply(1:length(results), function(i){results[[i]][[1]]}))
ests[,2] = list.rbind(lapply(1:length(results), function(i){results[[i]][[2]]}))
ests[,3] = list.rbind(lapply(1:length(results), function(i){results[[i]][[3]]}))
ests[,4] = list.rbind(lapply(1:length(results), function(i){results[[i]][[4]]}))
ests[,5] = list.rbind(lapply(1:length(results), function(i){results[[i]][[5]]}))
ests[,6] = list.rbind(lapply(1:length(results), function(i){results[[i]][[6]]}))
ests[,7] = list.rbind(lapply(1:length(results), function(i){results[[i]][[7]]}))
ests[,8] = list.rbind(lapply(1:length(results), function(i){results[[i]][[8]]}))
ests[,9] = list.rbind(lapply(1:length(results), function(i){results[[i]][[9]]}))
ests[,10] = list.rbind(lapply(1:length(results), function(i){results[[i]][[10]]}))
ests[,11] = list.rbind(lapply(1:length(results), function(i){results[[i]][[11]]}))
```

```
ests[,12] = list.rbind(lapply(1:length(results), function(i){results[[i]][[12]]}))
ests[,13] = list.rbind(lapply(1:length(results), function(i){results[[i]][[13]]}))

#Making the summary for the joint model fits, the summary for the
sumtab = matrix(NA, nrow=6, ncol=4)
colnames(sumtab) = c("True value", "Avg. est", "SD", "Avg. SE")
rownames(sumtab) = c("beta0", "beta1", "log(lambda)", "alpha", "sigma", "sigmab")
sumtab[,1] = c(1, 0.02, -2.302585, 1.2, 0.6, 0.64)
sumtab[,2] = apply(ests[,c(1,3,11,9,5,7)], 2, mean)
sumtab[,3] = apply(ests[,c(1,3,11,9,5,7)], 2, sd) #empirical sd
sumtab[,4] = apply(ests[,c(2,4,12,10,6,8)], 2, mean) #mean SE

xtable(sumtab, caption=c("Coefficient estimates for 1000 iterations"),
       align=c(c,c,c,c,c), digits=c(3,3,3,3,3))


#The data cleaning for the jackknife and bootstrap is mildly different.
#For the purposes of the jackknife and bootstrap, the code was cleaned into 1000 matrices
    with the jackknife or bootstrap results for each resample.
#the bootres results can be used interchangeably, except each temp.mat matrix has 200 rows
    and 14 columns.
for(i in 1:length(jackres)){ #1:1000 for each generated data set. jackres is the results of
    the jackknife
  temp.mat = matrix(NA, nrow=100, ncol=14)
  for(j in 1:100){ #For each of the jackknife samples in the 1000 iterations
    temp.mat[j,1] = jackres[[i]][[j]][[1]] #Beta0
    temp.mat[j,2] = jackres[[i]][[j]][[2]] #Beta0 SE
    temp.mat[j,3] = jackres[[i]][[j]][[3]] #Beta1
    temp.mat[j,4] = jackres[[i]][[j]][[4]] #Beta1 SE
    temp.mat[j,5] = jackres[[i]][[j]][[5]] #Sig
    temp.mat[j,6] = jackres[[i]][[j]][[6]] #Sig SE
    temp.mat[j,7] = jackres[[i]][[j]][[7]] #SigB
    temp.mat[j,8] = jackres[[i]][[j]][[8]] #SigB SE
    temp.mat[j,9] = jackres[[i]][[j]][[9]] #Alpha
    temp.mat[j,10] = jackres[[i]][[j]][[10]] #Alpha SE
    temp.mat[j,11] = jackres[[i]][[j]][[11]] #logLambda
    temp.mat[j,12] = jackres[[i]][[j]][[12]] #logLambda SE
    temp.mat[j,13] = jackres[[i]][[j]][[13]] #Sample size
    temp.mat[j,14] = jackres[[i]][[j]][[14]] #COnvergence
  }
  jackests[[i]] = temp.mat
}

#Simulation EJK code
EJK = function(data){
  thetabar = mean(data)
  G = 100 #Established that there are 100 clusters.
  output = sqrt(((98)/100)*sum((data-thetabar)^2))
  return(output)
}

for(i in 1:length(jackests)){
```

```
  mat.results[i,1] = EJK(jackests[[i]][,1]) #beta0
  mat.results[i,2] = mean(jackests[[i]][,2])
  mat.results[i,3] = EJK(jackests[[i]][,3]) #beta1
  mat.results[i,4] = mean(jackests[[i]][,4])
  mat.results[i,5] = EJK(jackests[[i]][,5]) #sig
  mat.results[i,6] = mean(jackests[[i]][,6])
  mat.results[i,7] = EJK(jackests[[i]][,7]) #sigb
  mat.results[i,8] = mean(jackests[[i]][,8])
  mat.results[i,9] = EJK(jackests[[i]][,9]) #alpha
  mat.results[i,10] = mean(jackests[[i]][,10])
  mat.results[i,11] = EJK(jackests[[i]][,11]) #loglam
  mat.results[i,12] = mean(jackests[[i]][,12])
}

jacksumtab = matrix(NA, nrow=6, ncol=3)
colnames(jacksumtab) = c("True value", "Mean Unadj. SE", "EJK")
rownames(jacksumtab) = c("beta0", "beta1", "log(lambda)", "alpha", "sigma", "sigmab")

jacksumtab[,1] = c(1, 0.02, -2.302585, 1.2, 0.6, 0.64)
jacksumtab[,2] = apply(mat.results[,c(2,4,12,10,6,8)], 2, mean) #Not relevant
jacksumtab[,3] = apply(mat.results[,c(1,3,11,9,5,7)], 2, mean) #The EJK estimates.

#Results for each sigma_f (0, 5, 1) were saved as sumtab<sigf>, jacksumtab<sigf>, or
    bootsumtab<sigf>
table = matrix(NA, nrow=18, ncol=6)
table[,1] = rep(c(1, 0.02, -2.302585, 1.2, 0.6, 0.64), 3) #true parameters
table[,2] = c(sumtab0[,2], sumtab5[,2], sumtab1[,2])
table[,3] = c(sumtab0[,3], sumtab5[,3], sumtab1[,3]) #empirical SD
table[,4] = c(sumtab0[,4], sumtab5[,4], sumtab1[,4]) #JM SE
table[,5] = c(jacksumtab0[,3], jacksumtab5[,3], jacksumtab1[,3]) #EJK estimates
table[,6] = c(bootsumtab0[,3], bootsumtab5[,3], bootsumtab1[,3]) #BS estimates
```

C.4 Example Slurm Job Script for SHARCNET

```
#!/bin/bash

#SBATCH --account=<advisors_account>
#SBATCH --array=1-10%10
#SBATCH --ntasks=6
#SBATCH --mem-per-cpu=8GB
#SBATCH --time=03:00:00
#SBATCH --output=Sig1JM_test%A_%a.out
#SBATCH --mail-user=<email>
#SBATCH --mail-type=END
#SBATCH --mail-type=FAIL

module load gcc/10.3.0 r/4.1.0
module load jags/4.3.0
```

```
export R_LIBS=~/local/R_libs/

export NODESLIST=$(echo $(srun hostname | cut -f 1 -d '.'))

R -f fitJMmodels.R
```

Appendix D

Table of Notation

Table D.1: Table of notation

Variable	Description
g	Index representing clusters ($g = 1, 2, \ldots, G$)
i	Index representing the individuals ($i = 1, 2, \ldots, n_g$)
j	Index representing the time point ($j = 1, 2, \ldots, n_{gi}$)
k	Index representing the regression parameter β_k ($k = 0, 1, \ldots, p-1$)
m	Index representing the simulation iteration ($m = 1, 2, \ldots, n_{sim}$)
p	Number of fixed effects regression parameters for longitudinal model
q	Number of random effects regression parameters for longitudinal model
h	Number of baseline covariates for joint model survival sub-model
n	Number of individuals
n_g	Number of individuals within cluster g
n_{gi}	Number of repeated measurements for subject i in cluster g
t	Time
$m_{gi}(t)$	Smoothed time-varying covariate at time t for subject i in cluster g
$M_{gi}(t)$	Smoothed longitudinal covariate history for subject i in cluster g up to time point j
$y_{gi}(t)$	Observed longitudinal response at time t for subject i in cluster g
$\boldsymbol{\beta}$	$(p \times 1)$ vector of fixed effects parameters

Table D.1 – continued from previous page

Variable	Description
$\boldsymbol{b_{gi}}$	$(q \times 1)$ vector of random effects for subject i in cluster g
$\boldsymbol{X_{gi}}$	$(n_{gi} \times p)$ design matrix for subject i in cluster g
$\boldsymbol{Z_{gi}}$	$(n_{gi} \times q)$ random effects design matrix for subject i in cluster g
$\boldsymbol{\varepsilon_{gi}}$	$(n_{gi} \times 1)$ vector of random errors for subject i in cluster g
$\boldsymbol{D}$	$(q \times q)$ variance-covariance matrix of random-effects
$p(\cdot)$	Probability density function
$\boldsymbol{V_{gi}}$	$(n_{gi} \times n_{gi})$ variance-covariance matrix of observed longitudinal response, i.e. $\mathbf{Var}(\boldsymbol{y_{gi}})$
δ_{gi}	Censoring status for subject i in cluster g, $\delta_i \in (0, 1)$
$S(\cdot)$	Survival function
T^*_{gi}	Random variable of true time-to-event for subject i in cluster g
C_{gi}	Random variable of time to a censoring event for subject i in cluster g
T_{gi}	Random variable of observed time, $T_{gi} = \min(T^*_{gi}, C_{gi})$
$h(\cdot)$	Hazard function
$h_0(\cdot)$	Baseline hazard function
$H(\cdot)$	Cumulative hazard function
$\boldsymbol{\gamma}$	$(h \times 1)$ vector of survival baseline covariate coefficients
$\boldsymbol{w_{gi}}$	$(h \times 1)$ vector of survival baseline covariates for the i^{th} subject in the g^{th} cluster
$\boldsymbol{I}$	Identity matrix
$\ell(\cdot)$	Log-likelihood function
α	Association parameter, regression coefficient of the unobserved $m_{gi}(t)$
σ	Standard deviation of the measurement error random effects
σ_b	Standard deviation of the subject-level random effects of a joint model with a random intercept only longitudinal sub-model
λ	Scale parameter of the Weibull-PH hazard function
η	Shape parameter of the Weibull-PH hazard function
$\boldsymbol{\mathcal{S}(\cdot)}$	Score vector

Table D.1 – continued from previous page

Variable	Description
$\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}})$	Information matrix
$\hat{\theta}_{(-g)}$	Jackknife estimate of parameter θ when cluster g is omitted
$\hat{\theta}_{(\cdot)}$	Mean of the jackknife estimates for parameter θ, $\hat{\theta}_{(\cdot)} = \sum_{g=1}^{G} \hat{\theta}_{(-g)}/G$
u_g	Proportion of subjects in cluster g to total number of subjects n, $u_g = \frac{n_g}{n}$
$\tilde{\theta}_{(-g)}$	Pseudo-value for jackknife estimate of parameter θ when cluster g is omitted
$\hat{\theta}_n$	Estimate for joint model parameter θ based on the full sample
$\bar{\theta}_{(\cdot)}$	Weighted average of the G jackknife estimates for parameter θ, $\bar{\theta}_{(\cdot)} = G\hat{\theta}_n - \sum_{g=1}^{G}(1-u_g)\hat{\theta}_{(-g)}$
$\hat{\theta}^{(b)}$	Bootstrap estimate of parameter θ for bootstrap resample b for $(b = 1, 2, \ldots, B)$
$\hat{\theta}_W^{(b)}$	Weighted bootstrap estimate of parameter θ for bootstrap resample b for $(b = 1, 2, \ldots, B)$, $\hat{\theta}_W^{(b)} = \left(\frac{n_b^*}{n}\right)^{1/2} \hat{\theta}^{(b)}$
n_b^*	Number of subjects in the b^{th} bootstrap resample
B	Number of bootstrap resamples
$\zeta_{gi}(t)$	Cox model time-varying covariate
τ	Regression coefficient for Cox model time-varying covariate
n_{sim}	Number of simulation iterations
f_g	Family-level random effect for individuals within cluster g
$m_{gi}^F(t)$	Smoothed time-varying covariate at time t for subject i in cluster g with an added family random effect
$y_{gi}^F(t)$	Observed longitudinal response at time t for subject i in cluster g with an added family random effect