KNOT SELECTION STRATEGIES FOR SEMIPARAMETRIC VARYING COEFFICIENT MODELS APPLIED TO LONGITUDINAL COHORTS WITH MULTIPLE DROPOUT REASONS

by

CAMILLE MARIE MOORE

B.A., Cornell University 2003

M.A., New York University 2007

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Biostatistics Program

2013

This thesis for the Master of Science degree by

Camille Marie Moore

has been approved for the

Biostatistics Program

by

Samantha MaWhinney, Chair

Nichole Carlson

Jeri Forster

Date: 4/29/13

Moore, Camille Marie (M.S., Biostatistics)

Knot Selection Strategies for Semiparametric Varying Coefficient Models Applied to Longitudinal Cohorts with Multiple Dropout Reasons

Thesis directed by Professor Samantha MaWhinney

ABSTRACT

Dropout is a common source of missing data in longitudinal studies, and often occurs for reasons that may be related to an outcome of interest. When dropout depends on unobserved outcomes, even after conditioning on observable data, data are potentially missing not at random and dropout is therefore not ignorable. When the dropout mechanism is unspecified, semiparametric varying coefficient models can be used to account for non-ignorable dropout. This method is more robust than the parametric conditional linear approach. However, fitting these varying coefficient models requires the specification of the number and location of spline knots, which may influence model fit. We present simulation results comparing the natural cubic B spline varying coefficient method with a knot location selection algorithm to the same method with knots evenly placed at the quantiles of the dropout distribution, as well as to the classic conditional linear model. In addition, semiparametric varying coefficient models accounting for dropout time are extended to account for dropout reason as well. These methods are applied to data from the Acute Infection and Early Disease Research Program (AIEDRP) to determine the effect of injection drug use on the longitudinal trajectory of CD4+ T cell count in untreated HIV seropositive subjects.

The form and content of this abstract are approved. I recommend its publication.

Approved: Samantha MaWhinney

ACKNOWLEDGEMENTS

I wish to thank Samantha MaWhinney for her endless patience, support and guidance on this project. In addition, I would like to thank Jeri Forster for her extensive research developing natural cubic B spline varying coefficient models for non-ignorable dropout, which form the foundation for the statistical work presented in this thesis, and for generously sharing her knowledge (and R code) for fitting these models. I would also like to acknowledge Nichole Carlson for her statistical insights and Elizabeth Connick for sharing her clinical expertise.

CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

ABBREVIATION	
AIC	Akaike's information criterion
CLM	Conditional linear model
IDU	Injection drug users
MSE	Mean square error
NIDU	Non-injection drug user
NSV	Natural cubic B spline varying coefficient method
KSE	NSV with knot selection (evenly spaced candidate knots)
KSQ	NSV with knot selection (candidate knots at deciles)
RE	Random effects
ST	Started treatment
VCM	Varying coefficient model

CHAPTER I

INTRODUCTION

Non-ignorable Dropout in Longitudinal Studies

Dropout is a common problem in longitudinal clinical trials and cohort studies, and is of particular concern when dropout occurs for reasons that may be related to the outcome of interest. When the probability of dropout depends on unobserved outcomes, even after conditioning on available data, missing data are potentially missing not at random and therefore not ignorable. For example, in HIV/AIDS studies, subjects may drop out due to drug related side effects, viral resistance to the therapy, or disease progression. These reasons may be related to an outcome of interest, such as viral load, which quantifies the amount of virus in the body, or CD4+ T cell count, a measure of immunologic health. Traditional longitudinal data analysis methods, such as mixed models, do not account for non-ignorable dropout, and can potentially lead to biased results.

The parametric conditional linear model (Wu and Bailey, 1989; Verbeke et al., 2001) is an established mixture model method commonly used to account for non-ignorable dropout. The method assumes that regression coefficients are polynomial functions of dropout time, and is relatively simple to implement; however, misspecification of the parametric function can lead to biased estimates (Hogan et al., 2004; Forster et al., 2012). Semiparametric varying coefficient models can flexibly account for dropout and do not require distributional assumptions about the dropout mechanism, making them more robust when the form of the dropout mechanism is unknown (Hogan et al., 2004; Forster et al., 2012).

Traditionally, the conditional linear model and varying coefficient models have been used to account for the effect of dropout time, however,the reason a subject drops out of a study may also be associated with the trajectory of his or her outcome over time (Pauler et al., 2003). For example, subjects who dropout of a study due to death may have a different dropout varying slope than those who are simply lost

to follow up. In this paper, semiparametric varying coefficient models are extended to account for both dropout time and reason by allowing different dropout varying slopes for each distinct dropout reason.

In addition, this paper explores methods of improving the model fit of semiparametric varying coefficient methods. In these models, regression splines, such as natural cubic B splines, can be used to model dropout varying regression coefficients. however,both the number and location of knots for the splines must be specified (Liang et al., 2003; Hastie and Tibshirani, 1993; Forster et al., 2012). These parameters have the potential to greatly influence model fit; the number of knots controls the smoothness of the spline, while knot locations can influence the shape and flexibility of the spline over the range of dropout times. Correctly identifying knot locations may result in better model fit with fewer parameters, as well as less biased estimates.

Clinical Motivation: The Effect of Drug Use on HIV Disease Progression

Drug use has been hypothesized to accelerate the progression of HIV disease both by directly enhancing virus replication and/or impairing immune responses and by reducing compliance to effective therapy of HIV infection (Celetano and Lucas, 2007). While laboratory in vitro and animal studies suggest that opiates, cocaine, methamphetamines, and alcohol can impair the immune system and increase HIV replication (Peterson et al., 1993; Chao et al., 1996; Peterson et al., 2001; Squinto et al., 1990; Donahoe et al., 1993; Chuang et al., 1995; Donahoe, 2004; Veyries et al., 1995; Tashkin, 2004; Kresina et al., 2002; Bagasra et al., 1990), clinical data from epidemiological studies of the effect of drug and alcohol use on CD4+ T cell count and viral load have been mixed (Kapadia et al., 2005), with some studies showing no effect of drug use on HIV infection (Margolick et al., 1994; Study, 1992; Pezzotti et al., 1999; Rompalo et al., 2004; von Overbeck et al., 1994; Chaisson et al., 1995), others finding harmful effects (DesJarlais et al., 1987; Weber et al., 1990; Lucas et al., 2001), and some even finding protective effects (Farzadegan et al., 1996 ; Donahoe and Vlahov,

1998). It is possible these conflicting outcomes resulted from a failure to account for non-ignorable dropout in the analyses, which may have biased estimates towards subjects who completed the studies. Subjects who complete a study may have improved outcomes and may be less likely to engage in high risk behaviors, such as drug use, compared to those who drop out of the study earlier (Lanoya et al., 2006). Accounting for non-ignorable dropout may reveal a more consistent relationship between drug use and longitudinal CD4+ T cell count, and provide better insight into the effects of drug use on HIV disease progression.

CHAPTER II

METHODS

There are several likelihood based approaches for handling dropout that is potentially missing not at random, including selection, frailty, and mixture models. These methods are all based on factoring the joint distribution of the outcome and dropout time (Daniels and Hogan, 2008). This paper will focus on the mixture model approach, in which the full data are modeled as a mixture over dropout times or patterns.

Mixture Models

The use of mixtures of random effects models for handling non-ignorable dropout in longitudinal studies has been described by several authors (Wu and Carroll, 1988; Wu and Bailey, 1989; Mori et al., 1992; Hogan and Laird, 1997). Mixture models account for dropout by factoring the joint distribution of the outcome, y, and dropout time, u, as the product of the conditional distribution of the outcome for a given dropout time and the distribution of dropout times, $f(y,u) = f(y|u)f(u)$, and the full-data response distribution $f(y)$ is given by $\int f(y|u)dF(u)$.

Since outcomes for a subject are not observed after his or her dropout time, u, mixture models require assumptions about the behavior of the unobserved outcomes that occur after u. For for example, if $E[Y(t)|U=u]$ is linear in t, with intercept and slope depending on U, it is usual to assume that the relationship between the outcome and time is the same prior to and after u, so $E[Y(t)|U=u, t<u] = E[Y(t)|U=u, t \geq u]$. Even though time points after u are not observed, it is assumed that the outcome would continue along the same linear trajectory over time.

Varying Coefficient Models

Varying coefficient models (VCM) provide a general framework for fitting the conditional distribution $f(y|u)$. Wu and Bailey's conditional linear model (CLM), as well as pattern mixture models, can be viewed as special cases of VCM (Daniels and

Hogan, 2008). Assuming the outcome is distributed normally for a given dropout time, there are M subjects, and the ith subject has n_i observations and dropout time u_i, the subject specific form of the random effects VCM can be written:

$$(\mathbf{Y_i}|U = u_i) = \boldsymbol{X_i\beta(u_i)} + \boldsymbol{Z_i\alpha_i} + \boldsymbol{\epsilon_i} \tag{2.1}$$

where $\mathbf{X_i}$ is the n_i x p matrix of covariates for the fixed effects for subject i, $\boldsymbol{\beta(u_i)}$ is the p x 1 vector of dropout varying fixed effects regression coefficients, $\mathbf{Z_i}$ is the n_i x q design matrix for the random effects, $\boldsymbol{\alpha_i}$ is the q x 1 vector of random effects for subject i, which are distributed $N(0, D)$, and $\boldsymbol{\epsilon_i}$ is an n_i x 1 vector of error, with ϵ_{ij} distributed $N(0, \sigma^2_\epsilon)$. This model reduces to a standard random effects model, which does not depend on dropout time, if $\boldsymbol{\beta(u)} = \boldsymbol{\beta}$.

Assuming a simple linear model with a random intercept and slope ($D = \left(\begin{smallmatrix} \sigma_0^2 & \sigma_{01} \\ \sigma_{01} & \sigma_1^2 \end{smallmatrix}\right)$), where the outcome can be modeled as a linear function of time, the observation specific model can be re-written as:

$$(Y_{ij}|U = u_i) = \beta_0(u_i) + \beta_1(u_i)t_{ij} + \alpha_{0i} + \alpha_{1i}t_{ij} + \epsilon_{ij} \tag{2.2}$$

We can think of the outcome vector for a subject $\mathbf{Y_i} = (Y_{i1}, ..., Y_{in_i})^T$ as arising from a process $\{Y_i(t) : t \geq 0\}$ observed at times t_{i1} to t_{in_i}. Conditional on dropout time $U = u_i$ and the subject specific random effects, $Y_i(t)$ has a mean $\beta_0(u_i) + \beta_1(u_i)t + \alpha_{0i} + \alpha_{1i}t$. Since we assume the outcome is normally distributed and the random effects have mean 0, the population or marginal mean, $Y(t)$, conditional on $U = u$ is given by the mean function $\mu(t|U = u) = \beta_0(u) + \beta_1(u)t$. This process is conditionally normal, such that

$$(Y(t)|U = u) \sim N(\mu(t|u), \sigma^2_\epsilon + \sigma^2_0 + 2\sigma_{01}t + \sigma^2_1t^2) \tag{2.3}$$

For a given u, the functional form of $\mu(t|u)$ is assumed to be known, but as a function of u, $\mu(t|u)$ can be any smooth function. To obtain $E[Y(t)]$, which does not depend on dropout time, we integrate the conditional mean function over the distribution of the dropout times:

$$E[Y(t)] = \int \mu(t|u)dF(u) \tag{2.4}$$

Note that $E[Y(t)]$ is also a linear function of time since:

$$E[Y(t)] = \int [\beta_0(u) + \beta_1(u)t]dF(u) \tag{2.5}$$

$$= \int \beta_0(u)dF(u) + t\int \beta_1(u)dF(u) \tag{2.6}$$

Therefore the marginal intercept and slope are given by the expected values of their corresponding dropout varying coefficients, $\beta_k^* = E[\beta_k(u_i)]$, where $k = 0$ for the intercept and $k = 1$ for the slope. In practice, the distribution of u is unknown, and the marginal coefficients are estimated using the empirical distribution of dropout times:

$$\hat{\beta}_k^* = \int \hat{\beta}_k(u)d\hat{F}(u) \tag{2.7}$$

$$= \hat{\mathbf{\Pi}}^{\boldsymbol{T}}\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}(\boldsymbol{u^0}) \tag{2.8}$$

where $\mathbf{u^0} = (u_1^0, ..., u_R^0)^T$ is a vector of the R ordered unique dropout times, $\mathbf{\Pi}$ is a vector of the R proportions of subjects with each unique dropout time u_r^0, and $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}(\boldsymbol{u^0})$ is an R x 1 vector of smooth function values. This is a weighted average of the dropout varying coefficients over the unique dropout times, and is equivalent to taking the average of the dropout varying coefficients for each subject:

$$\hat{\beta}_k^* = \frac{1}{M}\sum_{i=1}^{M}\hat{\beta}_k(u_i) \tag{2.9}$$

An important feature of this model is that $f(u)$ can be left unspecified. It is important to note, however, that the full data mean of $Y(t)$ at a fixed time $t = t^*$ will be based on extrapolations for those who have dropped out prior to t^*.

Extending VCM to Account for Dropout Reason

VCMs can easily be extended to account for dropout reason by allowing a different dropout varying slope for each distinct dropout reason. Let $h = \{1, ..., H\}$ denote dropout reason and $g = \{1, ..., G\}$ denote group, such as a treatment or drug use group. Let m_g be the number of subjects in group g, m_h be the number of subjects with dropout reason h, and $m_{h|g}$ be the number of subjects in group g with dropout reason h. Assume a total of M subjects with n_i observations each. Let u_i represent the ith subject's dropout time. For the ith subject, $\mathbf{Y_i}$ is an n_i x 1 vector of outcomes, $\mathbf{1_i}$ is an n_i x 1 vector of 1's, and $\mathbf{t_i}$ is an n_i x 1 vector of observation times.

The joint distribution of the outcome, y, dropout time, u, and reason, h, can be written as:

$$f(y, u, h|g) = f(y|u, h, g)f(u, h|g) \tag{2.10}$$

$$= f(y|u, h, g)f(u|h, g)p(h|g) \tag{2.11}$$

Assume the outcome, y, given the dropout time and reason is normally distributed, and that the distribution of dropout reasons for a given group is multinomial. The distribution of dropout times given dropout reason and group can be left unspecified.

Define $\mathbf{u^0_{h|g}} = (u^0_{1_{h|g}}, ..., u^0_{R_{h|g}})^T$ as the vector of $R_{h|g}$ ordered dropout times for subjects with dropout reason h in group g, $\mathbf{\Pi_{u|h,g}}$ as the vector of $R_{h|g}$ proportions of subjects with dropout reason h in group g with each dropout time $u^0_{r_{h|g}}$ (vector of proportions with denominator $m_{h|g}$), and $\mathbf{\Pi_{h|g}}$ as the vector of H proportions of subjects with each dropout reason h in group g (vector of $\frac{m_{h|g}}{m_g}$). Assume the following

model:

$$(\mathbf{Y_i}|u = u_i, g = g_i, h = h_i) = \mathbf{1_i}\beta_{g_i h_i 0}(u_i) + \mathbf{t_i}\beta_{g_i h_i 1}(u_i) + \mathbf{Z_i}\boldsymbol{\alpha_i} + \boldsymbol{\epsilon_i} \tag{2.12}$$

The expected value of $(Y_{ij}|h, g)$ is given by:

$$E(Y_{ij}|h, g) = \int \left[\beta_{gh0}(u_{h|g}) + t_{ij}\beta_{gh1}(u_{h|g})\right] dF(u_{h|g}) \tag{2.13}$$

$$= \int \beta_{gh0}(u_{h|g})dF(u_{h|g}) + t_{ij} \int \beta_{gh1}(u_{h|g})dF(u_{h|g}) \tag{2.14}$$

The marginal coefficients for each dropout reason and group combination, $\beta_{gk}(h)$, are given by:

$$\beta_{gk}(h) = \int \beta_{ghk}(u_{h|g})dF(u_{h|g}) \tag{2.15}$$

where $k = 0$ for the intercept and $k = 1$ for the slope.

Since the distribution of $u_{h|g}$ is unknown, the marginal coefficients are estimated using the empirical distribution of dropout times:

$$\hat{\beta}_{gk}(h) = \int \hat{\beta}_{ghk}(u_{h|g})d\hat{F}(u_{h|g}) \tag{2.16}$$

$$= \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{u|h,g}}\hat{\boldsymbol{\beta}}_{\boldsymbol{ghk}}(\boldsymbol{u}^{\mathbf{0}}_{\boldsymbol{h|g}}) \tag{2.17}$$

This is a weighted average of the dropout varying coefficients over the unique dropout times for dropout reason h in group g. It is equivalent to taking the average of the dropout varying coefficients for each subject with dropout reason h in group g:

$$\hat{\beta}_{gk}(h) = \frac{1}{m_{h|g}} \sum_{i=1}^{m_{h|g}} \hat{\beta}_{ghk}(u_i) \tag{2.18}$$

Marginal coefficients averaged over dropout reason for each group, β^*_{gk}, can be

obtained as well:

$$\beta^*_{gk} = \sum_{h=1}^{H} p(h|g) \int \beta_{ghk}(u_{h|g}) dF(u_{h|g}) \tag{2.19}$$

$$= \sum_{h=1}^{H} p(h|g) \beta_{gk}(h) \tag{2.20}$$

$$\hat{\beta}^*_{gk} = \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{h|g}} \hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h}) \tag{2.21}$$

$$= \frac{1}{m_g} \sum_{i=1}^{m_g} \hat{\beta}_{gk}(h_i) \tag{2.22}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h})$ is the H x 1 vector of estimated dropout reason specific coefficients for group g.

Differing Forms of the Dropout Time and Reason VCM

VCMs can account for group, dropout reason and dropout time in several different frameworks depending on the assumptions made about the relationship between group and the dropout varying coefficients (Figure 2.1). In the model outlined above each dropout reason and group combination is allowed to have a distinct functional form for the dropout varying coefficients (Figure 2.1-B). This is the most general case, and all other examples presented are special cases of this model. For example, $H = 1$ results in the model that does not account for dropout reason (Figure 2.1-A). Two additional special cases of this model are presented below. In these models, it is assumed that the dropout varying coefficients are comprised of a group effect that does not depend on dropout time and a dropout varying component that depends on dropout reason but not group (Figure 2.1-C and D). These models may be advantageous when sample sizes are low in certain dropout reason and group combinations, which may make it unreasonable to estimate a distinct functional form of the dropout varying coefficients for all dropout reason and group combinations.

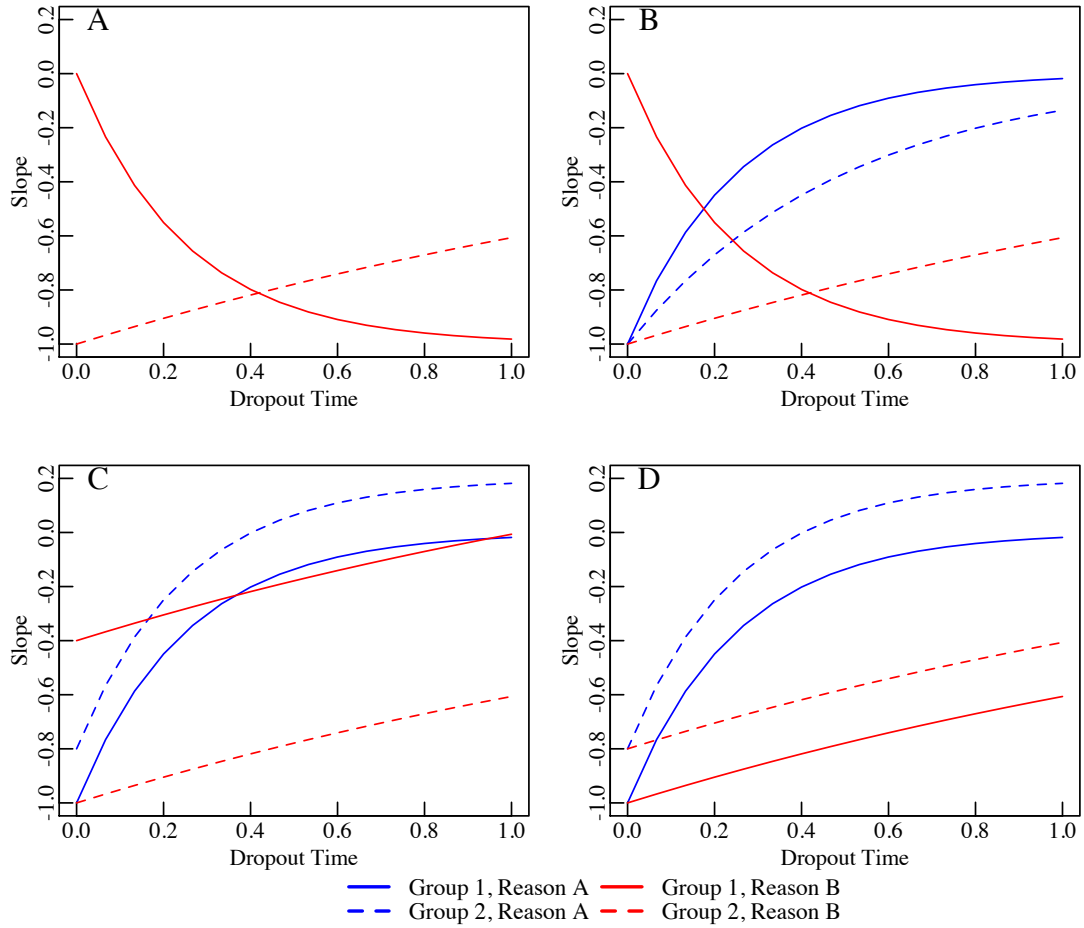


Figure 2.1: Examples of Relationships Between Dropout Time, Reason, and Group. Panel A depicts a VCM model with a group effect that does not account for dropout reason. In Panel B, dropout reason is accounted for and a different functional form of the slope is allowed for each dropout reason and group combination. In Panel C, the functional form of the slope depends only on dropout reason and not group. In Panel D, the functional form of the slope depends only on dropout reason, and in addition, the effect of group is assumed to be the same across dropout reasons.

Common dropout varying component of the coefficients for the groups, different effect of group for each dropout reason. This model assumes that the functional form of the dropout varying coefficients is the same for groups within a dropout reason, but group by dropout reason and group by dropout reason by time interactions are included as fixed effects in the model in order to allow the effect of group to vary according to dropout reason (Figure 2.1-C). This results in the following model:

$$(\mathbf{Y_i}|u = u_i, g = g_i, h = h_i) = \mathbf{1_i}\left[\beta_{g_i h_i 0} + \beta_{h_i 0}(u_i)\right] + \mathbf{t_i}\left[\beta_{g_i h_i 1} + \beta_{h_i 1}(u_i)\right] + \mathbf{Z_i}\boldsymbol{\alpha_i} + \boldsymbol{\epsilon_i} \tag{2.23}$$

The expected value of $(Y_{ij}|h, g)$ is given by:

$$E(Y_{ij}|h, g) = \int \left[\beta_{gh0} + \beta_{h0}(u_{h|g}) + t_{ij}\beta_{gh1} + t_{ij}\beta_{h1}(u_{h|g})\right] dF(u_{h|g}) \tag{2.24}$$

$$= \beta_{gh0} + \int \beta_{h0}(u_{h|g})dF(u_{h|g}) + t_{ij}\left[\beta_{gh1} + \int \beta_{h1}(u_{h|g})dF(u_{h|g})\right] \tag{2.25}$$

The marginal coefficients for each dropout reason and group combination $\beta_{gk}(h)$ are given by:

$$\beta_{gk}(h) = \beta_{ghk} + \int \beta_{hk}(u_{h|g})dF(u_{h|g}) \tag{2.26}$$

where $k = 0$ for the intercept and $k = 1$ for the slope.

The marginal coefficients can be estimated using the empirical distribution of dropout times:

$$\hat{\beta}_{gk}(h) = \hat{\beta}_{ghk} + \int \hat{\beta}_{hk}(u_{h|g})d\hat{F}(u_{h|g}) \tag{2.27}$$

$$= \hat{\beta}_{ghk} + \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{u|h,g}}\hat{\boldsymbol{\beta}}_{\boldsymbol{hk}}(\boldsymbol{u}^{\mathbf{0}}_{\boldsymbol{h|g}}) \tag{2.28}$$

Again, this is a weighted average of the dropout varying coefficients over the unique dropout times for dropout reason h in group g, and is equivalent to taking the average

of the dropout varying coefficients for each subject with dropout reason h in group g:

$$\hat{\beta}_{gk}(h) = \hat{\beta}_{ghk} + \frac{1}{m_{h|g}} \sum_{i=1}^{m_{h|g}} \hat{\beta}_{hk}(u_i) \tag{2.29}$$

Marginal coefficients averaged over dropout reason for each group, β^*_{gk}, are given by:

$$\beta^*_{gk} = \sum_{h=1}^{H} p(h|g) \left[\beta_{ghk} + \int \beta_{hk}(u_{h|g}) dF(u_{h|g}) \right] \tag{2.30}$$

$$= \sum_{h=1}^{H} p(h|g) \beta_{gk}(h) \tag{2.31}$$

$$\hat{\beta}^*_{gk} = \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{h|g}} \hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h}) \tag{2.32}$$

$$= \frac{1}{m_g} \sum_{i=1}^{m_g} \hat{\beta}_{gk}(h_i) \tag{2.33}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h})$ is the H x 1 vector of estimated dropout reason specific coefficients for group g.

Common dropout varying components of the coefficients for the groups, same effect of group across dropout reason. This model assumes that the effect of group is the same across dropout reasons (Figure 2.1-D). Assume the following model:

$$(\mathbf{Y_i}|u = u_i, g = g_i, h = h_i) = \mathbf{1_i} \left[\beta_{g_i 0} + \beta_{h_i 0}(u_i)\right] + \mathbf{t_i} \left[\beta_{g_i 1} + \beta_{h_i 1}(u_i)\right] + \mathbf{Z_i}\boldsymbol{\alpha_i} + \boldsymbol{\epsilon_i} \tag{2.34}$$

The expected value of $(Y_{ij}|h, g)$ is given by:

$$E(Y_{ij}|h, g) = \int \left[\beta_{g0} + \beta_{h0}(u_{h|g}) + t_{ij}\beta_{g1} + t_{ij}\beta_{h1}(u_{h|g})\right] dF(u_{h|g}) \tag{2.35}$$

$$= \beta_{g0} + \int \beta_{h0}(u_{h|g}) dF(u_{h|g}) + t_{ij} \left[\beta_{g1} + \int \beta_{h1}(u_{h|g}) dF(u_{h|g})\right] \tag{2.36}$$

The marginal coefficients for each dropout reason and group combination $\beta_{gk}(h)$ are given by:

$$\beta_{gk}(h) = \beta_{gk} + \int \beta_{hk}(u_{h|g}) dF(u_{h|g}) \tag{2.37}$$

where $k = 0$ for the intercept and $k = 1$ for the slope.

Marginal coefficients are estimated using the empirical distribution of dropout times:

$$\hat{\beta}_{gk}(h) = \hat{\beta}_{gk} + \int \hat{\beta}_{hk}(u_{h|g}) d\hat{F}(u_{h|g}) \tag{2.38}$$

$$= \hat{\beta}_{gk} + \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{u|h,g}} \hat{\boldsymbol{\beta}}_{\boldsymbol{hk}}(\boldsymbol{u}^{\boldsymbol{0}}_{\boldsymbol{h|g}}) \tag{2.39}$$

resulting in a weighted average of the dropout varying coefficients over the unique dropout times for dropout reason h in group g. It is equivalent to taking the average of the dropout varying coefficients for each subject with dropout reason h in group g:

$$\hat{\beta}_{gk}(h) = \hat{\beta}_{gk} + \frac{1}{m_{h|g}} \sum_{i=1}^{m_{h|g}} \hat{\beta}_{hk}(u_i) \tag{2.40}$$

Marginal coefficients averaged over dropout reason for each group, β^*_{gk}, can be obtained as well:

$$\beta^*_{gk} = \sum_{h=1}^{H} p(h|g) \left[\beta_{gk} + \int \beta_{hk}(u_{h|g}) dF(u_{h|g}) \right] \tag{2.41}$$

$$= \sum_{h=1}^{H} p(h|g) \beta_{gk}(h) \tag{2.42}$$

$$\hat{\beta}^*_{gk} = \hat{\mathbf{\Pi}}^{\boldsymbol{T}}_{\boldsymbol{h|g}} \hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h}) \tag{2.43}$$

$$= \frac{1}{m_g} \sum_{i=1}^{m_g} \hat{\beta}_{gk}(h_i) \tag{2.44}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{gk}}(\boldsymbol{h})$ is the H x 1 vector of estimated dropout reason specific coefficients for group g.

Fitting the VCM

In order to fit the VCM, a method of modeling and estimating the functions $\beta(u)$ must be chosen. Using parametric, polynomial functions for $\beta(u)$ results in Wu and Bailey's CLM. However, there are nonparametric approaches as well. Hogan et al. (2004) used smoothing splines for $\beta(u)$ and used a roughness penalty to estimate the smoothing parameter as an extra variance component. However, this method is not trivial to implement and suffers from convergence problems. Forster et al. (2012) proposed using natural cubic B splines to model the dropout mechanism instead. Natural cubic B splines have the advantage of being numerically stable, and are linear past their boundary knots, which can improve model behavior when the data are sparse near the boundaries. In addition, compared to smoothing splines, regression splines require fewer knots and few unknown basis coefficients. The main obstacle to using regression splines, however, is the choice of the number and the location of the knots.

VCM with Natural Cubic B Splines

The VCM proposed by Forster et al. (2012) relies on natural cubic B-spline basis functions to model the dropout mechanism, and only requires that the dropout mechanism be a smooth, continuous function. B-splines are piecewise polynomials that join smoothly at a sequence of k interior knots, resulting in $k + 1$ knot spans. Additionally, there are boundary knots, which are usually placed at the data endpoints. Standard B-spline basis functions are flexible, nearly orthogonal, numerically stable, easy to compute and possess local support. Natural cubic B-splines add the constraint that the spline function is linear beyond the boundary knots. While not all natural cubic B-splines possess local support, the linearity constraint reduces the number of parameters needed to define the spline and can improve model behavior if data are sparse near the boundaries.

The conditional natural spline VCM (NSV) accounting for dropout time and rea-

son can be written as:

$$(\mathbf{Y_i}|U = u_i, g = g_i, h = h_i) = \mathbf{1_i} \sum_{j=0}^{J_{gh0}} \theta_{ghj0} \tilde{u}_{ghij0} + \mathbf{t_i} \sum_{j=0}^{J_{gh1}} \theta_{ghj1} \tilde{u}_{ghij1} + \boldsymbol{Z_i \alpha_i} + \boldsymbol{\epsilon_i} \quad (2.45)$$

where $\tilde{u}_{ghijk} = \tilde{B}(u_{h|g}, J_{ghk})_{[i,j+1]}$, for $k = 0, 1$. For $j > 0$, $\tilde{B}(u_{h|g}, J_{ghk})$ is the matrix of basis functions with J_{ghk} degrees of freedom and $\tilde{B}(u_{h|g}, J_{ghk})_{[,1]} = 1$. The ith subjects dropout varying coefficients are given by:

$$\beta_{ghik} = \sum_{j=0}^{J_{ghk}} \theta_{ghjk} \tilde{u}_{ghijk} = \sum_{j=0}^{J_{ghk}} \theta_{ghjk} \tilde{B}(u_{h|g}, J_{ghk}) \quad (2.46)$$

Since subjects that drop out very early have few observations, estimates of their slopes are inherently less stable. To increase stability, the lower boundary of the spline, d_L, can be moved inward, so the coefficients below the boundary cutoff are restricted to change linearly as a function of dropout time.

Previously Published Knot Placement Methods

Several methods have been proposed to determine the number and location of knots for regression splines in semiparametric models, although most have focused on additive rather than varying coefficient models. From the frequentist perspective, Friedman and Silverman (1989) proposed a forward/backward selection algorithm to determine the location of knots for piecewise polynomial splines in a simple additive model. These methods chose knots from a large set of candidates comprised of all the distinct values of the predictor in the dataset. A forward selection procedure gradually added knots to the model based on a generalized cross validation measure, and then a backwards selection procedure removed knots from the model until a final model with an optimal generalized cross validation measure was found.

Denison et al. (1998) proposed fitting general additive models using a hybrid reversible jump Markov chain Monte Carlo/least squares method. Rather than im-

plementing a fully Bayesian approach, regression coefficients for the splines' basis functions were taken to be the least squares estimates. A complete Bayesian approach would have assigned prior distributions to these coefficients and worked with an extended posterior distribution which would have also included these parameters. Denison et al. cite serious additional computational burden as the reason for using the least squares estimator.

Biller and Fahrmeir (2000) proposed a fully Bayesian reversible jump Markov chain Monte Carlo (RJMCMC) approach to fitting VCMs with regression splines. For each of the unknown varying coefficients, the number and location of knots and the natural spline coefficients were estimated simultaneously using RJMCMC sampling. The overall procedure can be thought of as a kind of Bayesian model averaging, since a single collection of knots does not need to be chosen. This method was applied to several datasets, however,it has not been used to account for non-ignorable dropout or in longitudinal models that include random effects.

Proposed Knot Selection Methods

In order to determine the number and placement of knots, a simple combinatorial knot selection method is proposed. The algorithm determines the best model from a set of candidates based on Akaike's Information Criterion (AIC). First, a set of candidate knots based on the observed distribution of dropout times must be chosen. The boundary knots for the splines may be set at the minimum and maximum dropout times or moved inward to produce more stable results at the tails. Reasonable sets of interior knots might include the deciles of the dropout times between the boundary knots or 9 or 10 evenly spaced dropout times between the boundaries. In this paper, knots placed at the deciles (KSQ) and 9 evenly spaced knots (KSE) are considered. Models are fit using maximum likelihood for all possible combinations of candidate knots. The model with the lowest AIC is taken and a final model is fit using restricted maximum likelihood estimation (REML). While this method of knot

selection is conceptually simple, it can be computationally intensive, since the model must be fit many times. Due to this limitation, only a small set of candidate knots can be reasonably considered. In addition, this method chooses one "best" model, even though there may only be small differences in AIC between several of the models.

CHAPTER III

SIMULATION STUDY

Description of Simulated Datasets

The proposed knot selection methods were used to fit models for the same simulated datasets described in Forster et al. (2012), which had uniform dropout over time, as well as for additional simulated datasets with heavy early and heavy late distributions of dropout times. Three different forms of the dropout mechanism were considered: (i) a continuous and smooth function meeting assumptions of the NSV, (ii) a continuous, but not smooth function, and (iii) a discontinuous step function. This resulted in nine different simulation scenarios.

More specifically, the following form for the data was assumed:

$$Y_{ij} = \beta_0(u_i) + \beta_1(u_i)t_{ij} + \alpha_{0i} + \alpha_{1i}t_{ij} + \epsilon_{ij}, i = 1...m, j = 1...n_i \tag{3.1}$$

for m subjects with n_i observations for the ith subject, where $(\alpha_{0i}, \alpha_{1i})^T \sim N(0, D)$, $\epsilon_{ij} \sim N(0, \sigma^2)$, and $\beta_0(u_i) = 0$. Uniform dropout was created from a beta-binomial where $p \sim Beta(1.5, 1.5)$ and $U \sim Bin(15, p)$. Heavy early dropout and heavy late dropout were also created from beta-binomials where $p \sim Beta(3, 1.5)$ and $U \sim Bin(15, p)$, and $p \sim Beta(1.5, 3)$ and $U \sim Bin(15, p)$, respectively.

Dropout times are $u = U/15 \in [0, 1]$, resulting in 16 time points spaced equally from 0 to 1. The forms of the dropout varying slope, $\beta_1(u)$, are: (i) $-exp(\alpha u)$, (ii) $-exp(\alpha u)I_{(u<t^*)} - exp(\alpha t^*)I_{(u\geq t^*)}$ and (iii) $\alpha_1 I_{(u<t^*)} + \alpha_2 I_{(u\geq t^*)}$.We set $\alpha = -4$ for forms (i) and (ii), defined $t^* = 2/3$ and for form (iii), $\alpha_1 = 0$ and $\alpha_2 = 1$. The within-subject variance, σ^2 was set at 0.067. The elements of D are as follows: $d_{11} = 0.4$, $d_{22} = 0.01$ and $d_{12} = -0.01$. For each form and dropout distribution combination, 1,000 datasets with 400 subjects each were created.

Analysis Methods

KSQ, KSE, NSV, and CLM models were fit to each dataset. For the KSQ, KSE and NSV, a maximum of 6 degrees of freedom were considered for the dropout varying component of the slope, for a maximum of 7 total degrees of freedom for the slope. The intercept was not allowed to vary with dropout time. AIC was used to choose the number of knots for the NSV, and the number and location of knots for the KSQ and KSE methods. The left boundary offset was set to 0.20 (corresponding to 4 observations) for these models, after Forster et al. Likelihood ratio tests were used to select the best CLM for each dataset with a cubic polynomial as the full model.

The performance of the methods is compared graphically and in terms of bias, variance, and mean square error for the marginal slope, as well as for the sixteen dropout time specific slopes. All analyses were implemented in R using the nlme, lme4, and splines packages.

Knot Selection Algorithm Results

Model Fit

Graphs of the predicted KSQ, KSE, NSV, and CLM slopes at each dropout time are presented in Figure 3.1. These graphs show that the KSE and KSQ methods are able to fit the dropout varying slope accurately for forms that meet as well as forms that violate model assumptions, and provide better or comparable model fit to NSV and and better fit than CLM methods over the range of scenarios. In particular, the NSV, KSQ, and KSE methods perform very well for form (iii) despite the violations of the model assumptions. For this form, the CLM predominantly fits a cubic polynomial, which results in poor model fit for those with early dropout times.

Bias, Variance, and Mean Square Error for the Marginal Slope

The performance of the models is also quantified in terms of bias, variance, and mean squared error (MSE) for the marginal slope. The error of the marginal slope

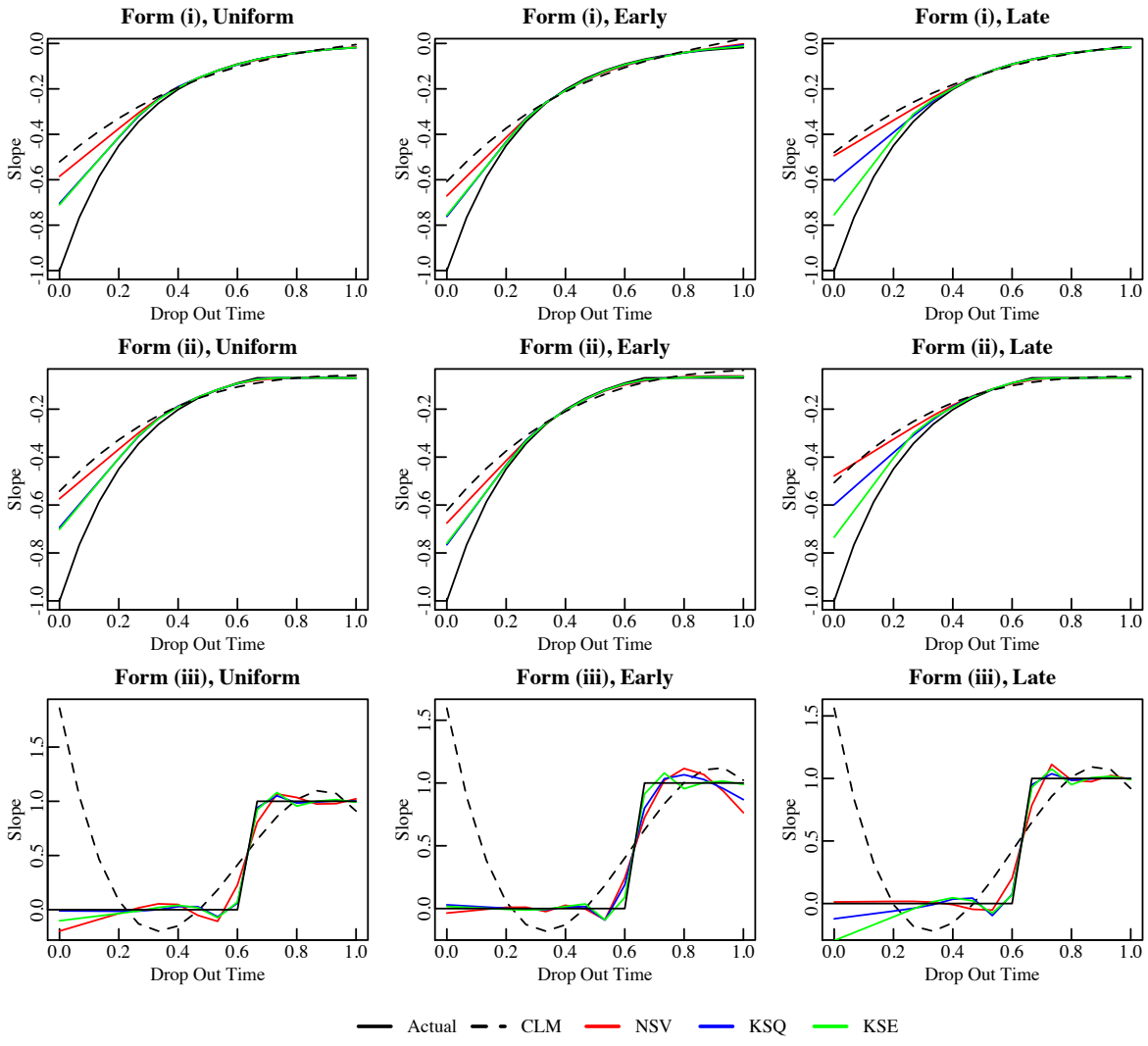


Figure 3.1: Dropout Varying Slopes

Table 3.1: Marginal Slopes: Mean(SD). Slopes with the lowest bias are bolded in each row.

Form	True Slope	KSQ	KSE	NSV	CLM
i Uniform	-0.2343	-0.2066 (0.1055)	**-0.2072** (0.1084)	-0.1912 (0.0955)	-0.1795 (0.0822)
i Early	-0.3736	**-0.3378** (0.1311)	-0.3375 (0.1267)	-0.3184 (0.1172)	-0.2966 (0.105)
i Late	-0.1104	-0.1034 (0.0368)	**-0.1053** (0.0385)	-0.0989 (0.0352)	-0.0964 (0.0324)
ii Uniform	-0.2434	-0.213 (0.1076)	**-0.2138** (0.11)	-0.1973 (0.0992)	-0.1888 (0.0879)
ii Early	-0.3757	**-0.3404** (0.1323)	**-0.3404** (0.1278)	-0.3214 (0.121)	-0.3009 (0.1109)
ii Late	-0.1277	-0.1189 (0.0385)	**-0.1208** (0.04)	-0.114 (0.0368)	-0.1131 (0.034)
iii Uniform	0.3469	**0.3448** (0.1022)	0.3362 (0.1005)	0.3303 (0.103)	0.4697 (0.0963)
iii Early	0.1199	0.1248 (0.128)	**0.1209** (0.1182)	0.1156 (0.13)	0.3462 (0.1342)
iii Late	0.5939	0.5906 (0.0429)	0.5879 (0.0425)	**0.5948** (0.0465)	0.5994 (0.0398)

for each simulated dataset was calculated as the estimated marginal slope less the expected value of the slope based on the simulation settings. Bias was calculated as the mean of these errors. In addition, the MSE was calculated as the mean of the squared errors for each dataset. The marginal slopes, bias, variance, MSE, and degrees of freedom are presented in Tables 3.1-3.3.

Over the range of simulations, bias for the marginal slope is reduced using KSQ and KSE, although the NSV did perform better than these methods for form (iii) with heavy late dropout. The KSQ and KSE outperform the CLM in terms of bias in all nine scenarios. In terms of MSE for the marginal slope, the KSQ and KSE tend to perform similarly to the NSV and the CLM, however, the CLM does often have the lowest MSE, despite the increase in bias using this method. While the CLM is more biased than the other methods, the variance for the marginal slope is perhaps artificially reduced since the CLM is less flexible and there are fewer models investigated for each dataset (4 compared to approximately 1500 for the knots selection methods).

The average number of degrees of freedom used for the models was slightly increased using the KSE and KSQ knot selection methods, rather than decreased as had been hypothesized (Table 3.3).

Table 3.2: Bias, Variance, and MSE for the Marginal Slope: Bias(Variance), MSE. The lowest bias and variance are bolded in each row.

Form	KSQ	KSE	NSV	CLM
i Uniform	0.0277 (0.0111)	**0.0272** (0.0118)	0.0431 (0.0091)	0.0548 **(0.0068)**
	0.0119	0.0125	0.011	**0.0098**
i Early	**0.0357** (0.0172)	0.036 (0.016)	0.0552 (0.0137)	0.0769 **(0.011)**
	0.0184	0.0173	**0.0168**	0.0169
i Late	0.0071 (0.0014)	**0.0051** (0.0015)	0.0115 (0.0012)	0.0141 **(0.001)**
	0.0014	0.0015	0.0014	**0.0012**
ii Uniform	0.0305 (0.0116)	**0.0296** (0.0121)	0.0461 (0.0098)	0.0546 **(0.0077)**
	0.0125	0.013	0.0119	**0.0107**
ii Early	**0.0353** (0.0175)	**0.0353** (0.0163)	0.0543 (0.0146)	0.0748 **(0.0123)**
	0.0187	**0.0176**	**0.0176**	0.0179
ii Late	0.0088 (0.0015)	**0.0069** (0.0016)	0.0137 (0.0014)	0.0146 **(0.0012)**
	0.0016	0.0017	0.0015	**0.0014**
iii Uniform	**-0.002** (0.0104)	-0.0107 (0.0101)	-0.0166 (0.0106)	0.1229 **(0.0093)**
	0.0104	**0.0102**	0.0109	0.0244
iii Early	0.0048 (0.0164)	**0.001 (0.014)**	-0.0044 (0.0169)	0.2262 (0.018)
	0.0164	**0.014**	0.0169	0.0692
iii Late	-0.0033 (0.0018)	-0.006 (0.0018)	$<$ **0.0001** (0.0022)	0.0055 **(0.0016)**
	0.0018	0.0018	0.0022	**0.0016**

Table 3.3: Degrees of Freedom for the Slope: 1= No Dropout Varying Effect

Form	KSQ	KSE	NSV
i Early	3.387	3.566	2.874
i Uniform	3.394	3.425	2.772
i Late	3.250	3.325	2.701
ii Early	3.408	3.599	2.960
ii Uniform	3.311	3.331	2.709
ii Late	3.143	3.231	2.580
iii Early	5.962	6.696	6.716
iii Uniform	6.659	6.749	6.505
iii Late	6.794	6.915	6.906

Bias, Variance, and MSE for the Dropout Time Specific Slopes

It is difficult to summarize the fit of an entire model with a single summary measure, such as the MSE for the marginal slope. The bias, variance and MSE were calculated again for the slope at each of the 16 possible dropout times in the simulations. Figures 3.2-3.4 depict the bias, variance and MSE for the dropout time specific slopes for each method. While the CLM had the lowest MSE for the marginal slope in several of the scenarios, looking at the dropout specific plots, the methods

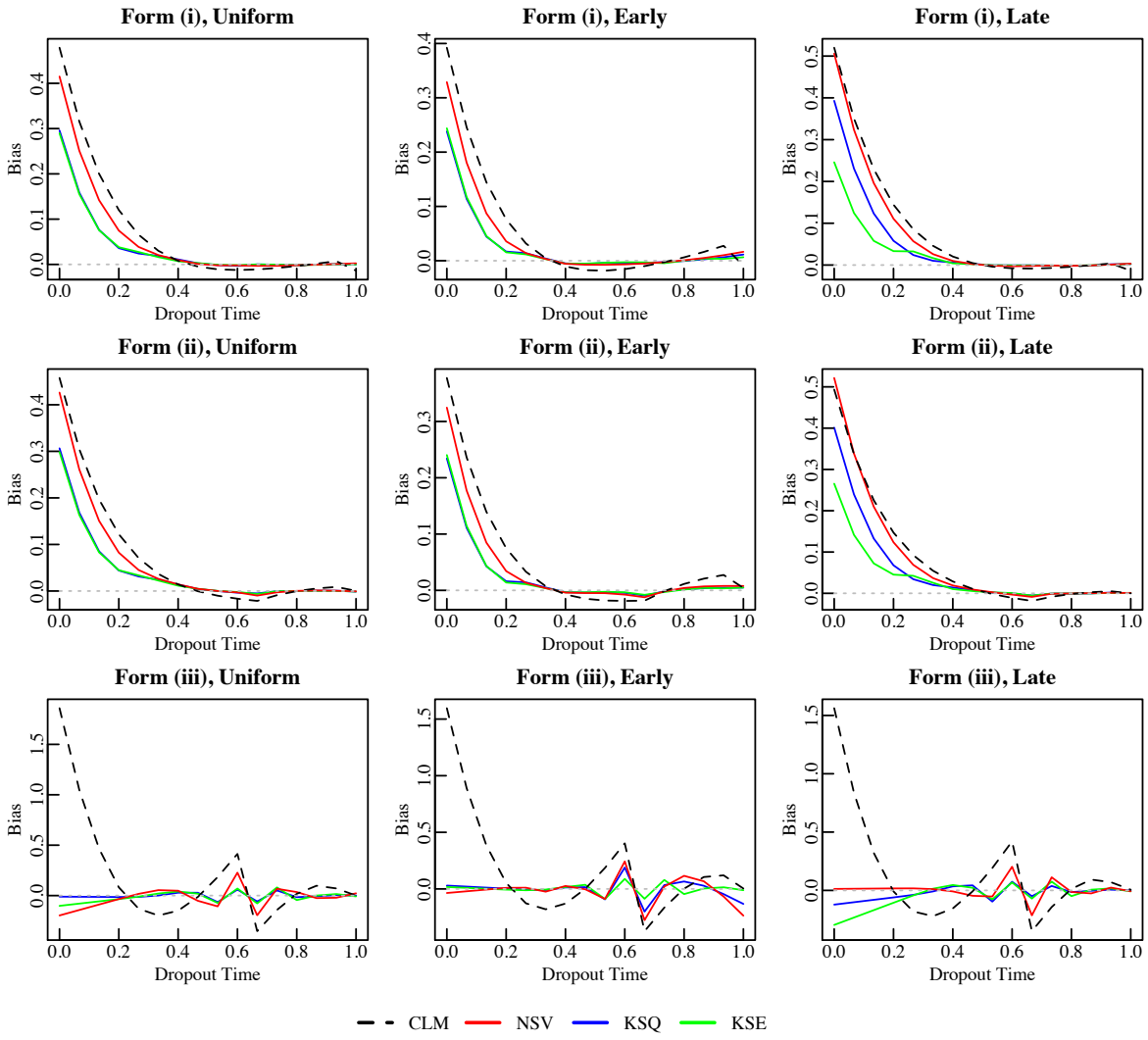


Figure 3.2: Bias for Dropout Time Specific Slopes

all seem to perform similarly in terms of MSE for forms (i) and (ii). The lack of fit of the CLM is highlighted in the plots for form (iii). The MSE for the KSE for forms (i) and (ii) with heavy late dropout is higher than for the other methods at early dropout times. While the bias for the KSE and KSQ is lower, the variance of the KSE and KSQ methods is higher than the CLM and NSV at early dropout times. This increased variance may reflect the uncertainty of the dropout varying slope at early dropout times when there are few observations, as well as the increased flexibility of the KSQ and KSE methods.

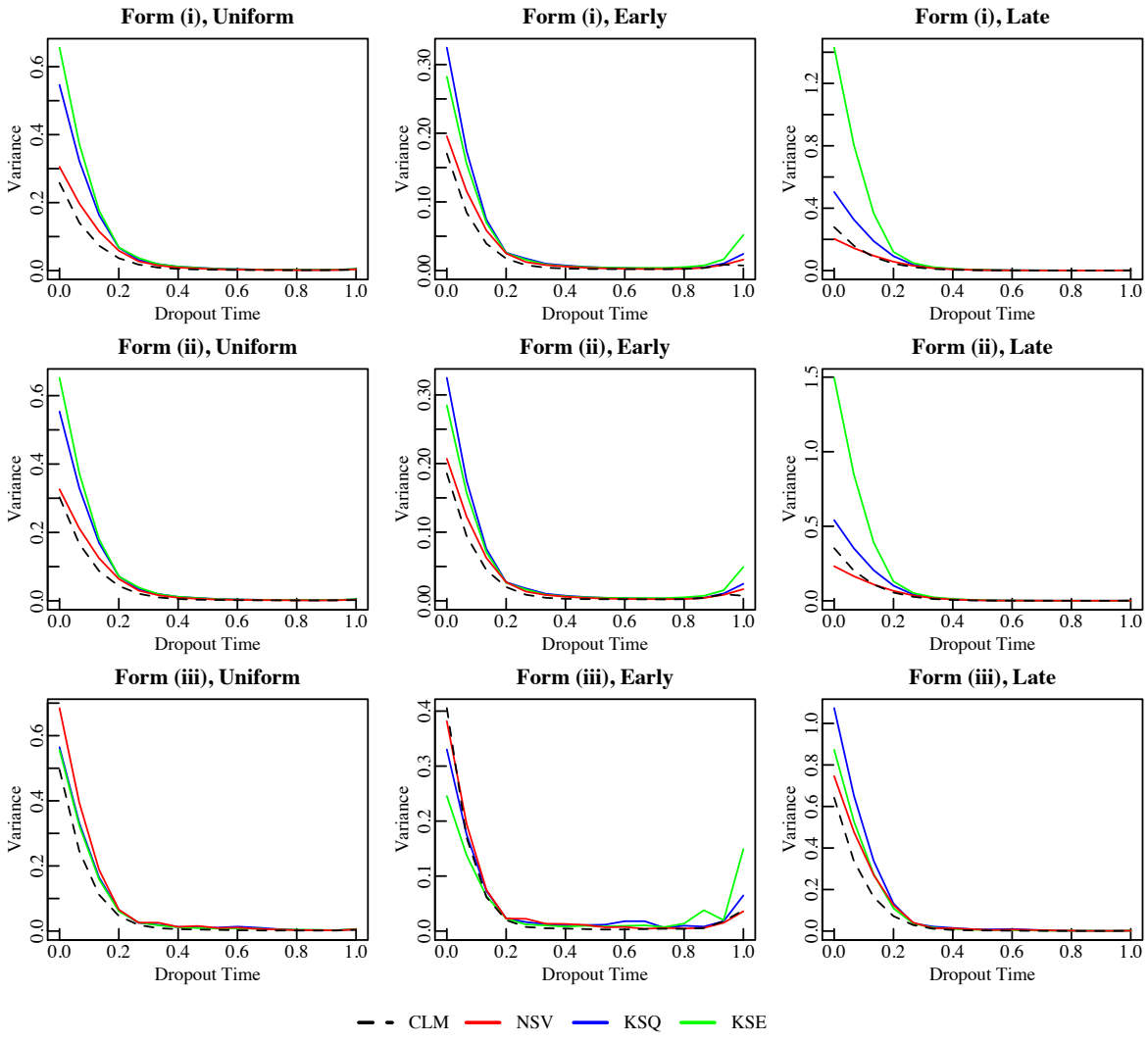


Figure 3.3: Variance for Dropout Time Specific Slopes

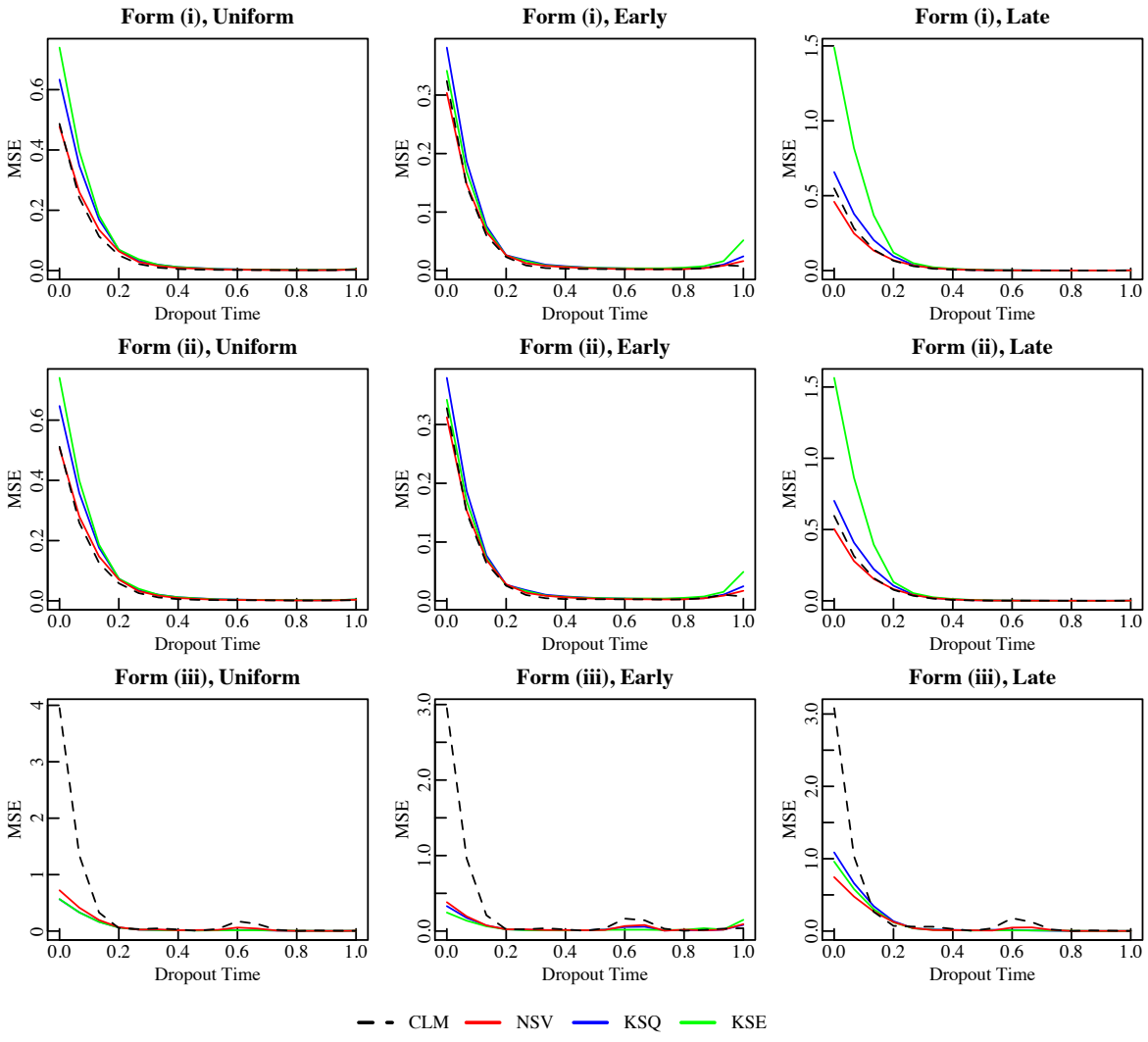


Figure 3.4: MSE for Dropout Time Specific Slopes

CHAPTER IV

APPLICATION TO THE AIEDRP DATASET

The Acute Infection and Early Disease Research Program (AIEDRP) was a multicenter, observational cohort study of subjects acutely and recently infected with HIV at enrollment. Subjects were enrolled between 1997 and 2007 at multiple sites throughout the United States, Australia, Canada, and Brazil. The study closed in 2008. Subjects in the study were allowed to self-select when and whether to initiate anti-retroviral therapy. The AIEDRP dataset presents an opportunity to study the effect of injection drug use on longitudinal CD4+ T cell count for HIV seropositive subjects who have not yet begun antiretroviral therapy. Since these subjects had not yet begun treatment, steeper declines in CD4+ T cell count over time for injection drug users (IDU) would support the hypothesis that injection drug use accelerates the progression of HIV disease by directly enhancing virus replication and/or impairing immune responses. It is well accepted that immune activation drives disease progression (Deeks et al., 2004; Liu et al., 1997), and limited research has suggested that both HIV seropositive and uninfected IDU have higher levels of immune activation than non injection drug users (NIDU) (Mehandru et al., 2012). IDU may also have more immune activation due to co-infections, such a hepatitis B or C (Kovacs et al., 2008). Investigating HIV disease progression in this untreated population allows for better understanding of the effects of injection drug use on HIV disease progression while eliminating the possibility that differences between IDU and NIDU are simply due to increased lack of adherence to treatment regimens among injection drug users.

Descriptive Statistics

Data from 1,074 AIEDRP subjects who had not begun antiretroviral therapy were evaluated. Seventy-eight subjects were classified as IDU at their baseline visit. Descriptive statistics are presented in Table 1. Enrollment criteria included either negative or indeterminate HIV antibody and viral load of >15,000 copies/mL, positive HIV antibody with documented negative HIV antibody within 12 months of

enrollment, or positive HIV antibody and an optical density of <1.0, as determined by a less sensitive enzyme immunoassay. Subjects were classified as acutely infected if they presented within 2 weeks and recently infected if they presented within between 3 weeks and 12 months of seroconversion. Baseline data were collected on race/ethnicity, age, sex, HIV risk factors, including injection drug use, viral load and CD4+ T cell count. Viral load and CD4+ T cell count were determined again at approximately 2, 4, and 12 weeks, and then every 12 weeks until week 168, and every 24 weeks for the remainder of the study.

Table 4.1: Descriptive Statistics: Mean (SD) or %(N). * indicates a geometric mean and standard deviation.

Variable	Overall	Non-IDU	IDU
Nadir CD4+ T cell count*	501.48 (1.52)	506.32(1.50)	443.63(1.73)
Dropout Time (days)*	355.31 (2.57)	359.77 (2.57)	302.96(2.55)
Age (years)	35.57(8.70)	35.43(8.77)	37.32(7.60)
Injection Drug Use	7.26%(78)	NA	NA
Female	5.96%(64)	5.32%(53)	14.10%(11)
Minority	21.04%(226)	20.98%(209)	21.79%(17)
Started Treatment	37.99%(408)	38.25%(381)	34.62%(27)
Acutely Infected	11.73%(126)	11.55%(115)	14.10%(11)

Potential for Non-Ignorable Dropout

Given our focus on untreated observations, subjects may not have had complete data due to starting treatment (ST) or becoming lost to followup. IDU in the cohort tended to dropout of the study earlier than NIDU (Figure 4.1). In addition, initial analyses of the subject specific trajectories of log(CD4+) over time suggested that those who dropped out of the study earlier tended to have steeper declines in CD4+ T cell count than those who completed the study (Figure 4.2). Since the outcome of interest, the change in CD4+ T cell count over time, is related to dropout time, there is evidence to suggest that data are missing not at random and dropout is non-ignorable. In addition, since there are two potential causes for dropout, we would like to account for dropout reason in the analysis as well.

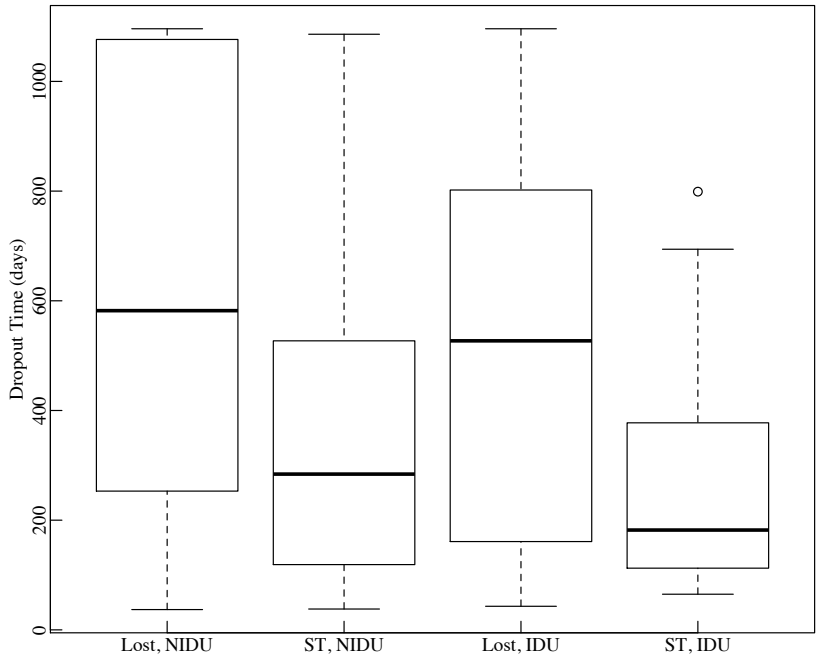


Figure 4.1: Distribution of Dropout Times by IDU and Dropout Reason

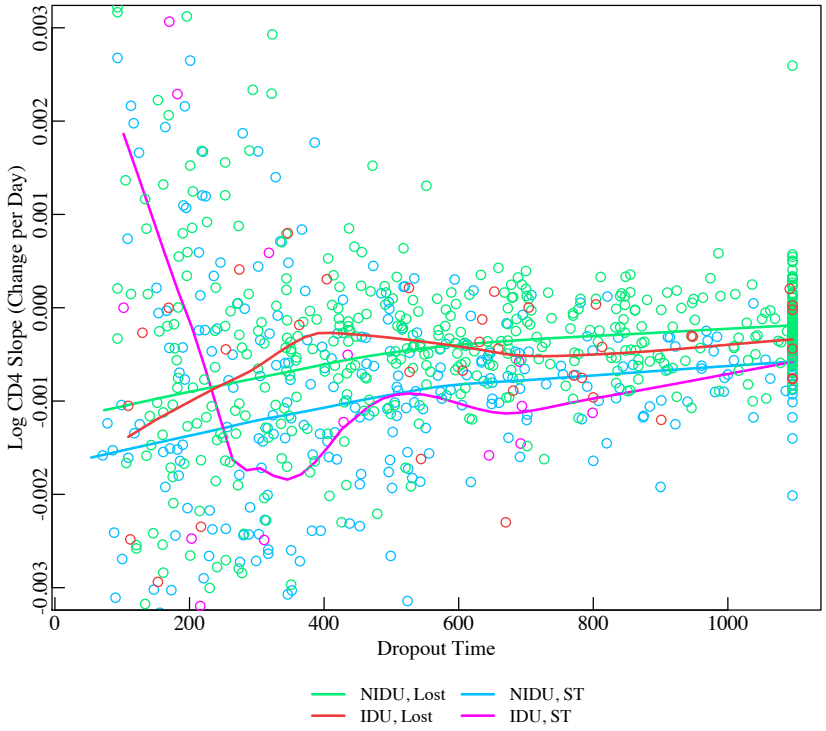


Figure 4.2: Subject Specific OLS Slopes by Dropout Time with Loess Curve. ST indicates a subject started treatment. Lost indicates a subject was lost to followup

Methods

Only those subjects who had more than 5 weeks of observations were considered in the analysis. The reason for this is that it is common for CD4+ T cell counts to acutely decline during the high titer viremia of acute infection, and then subsequently recover a few weeks later in concert with declines in viremia. This recovery in CD4+ T cell counts is thought to be the result of the development of the host immune response to the virus (Schacker et al., 1996). CD4+ T cell counts then decline over time as the disease progresses. Many subjects in both the acute and recent infection groups showed these patterns of early decline and recovery before their long term declines in CD4+ T cell count began. Graphical analyses suggested that these fluctuations ceased for the majority of subjects by 5 weeks after study entry. Therefore only data after 5 weeks of study enrollment were included in the analysis. In order to incorporate information from subjects' first 5 weeks on the study, the minimum or nadir CD4+ T cell count for each subject from the initial 5 week period was calculated and included as a covariate in the analysis. In addition, since few subjects had more than 3 years of untreated followup, the analysis was limited to visits occurring within 3 years of study entry.

Log(CD4+ T cell count) was modeled as a function of injection drug use group and time with dropout reason and time-varying B-spline bases for the slopes in the fixed effects and a random subject-specific intercept and slope. In this model, a different dropout time varying component of the slope was allowed for those who were lost to follow up and those who started treatment. An injection drug use by dropout reason by time interaction was included in the model to allow IDU to have different changes in CD4+ T cell count over time compared to NIDU who dropped out of the study for the same reason. In addition, age, race, sex, acute vs. recent infection, log(nadir CD4+ T cell count), an acute vs. recent infection by time interaction and a log(nadir CD4+ T cell count) by time interaction were included in the model as fixed effect

covariates. The KSE method was used to choose the number and location of knots for the dropout varying slopes. A maximum of 7 degrees of freedom were allowed for each dropout varying effect. For stable slope estimation, a lower boundary offset for each dropout-varying slope was set to day 168, corresponding to approximately 4 observations.

The final model had the following form:

$$\mathbf{Log(CD4+)_i} = \mathbf{1_i}\beta_{g_i h_i 0} + \beta_{g_i h_i 1}\mathbf{t_i} + \beta_{h_i}(u_i)\mathbf{t_i} + \mathbf{C_i}\boldsymbol{\beta_C} + \mathbf{Z_i}\boldsymbol{\alpha_i} + \boldsymbol{\epsilon_i}$$

where h indicates dropout reason, g indicates injection drug use group, u_i is the ith subject's dropout time, $\mathbf{1_i}$ is an n_i x 1 vector of 1's, $\mathbf{t_i}$ is an n_i x 1 vector of observation times, and $\mathbf{C_i}$ is an n_i x p matrix of covariates for the ith subject. $\mathbf{Z_i}$ is the n_i x 2 design matrix for the random effects, $\boldsymbol{\alpha_i}$ is the 2 x 1 vector of random effects, and $\boldsymbol{\epsilon_i}$ is the n_i x 1 vector of error for the ith subject.

1,000 bootstrap samples were drawn from the dataset and modeled as described above. Confidence intervals for the marginal slopes were calculated based on the 2.5th and 97.5th percentiles of the marginal slopes from the bootstrapped datasets. In addition, standard errors were calculated as the standard deviation of the marginals from the bootstrapped datasets, and were used to perform t-tests and calculate p-values based on the normal distribution. In addition, the KSQ, NSV, CLM and a random effects (RE) model that did not account for dropout time or reason were also fit to the data to compare results. A maximum of seven degrees of freedom was allowed for each dropout varying effect in the KSQ and NSV models, and linear, quadratic, and cubic polynomials were considered for the CLM. All models were chosen based on AIC.

Results

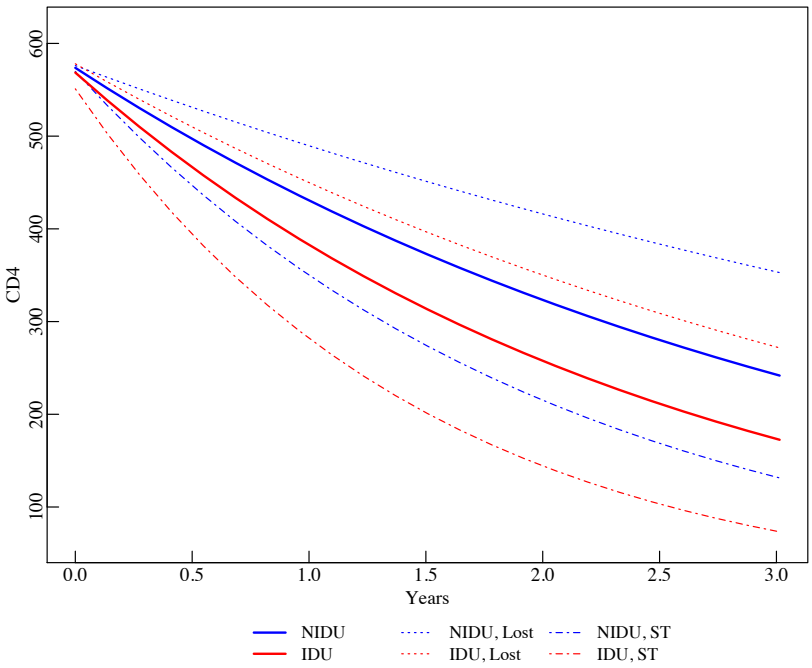
Controlling for age, sex, race, nadir CD4+ T cell count, and acute vs. recent infection, IDU lost to follow up had significantly greater declines in CD4+ T cell count per year compared to NIDU lost to follow up (p=0.03, Tables 4.2-4.3). On average, IDU had 8.42% (95% CI: 1.57%-15.54%) steeper declines in CD4 than NIDU lost to follow up. Assuming recent infection and the median nadir CD4+ T cell count of 500, CD4+ T cell counts declined by 22.2% (95% CI: 15.1%-29.5%) per year for IDU, compared to 15.0% (95%CI: 11.7%-18.3%) per year for NIDU.

IDU who started treatment were not significantly different from NIDU who started treatment (p=0.08), however these conclusions are based on a relatively small sample of IDU who started treatment (N=27), and the magnitude of the difference between IDU and NIDU who started treatment is actually larger than the difference between IDU and NIDU lost to follow up. Assuming recent infection and the median nadir CD4+ T cell count of 500, IDU who started treatment, on average, saw a 48.2% decline in CD4+ T cell count per year (95% CI: 36.8%-60.4%), while NIDU saw a 38.6% decline (95% CI: 32.5%-45.0%). Overall, those who started treatment had significantly steeper declines than those who did not start treatment ($p<0.001$ for IDU, and $p<0.001$ for NIDU).

Similar results were obtained using each of the three natural cubic B spline varying coefficient methods (see Tables 4.6-4.8). All three methods used 3 degrees of freedom for the dropout varying components of the slopes and resulted in models with similar AIC values. The CLM with the lowest AIC had a linear effect of dropout time for both those lost to follow up and those who started treatment, and resulted in a model with a slightly higher AIC and somewhat reduced magnitudes of the estimates of the marginal slopes.

Weighted averages over dropout reason were used to obtain marginal slopes for IDU and NIDU (Tables 4.4-4.5). Assuming recent infection and the median nadir

(a) Recently Infected



(b) Acutely Infected

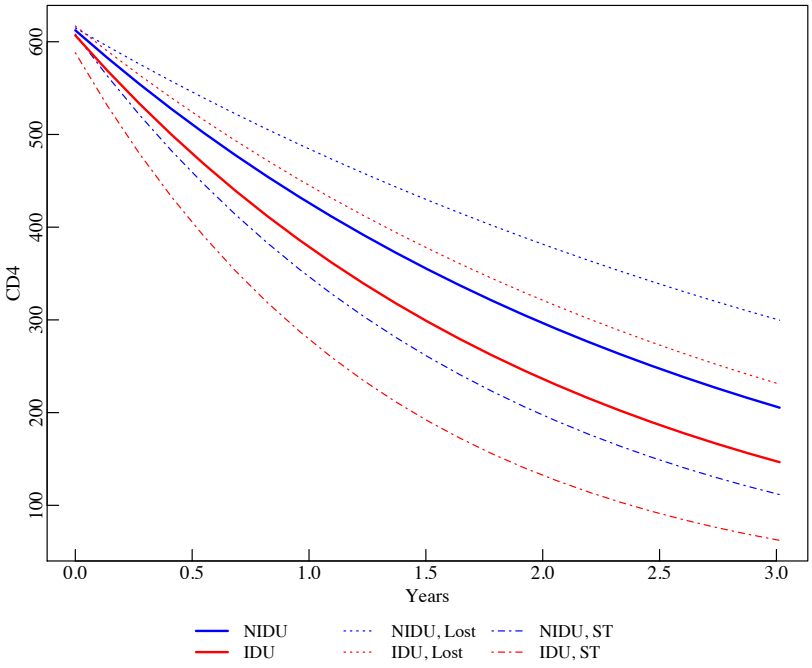
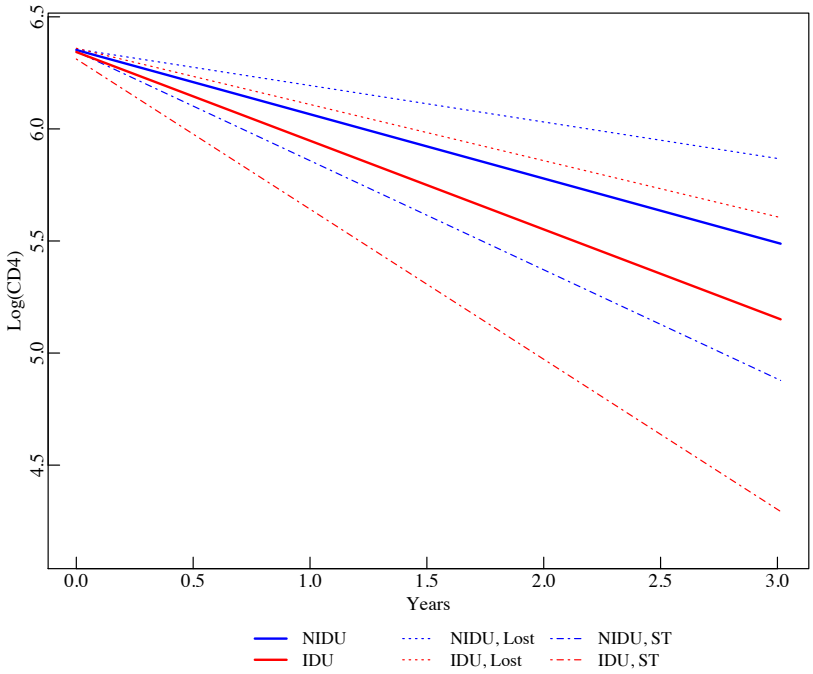


Figure 4.3: Marginal CD4+ T cell count Over Time for a White Male, with a Median Nadir CD4+ T cell count and Median Age

(a) Recently Infected



(b) Acutely Infected

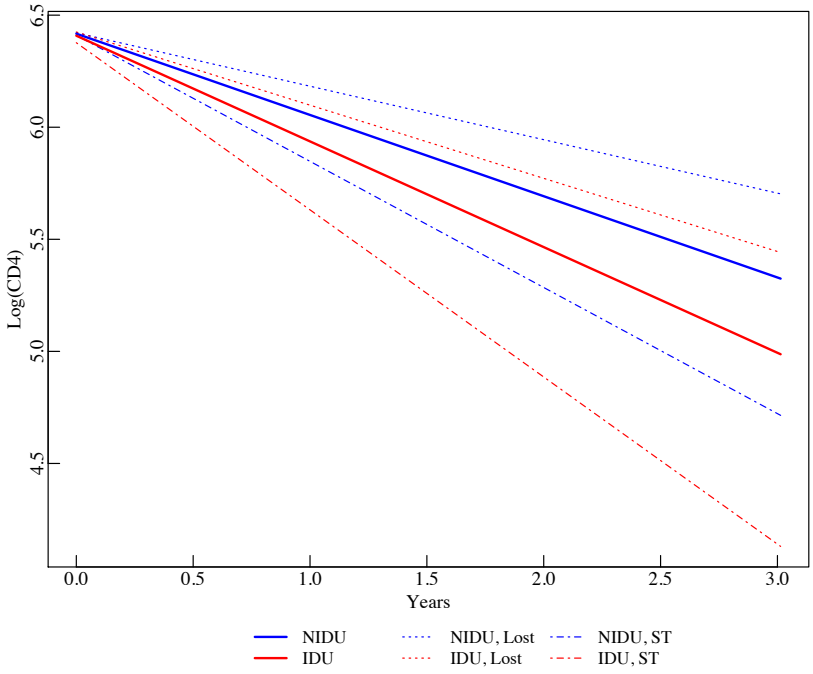


Figure 4.4: Marginal Log(CD4) Over Time for a White Male, with a Median Nadir CD4+ T cell count and Median Age

CD4+ T cell count of 500, IDUs' CD4+ T cell counts declined by 32.7% per year (95% CI: 26.3%-39.6%), compared to 24.9% per year (95% CI: 21.6%-28.4%) for NIDU. IDU had 10.3% steeper declines (95% CI: 3.2%-18.7%) than NIDU (p=0.02).

Using a random effects model that does not account for dropout reason or dropout time (Table 4.10), the magnitude of the change in CD4+ T cell count over time is reduced and the difference between IDU and NIDU is smaller compared to the models that account for dropout. In the random effects model, NIDU are found to have a 14.6% decline in CD4+ T cell count per year (95% CI: 13.1%-16.1%), while IDU have declines of 20.4% per year (95% CI: 15.1%-25.4%). Using the random effects model, IDU are found to have only a 6.77% steeper decline in CD4+ T cell count than NIDU (95% CI: 0.37%-12.8%;p=0.04).

Table 4.2: Dropout Reason Specific Change in Log(CD4) Per Year, by Group with Bootstrap Confidence Intervals and P Values

Group	Slope	SE	Bootstrap 95% CI		T Value	P Value
Lost, NIDU	−0.163	0.020	−0.203	−0.124	−8.237	< 0.001
Lost, IDU	−0.251	0.046	−0.349	−0.164	−5.472	< 0.001
ST, NIDU	−0.487	0.052	−0.597	−0.393	−9.294	< 0.001
ST, IDU	−0.670	0.121	−0.926	−0.459	−5.526	< 0.001
Differences between IDU and NIDU						
Lost IDU - Lost NIDU	−0.088	0.040	−0.169	−0.016	−2.218	0.027
ST IDU - ST NIDU	−0.183	0.105	−0.408	0.009	−1.743	0.081
Differences between ST and Lost						
ST NIDU - Lost NIDU	−0.324	0.056	−0.439	−0.216	−5.742	< 0.001
ST IDU - Lost IDU	−0.507	0.131	−0.693	−0.174	−3.863	< 0.001

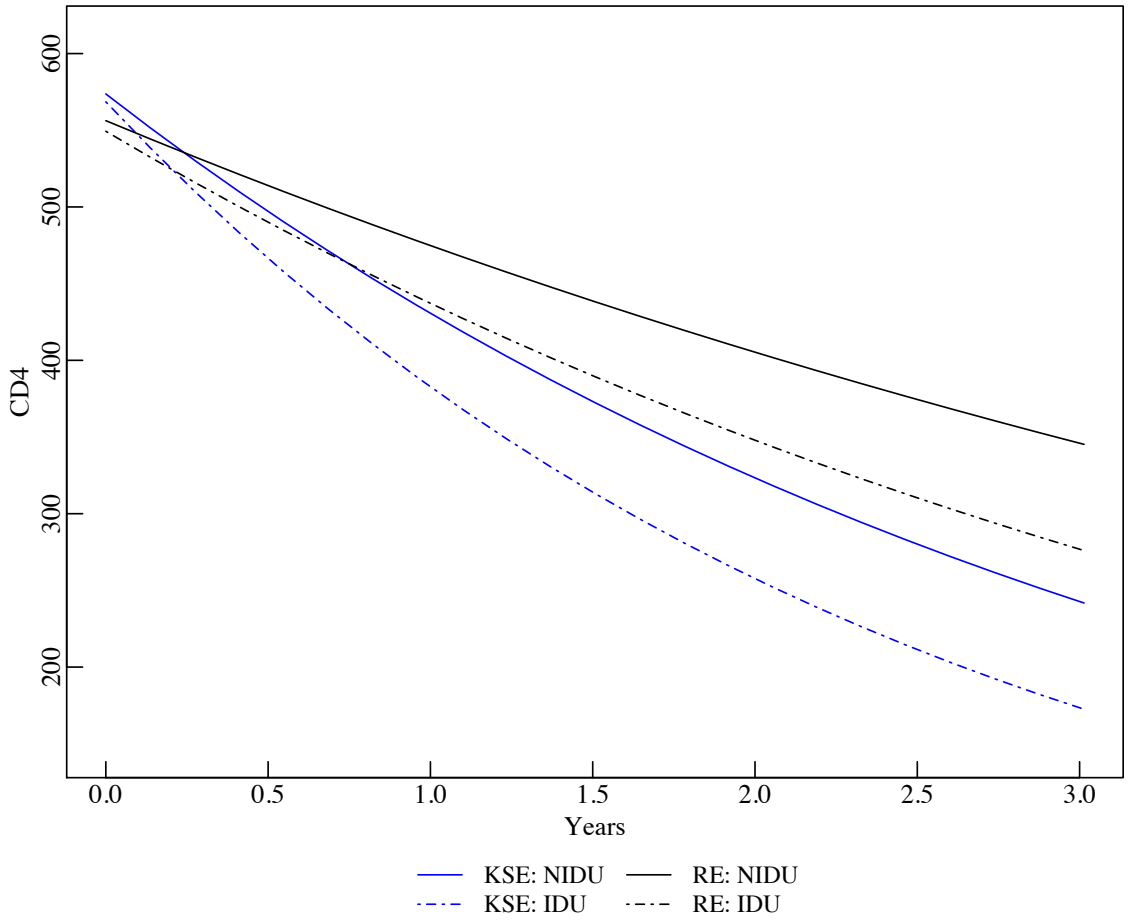


Figure 4.5: Comparison of the KSE and Random Effects Model for Marginal CD4 Over Time for a White Male, with a Median Nadir CD4+ T cell count and Median Age

Table 4.3: Dropout Reason Specific Percent Reduction in CD4 Per Year

Group	Percent Reduction	Bootstrap 95% CI	
Lost, NIDU	−15.02	−18.34	−11.68
Lost, IDU	−22.17	−29.46	−15.12
ST, NIDU	−38.55	−44.97	−32.49
ST, IDU	−48.82	−60.39	−36.81
Differences between IDU and NIDU			
Lost IDU - Lost NIDU	−8.42	−15.54	−1.57
ST IDU - ST NIDU	−16.71	−33.50	0.89
Differences between ST and Lost			
ST NIDU - Lost NIDU	−27.69	−35.52	−19.41
ST IDU - Lost IDU	−39.77	−50.01	−15.96

Table 4.4: Marginal Change in Log(CD4) per Year, Averaged Over Dropout Time and Reason

Group	Slope	SE	Bootstrap 95% CI		T Value	P Value
NIDU	−0.287	0.023	−0.334	−0.243	−12.364	< 0.001
IDU	−0.396	0.051	−0.504	−0.306	−7.826	< 0.001
Difference						
IDU-NIDU	−0.109	0.045	−0.207	−0.033	−2.443	0.015

Table 4.5: Marginal Percent Reduction in CD4 per Year, Averaged Over Dropout Time and Reason

Group	Percent Reduction	95% CI	
NIDU	−24.93	−28.36	−21.60
IDU	−32.68	−39.61	−26.34
Difference			
IDU-NIDU	−10.33	−18.66	−3.23

Table 4.6: Model Fitting Comparison

Method	AIC	Lost		Started Treatment	
		DF	Interior Knots	DF	Interior Knots
KSE	−1044.92	3	260.8	3	351.6
KSQ	−1044.92	3	253.8	3	365.2
NSV	−1043.58	3	701	3	453
CLM	−1037.96	2	NA	2	NA
RE	−927.16	NA	NA	NA	NA

Table 4.7: Model Comparison: Dropout Reason Specific Change in Log(CD4) Per Year

Group	KSE	KSQ	NSV	CLM
Lost, NIDU	−0.163	−0.164	−0.157	−0.135
Lost, IDU	−0.251	−0.252	−0.242	−0.212
ST, NIDU	−0.487	−0.492	−0.487	−0.413
ST, IDU	−0.670	−0.675	−0.670	−0.569
Differences				
Lost: IDU-NIDU	−0.088	−0.088	−0.085	−0.076
ST: IDU-NIDU	−0.183	−0.183	−0.183	−0.156

Table 4.8: Model Comparison: Dropout Reason Specific Percent Reduction in CD4 Per Year

Group	KSE	KSQ	NSV	CLM
Lost, NIDU	−15.02	−15.12	−14.49	−12.65
Lost, IDU	−22.17	−22.30	−21.48	−19.08
ST, NIDU	−38.55	−38.83	−38.55	−33.85
ST, IDU	−48.82	−49.08	−48.82	−43.40
Differences				
Lost: IDU-NIDU	−8.42	−8.47	−8.18	−7.36
ST: IDU-NIDU	−16.71	−16.76	−16.71	−14.44

Table 4.9: Model Comparison: Marginal Change in Log(CD4) Per Year, Averaged over Dropout Time and Reason

Group	KSE	KSQ	NSV	CLM	RE
NIDU	−0.287	−0.289	−0.283	−0.242	−0.158
IDU	−0.396	−0.399	−0.390	−0.336	−0.228
Difference					
IDU-NIDU	−0.109	−0.109	−0.107	−0.094	−0.070

Table 4.10: Model Comparison: Marginal Percent Reduction in CD4 Per Year, Averaged over Dropout Time and Reason

Group	KSE	KSQ	NSV	CLM	RE
NIDU	−24.93	−25.12	−24.64	−21.47	−14.64
IDU	−32.68	−32.88	−32.29	−28.50	−20.42
Difference					
IDU-NIDU	−10.33	−10.36	−10.15	−8.96	−6.77

Table 4.11: Random Effects Model: Change in Log(CD4) per Year

Group	Slope	SE	95% CI		T Value	P Value
NIDU	−0.158	0.009	−0.176	−0.141	−17.693	< 0.001
IDU	−0.228	0.033	−0.293	−0.164	−6.914	< 0.001
Difference						
IDU-NIDU	−0.070	0.034	−0.137	−0.004	−2.071	0.038

Table 4.12: Random Effects Model: Percent Reduction per Year

Group	Percent Reduction	95% CI	
NIDU	−14.64	−16.12	−13.13
IDU	−20.42	−25.41	−15.10
Difference			
IDU-NIDU	−6.77	−12.76	−0.37

CHAPTER V

CONCLUSIONS

We present a simple combinatorial knot selection algorithm to aid in fitting natural cubic B spline varying coefficient models for non-ignorable dropout. The method is conceptually simple and straight forward to implement. Simulation studies show that models including knot selection are more flexible than models that do not search for knot location, and can reduce bias compared to both the NSV and CLM. These methods are also more robust when model assumptions are violated, for example, when the form of the dropout varying slope is not smooth or continuous.

In addition, we extend existing varying coefficient models that account for dropout time to also account for dropout reason. Accounting for dropout reason, as well as time, recognizes that subjects that drop out of a study for different reasons may also have differing outcome trajectories. Population level marginal slopes, as well as dropout reason specific slopes can be calculated to more fully describe the data.

Applying these methods to the AIEDRP dataset, we find that the natural cubic B spline varying coefficient methods (KSE, KSQ, NSV) all result in similar estimates for the effect of injection drug use on longitudinal CD4+ T cell count. These methods show a steeper decline in longitudinal CD4+ T cell count for IDU compared to NIDU. The magnitude of the slopes as well as the difference between injection drug use groups is greater when using methods accounting for dropout compared to a standard mixed effects model. These findings indicate that drug use itself is associated with accelerated HIV disease, independent of noncompliance with antiretroviral medications. These data demonstrate the importance of utilizing methods that account for dropout, as the impact of drug use on CD4+ T cell declines was attenuated in models that failed to account for dropout.

Future directions of research will include extending these models for non-normal outcomes as well as non-linear relationships. In addition, since only a small number of knot locations can be considered using the combinatorial knot selection algorithm,

Bayesian methods, such as reversible jump and birth death Markov chain Monte Carlo, will be explored for determining the number and location of knots for the natural cubic B spline varying coefficient method.

REFERENCES

O. Bagasra, P. Whittle, A. Kajdacsy-Balla, and Lischner H.W. Effects of alcohol ingestion on in vitro susceptibility of peripheral blood mononuclear cells to infection with HIV-1 and on CD4 and CD8 lymphocytes. *Progress in Clinical and Biological Research*, 325:351–358, 1990.

C. Biller and L. Fahrmeir. Bayesian varying-coefficient models using adaptive regression splines. *Statistical Modelling*, 1:195–211, 2000.

D.D. Celetano and G. Lucas. Optimizing treatment outcomes in HIV-infected patients with substance abuse issues. *Clinical Infectious Diseases*, 45:S318–S323, 2007.

R.E. Chaisson, J.C. Keruly, and R.D. Moore. Race, sex, drug use, and pro- gression of human immunodeficiency virus disease. *New England Journal of Medicine*, 333: 751–756, 1995.

C.C. Chao, G. Gekker, S. Hu, and et al. Kappa opioid receptors in human microglia downregulate human immunodeficiency virus 1 expression. *Proceedings of the National Academy of Sciences*, 93:8051–8056, 1996.

R.Y. Chuang, L.F. Chuang, Y. Li, H.F. Kung, and K.F. Jr. Killam. SIV mutations detected in morphine-treated Macaca mulatta following SIVmac239 infection. *Advances in Experimental Medicine and Biology*, 373:175–181, 1995.

M.J. Daniels and J.W. Hogan. *Missing Data in Longitudinal Studies: Strategies for Bayesian Modeling and Sensitivity Analysis*. Chapman and Hall/CRC, Boca Raton, 2008.

S.G. Deeks, C.M. Kitchen, L. Liu, H. Guo, R. Gascon, A.B. Narvaez, P. Hunt, J.N. Martin, J.O. Kahn, J. Levy, M.S. McGrath, and F.M. Hecht. Immune activation set point during early HIV infection predicts subsequent CD4+ t-cell changes independent of viral load. *Blood*, 104(4):942–7, 2004.

D.G.T. Denison, B.K. Mallick, and A.F.M. Smith. Estimation and comparison of changes in the presence of informative right censoring; conditional linear model. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(2): 333–350, 1998.

D.C. DesJarlais, S.R. Friedman, M. Marmor, and et al. Development of AIDS and HIV seroconversion and potential co-factors for T4 cell loss in a cohort of intravenous drug users. *AIDS*, 1:105–111, 1987.

R.M. Donahoe. Multiple ways that drug abuse might influence AIDS progression: clues from a monkey model. *Journal of Neuroimmunology*, 147:28–32, 2004.

R.M. Donahoe and D. Vlahov. Opiates as potential cofactors in progression of HIV-1 infections to AIDS. *Journal of Neuroimmunology*, 83:77–87, 1998.

R.M. Donahoe, L.D. Byrd, H.M. McClure, and et al. Consequences of opiate-dependency in a monkey model of AIDS. *Advances in Experimental Medicine and Biology*, 335:21–28, 1993.

H. Farzadegan, D. Levy, Astemborski, A.J. Saah, D. Vlahov, J.B. Margolick, and N.M.H. Graham. Effect of gender, race, injecting drug use and disease stage on infectious viral load among IDUs and gay men. Abstract: XI International Conference on AIDS, Vancouver, Canada, 1996 .

J.E. Forster, S. MaWhinney, E.L. Ball, and D. Fairclough. A varying-coefficient method for analyzing longitudinal clinical trials data with nonignorable dropout. *Contemporary Clinical Trials*, 33:378–385, 2012.

J.H. Friedman and B.W. Silverman. Flexible parsimonious smoothing and additive modeling. *Technometrics*, 31(1):3–21, 1989.

T. Hastie and R. Tibshirani. Varying-coefficient models. *Journal of the Royal Statistical Society, Series B*, 55:757–796, 1993.

J.W. Hogan and N.M. Laird. Mixture models for the joint distribution of repeated measures and event times. *Statistics in Medicine*, 16:239–257, 1997.

J.W. Hogan, X. Lin, and B. Herman. Mixtures of varying-coefficient models for longitudinal data with discrete or continuous nonignorable dropout. *Biometrics*, 60:854–864, 2004.

F. Kapadia, D. Vlahov, R.M. Donahoe, and G. Friedland. The role of substance abuse in HIV disease progression: Reconciling differences fromlaboratory and epidemiologic investigations. *Clinical Infectious Diseases*, 41(7):1027–1034, 2005.

A. Kovacs, L. Al-Harthi, S. Christensen, W. Mack, M. Cohen, and A. Landay. CD8(+) t cell activation in women coinfected with human immunodeficiency virus type 1 and hepatitis c virus. *Journal of Infectious Diseases*, 197:1402–1407, 2008.

T.F. Kresina, C.W. Flexner, J. Sinclair, and et al. Alcohol use and HIV pharmacotherapy. *AIDS Research and Human Retroviruses*, 18:757–770, 2002.

E. Lanoya, M. Mary-Krausea, P. Tattevinb, R. Dray-Spirac, C. Duvivierd, P. Fischere, Y. Obadiaf, and F. Lert. Predictors identified for losses to follow-up among HIV-seropositive patients. *Journal of Clinical Epidemiology*, 59:829–835, 2006.

H. Liang, H. Wu, and R.J. Carroll. The relationship between virologic and immunologic responses in AIDS clinical research using mixed-effects varying-coefficient models with measurement error. *Biostatistics*, 4(2):297–312, 2003.

Z. Liu, W.G. Cumberland, L.E. Hultin, H.E. Prince, R. Detels, and J.V. Giorgi. Elevated CD38 antigen expression on CD8+ t cells is a stronger marker for the risk of chronic HIV disease progression to AIDS and death in the multicenter AIDS cohort study than CD4+ cell count, soluble immune activation markers, or combinations of HLA-DR and CD38 expression. *Journal of Acquired Immune Deficiency Syndromes*, 16(2):83–92, 1997.

G.M. Lucas, L.W. Cheever, R.E. Chaisson, and R.D. Moore. Detrimental effects of continued illicit drug use on the treatment of HIV-1 infection. *Journal of Acquired Immune Deficiency Syndromes*, 27:251–259, 2001.

J.B. Margolick, A. Munoz, D. Vlahov, and et al. Direct comparison of the relationship between clinical outcome and change in CD4+ lymphocytes in human immunodeficiency virus-positive homosexual men and injecting drug users. *Archives of Internal Medicine*, 154:869–875, 1994.

S. Mehandru, D Garmon, A Walker, and M Markowitz. Injection drug use is associated with significant levels of immune activation in the mucosal tissues and blood. XIX International AIDS Conference (AIDS 2012). Washington, DC, July 22-27, 2012. Poster MOPE022., July 2012.

M. Mori, G.G. Woodworth, and R.F. Woolson. Application of empirical Bayes inference to estimation of rate of change in the presence of informative right censoring. *Statistics in Medicine*, 11(5):621–631, 1992.

D.K. Pauler, S. McCoy, and C. Moinpour. Pattern mixture models for longitudinal quality of life studies in advanced stage disease. *Statistics in Medicine*, 22:795–809, 2003.

P.K. Peterson, G. Gekker, R. Schut, S. Hu, H.H. Jr. Balfour, and Chao C.C. Enhancement of HIV-1 replication by opiates and cocaine: the cytokine connection. *Advances in Experimental Medicine and Biology*, 335:181–188, 1993.

P.K. Peterson, G. Gekker, J.R. Lokensgard, and et al. Kappa-opioid receptor agonist suppression of HIV-1 expression in CD4+ lymphocytes. *Biochemical Pharmacology*, 61:1145–1151, 2001.

P. Pezzotti, N. Galai, D. Vlahov, G. Rezza, CM. Lyles, and J. Astemborski. Direct comparison of time to AIDS and infectious disease death between HIV seroconverter injection drug users in Italy and the United States: results from the ALIVE and ISS studies. *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology*, 20:275–282, 1999.

AM. Rompalo, N. Shah, JB. Margolick, and et al. Evaluation of possible effects of continued drug use on HIV progression among women. *International Journal of STD AIDS*, 15:322–327, 2004.

T. Schacker, A.C. Collier, and J. Hughes. Clinical and epidemiologic features of primary HIV infection. *Annals of Internal Mededicine*, 125(4):257–64, 1996.

S.P. Squinto, D. Mondal, A.L. Block, and O. Prakash. Morphine-induced transactivation of HIV-1 LTR in human neuroblastoma cells. *AIDS Research and Human Retroviruses*, 6:1163–1168, 1990.

Italian Seroconversion Study. Disease progression and early predictors of AIDS in HIV-seroconverted injecting drug users. The Italian Seroconversion Study. *AIDS*, 6:421–426, 1992.

D.P. Tashkin. Evidence implicating cocaine as a possible risk factor for HIV infection. *Journal of Neuroimmunology*, 147:26–27, 2004.

G. Verbeke, B. Spiessens, and E. Lesaffre. Conditional linear mixed models. *The American Statistician*, 55(1):25–34, 2001.

M.L. Veyries, M. Sinet, B. Desforges, and B. Rouveix. Effects of morphine on the pathogenesis of murine Friend retrovirus infection. *Journal of Pharmacology and Experimental Theraputics*, 272:498–504, 1995.

J. von Overbeck, M. Egger, G.D. Smith, and et al. Survival in HIV infection: do sex and category of transmission matter? Swiss HIV Cohort Study. *AIDS*, 8: 1307–1313, 1994.

R. Weber, B. Ledergerber, M. Opravil, W. Siegenthaler, and R. Luthy. Progression of HIV infection in misusers of injected drugs who stop injecting or follow a programme of maintenance treatment with methadone. *BMJ*, 301:1362–1365, 1990.

M.C. Wu and K. Bailey. Estimation and comparison of changes in the presence of informative right censoring; conditional linear model. *Biometrics*, 45:939–955, 1989.

M.C. Wu and R.J. Carroll. Estimation and comparison of changes in the presence of informative censoring by modeling the censoring process. *Biometrics*, 44:175–188, 1988.

APPENDIX A

S AND R CODE FOR SIMULATIONS

S Code for Uniform Dropout Simulated Data

```
# Simulation of Uniform data from the Forster et al Paper
#  New simulations will include 2000 datasets of 400 subjects each
#  1. Forms 1, 2 and 3 under variance scenarios 2 (small for 1000
    datasets) and 3 (large - Hogan-sized for 1000 datasets)
# clean out directory
  remove(ls())
# shut off graphics
  graphics.off()
# files

  dir_"C:/jeri/research/drop_rev_09/sim/data/"
  dir2_"C:/jeri/research/drop_rev_09/sim/programs/"
  file.run_paste(dir2,"data_sim_final.s",sep="")
# program / date / time

        cat("\n\n\n")
        cat("Program: data_sim.s")
        cat("\n\n")
        cat("Program to run create simulated data\n")
        print(date())
        cat("\n\n")

# THIS IS THE NEW SMALL VARIANCE (8/27/09)
# simulate 1000 datasets with small variance, each with 400 subjects -
    this small variance is 1/3 of the medium variance
# sigma-squared = 0.06666667
# d11=0.4; d22=0.01, d12=-0.01

# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:1000)  {
bb_rmvnorm(400, cov=matrix(c(0.4,-0.01,-0.01,0.01), 2))
bcorr_cor.test(bb[,1],bb[,2])
e_rnorm(6400,sd=0.2581988897)
p_rbeta(400,1.5,1.5)
u_rbinom(400,15,p)
drp_u/15
pat_c(1:400)

dat_data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])
  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i_-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I(u
      >=4/15)
```

```
  dat$beta2u.ii_dat$beta2u.i
  dat[dat$drptm>=4/15,]$beta2u.ii_-exp(dat[dat$drptm>=4/15,]$alpha*4/15)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii_as.numeric(dat$drptm>=2/3)

  # bring in timepoints

  tt_data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))

  # "complete" dataset

  dat.cmp_merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e_e

  # calculate outcome

  dat.cmp$y.i_(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii_(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.iii_(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e

  # dataset with dropout

  dat.drp_dat.cmp[dat.cmp$drptm>=dat.cmp$t,]

  # output datasets for use in sim programs:
  # clm_sml.s, clm_lrg.s, nsv_sml.s, nsv_lrg.s, rem_sml.s, rem_lrg.
      s

  file.out_paste(dir,"sim_sml_",i,".dat",sep="")
  write(t(dat.drp),file.out,ncol=13)

}


  file.dat_paste(dir,"sim_sml_1.dat",sep="")
  dat_read.table(file.dat,row.names=NULL,na.strings=".")

  attributes(dat)$names_c("patid","alpha","drp","b1","b2","b2ui","b2uii"
      ,"b2uiii","t","e","yi","yii","yiii")
  print(attributes(dat)$names)
  dat2_dat[,c(1,3)]
  dat2_unique(dat2)
  table(dat2$drp)

# simulate 1000 datasets with medium variance (triple the within-subject
    variance), each with 400 subjects
# sigma-squared = 0.2
# d11=0.4; d22=0.01, d12=-0.01
```

```
# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:1000) {
  bb_rmvnorm(400, cov=matrix(c(0.4,-0.01,-0.01,0.01), 2))
  bcorr_cor.test(bb[,1],bb[,2])
  e_rnorm(6400,sd=0.4472135955)
  p_rbeta(400,1.5,1.5)
  u_rbinom(400,15,p)
  drp_u/15
  pat_c(1:400)

  dat_data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])


  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i_-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I
      (u>=4/15)

  dat$beta2u.ii_dat$beta2u.i
  dat[dat$drptm>=4/15,]$beta2u.ii_-exp(dat[dat$drptm>=4/15,]$alpha*4/
      15)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii_as.numeric(dat$drptm>=2/3)

  # bring in timepoints

  tt_data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))

  # "complete" dataset

  dat.cmp_merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e_e

  # calculate outcome

  dat.cmp$y.i_(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii_(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e
  dat.cmp$y.iii_(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e

  # dataset with dropout

  dat.drp_dat.cmp[dat.cmp$drptm>=dat.cmp$t,]

  # output datasets for use in sim programs:
```

```
    # clm_sml.s, clm_lrg.s, nsv_sml.s, nsv_lrg.s, rem_sml.s, rem_
        lrg.s

    datt_dat.drp[,c(1,3,9,11:13)]
    file.out_paste(dir,"sim_med_",i,".dat",sep="")
    write(t(dat.drp),file.out,ncol=13)
    #write(t(datt),file.out,ncol=6)
  }

  file.dat_paste(dir,"sim_med_1.dat",sep="")
  dat_read.table(file.dat,row.names=NULL,na.strings=".")

  attributes(dat)$names_c("patid","alpha","drp","b1","b2","b2ui","b2uii"
      ,"b2uiii","t","e","yi","yii","yiii")
  print(attributes(dat)$names)
  dat2_dat[,c(1,3)]
  dat2_unique(dat2)
  table(dat2$drp)
```

R Code for Heavy Early and Late Dropout Simulated Data

```
#  Create Simulated Data with Heavy Early and Late Dropout
#  New simulations will include 1000 datasets of 400 subjects each with
    a nonuniform distribution of dropout times: 1 with heavy early
    dropout, 1 with heavy late dropout
#  1. Forms 1, 2 and 3 under variance scenarios 2 (small for 1000
    datasets) and 3 (large - Hogan-sized for 1000 datasets)

# clean out directory
  remove(ls())

# shut off graphics
  graphics.off()

# files

  #dir<-"C:/Users/moorecam/Desktop/simdata/"
  dir<-"/Users/camille/Desktop/simdata/"
  dir2<-dir
  dir3<-dir

#EARLY DROPOUT
# THIS IS THE NEW SMALL VARIANCE (8/27/09)
# simulate 1000 datasets with small variance, each with 400 subjects -
    this small variance is 1/3 of the medium variance
# sigma-squared = 0.06666667
# d11=0.4; d22=0.01, d12=-0.01

# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:100) {
```

```
bb<-mvrnorm(400, mu=c(0,0),Sigma=matrix(c(0.4,-0.01,-0.01,0.01), 2))
bcorr<-cor.test(bb[,1],bb[,2])
e<-rnorm(6400,sd=0.2581988897)
p<-rbeta(400,1.5,3) #early dropout
u<-rbinom(400,15,p)
drp<-u/15
pat<-c(1:400)

dat<-data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])


  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i<-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I(u
      >=4/15)

  dat$beta2u.ii<-dat$beta2u.i
  dat[dat$drptm>=2/3,]$beta2u.ii<-exp(dat[dat$drptm>=2/3,]$alpha*2/3)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii<-as.numeric(dat$drptm>=2/3)

  # bring in timepoints

  tt<-data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))

  # "complete" dataset

  dat.cmp<-merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e<-e


  # calculate outcome

  dat.cmp$y.i<-(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii<-(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat
      .cmp$t)+dat.cmp$e
  dat.cmp$y.iii<-(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e

  # dataset with dropout

  dat.drp<-dat.cmp[dat.cmp$drptm>=dat.cmp$t,]


  # output datasets for use in sim programs:

  file.out<-paste(dir,"sim_early_sml_",i,".dat",sep="")
  write(t(dat.drp),file.out,ncol=13)
```

```
}


# simulate 1000 datasets with medium variance (triple the within-subject
    variance), each with 400 subjects
# sigma-squared = 0.2
# d11=0.4; d22=0.01, d12=-0.01

# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:1000)  {
bb<-mvrnorm(400, mu=c(0,0),Sigma=matrix(c(0.4,-0.01,-0.01,0.01), 2))
bcorr<-cor.test(bb[,1],bb[,2])
e<-rnorm(6400,sd=0.4472135955)
p<-rbeta(400,1.5,3) #early dropout
u<-rbinom(400,15,p)
drp<-u/15
pat<-c(1:400)

dat<-data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])


  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i<-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I(u
      >=4/15)

  dat$beta2u.ii<-dat$beta2u.i
  dat[dat$drptm>=2/3,]$beta2u.ii<--exp(dat[dat$drptm>=2/3,]$alpha*2/3)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii<-as.numeric(dat$drptm>=2/3)

  # bring in timepoints

  tt<-data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))


  # "complete" dataset

  dat.cmp<-merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e<-e


  # calculate outcome

  dat.cmp$y.i<-(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii<-(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat
```

```
    .cmp$t)+dat.cmp$e
  dat.cmp$y.iii<-(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
    dat.cmp$t)+dat.cmp$e


  # dataset with dropout

  dat.drp<-dat.cmp[dat.cmp$drptm>=dat.cmp$t,]

  # output datasets for use in sim programs:

  file.out<-paste(dir,"sim_early_med_",i,".dat",sep="")
  write(t(dat.drp),file.out,ncol=13)

}

#LATE DROPOUT
# THIS IS THE NEW SMALL VARIANCE (8/27/09)
# simulate 1000 datasets with small variance, each with 400 subjects -
    this small variance is 1/3 of the medium variance
# sigma-squared = 0.06666667
# d11=0.4; d22=0.01, d12=-0.01

# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:1000)  {
bb<-mvrnorm(400, mu=c(0,0),Sigma=matrix(c(0.4,-0.01,-0.01,0.01), 2))
bcorr<-cor.test(bb[,1],bb[,2])
e<-rnorm(6400,sd=0.2581988897)
p<-rbeta(400,3,1.5) #late dropout
u<-rbinom(400,15,p)
drp<-u/15
pat<-c(1:400)

dat<-data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])


  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i<-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I(u
    >=4/15)

  dat$beta2u.ii<-dat$beta2u.i
  dat[dat$drptm>=2/3,]$beta2u.ii<-exp(dat[dat$drptm>=2/3,]$alpha*2/3)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii<-as.numeric(dat$drptm>=2/3)

  # bring in timepoints
```

```
  tt<-data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))

  # "complete" dataset

  dat.cmp<-merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e<-e

  # calculate outcome

  dat.cmp$y.i<-(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii<-(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat
      .cmp$t)+dat.cmp$e
  dat.cmp$y.iii<-(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e


  # dataset with dropout

  dat.drp<-dat.cmp[dat.cmp$drptm>=dat.cmp$t,]

  # output datasets for use in sim programs:

  file.out<-paste(dir,"sim_late_sml_",i,".dat",sep="")
  write(t(dat.drp),file.out,ncol=13)

}

# simulate 1000 datasets with medium variance (triple the within-subject
    variance), each with 400 subjects
# sigma-squared = 0.2
# d11=0.4; d22=0.01, d12=-0.01

# set seed
set.seed(42)

# create 1000 datasets
for(i in 1:1000)  {
bb<-mvrnorm(400, mu=c(0,0),Sigma=matrix(c(0.4,-0.01,-0.01,0.01), 2))
bcorr<-cor.test(bb[,1],bb[,2])
e<-rnorm(6400,sd=0.4472135955)
p<-rbeta(400,3,1.5) #late dropout
u<-rbinom(400,15,p)
drp<-u/15
pat<-c(1:400)

dat<-data.frame(patid=pat,alpha=-4,drptm=drp,b1=bb[,1],b2=bb[,2])

  # first functional form Beta2(u)=-exp(-4u)

  dat$beta2u.i<-exp(dat$alpha*dat$drptm)

  # second functional form Beta2(u)=-exp(-4u)I(u<4/15) - exp(-4*4/15)I(u
```

```
    >=4/15)

  dat$beta2u.ii<-dat$beta2u.i
  dat[dat$drptm>=2/3,]$beta2u.ii<-exp(dat[dat$drptm>=2/3,]$alpha*2/3)

  # third functional form Beta2(u)=0*I(u,2/3) + 1*I(u>=2/3)

  dat$beta2u.iii<-as.numeric(dat$drptm>=2/3)

  # bring in timepoints

  tt<-data.frame(patid=rep(1:400,each=16),t=rep(seq(0,1,by=1/15),400))

  # "complete" dataset

  dat.cmp<-merge(dat,tt,by=c("patid"),all=T)
  dat.cmp$e<-e

  # calculate outcome

  dat.cmp$y.i<-(dat.cmp$beta2u.i*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat.
      cmp$t)+dat.cmp$e
  dat.cmp$y.ii<-(dat.cmp$beta2u.ii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*dat
      .cmp$t)+dat.cmp$e
  dat.cmp$y.iii<-(dat.cmp$beta2u.iii*dat.cmp$t)+dat.cmp$b1+(dat.cmp$b2*
      dat.cmp$t)+dat.cmp$e

  # dataset with dropout

  dat.drp<-dat.cmp[dat.cmp$drptm>=dat.cmp$t,]
  # output datasets for use in sim programs:

  file.out<-paste(dir,"sim_late_med_",i,".dat",sep="")
  write(t(dat.drp),file.out,ncol=13)

}

file.dat<-paste(dir,"sim_late_med_1.dat",sep="")
dat<-read.table(file.dat,row.names=NULL,na.strings=".")
attributes(dat)$names_c("patid","alpha","drp","b1","b2","b2ui","b2uii","
    b2uiii","t","e","yi","yii","yiii")
  print(attributes(dat)$names)
  dat2_dat[,c(1,3)]
  dat2_unique(dat2)
  table(dat2$drp)
file.out<-paste(dir3,"sim_late_med_1.dat",sep="")
  write.table(dat,file.out)
```

R Code for KSE, KSQ, AND NSV

```
#Fit KSE, KSQ and NSV Models to the Simulated Datasets
# clean out directory, libraries
  remove(list=ls())
```

```
  library(lme4)
  library(snow)
  library(splines)

# directories
  dir <- "C:/Users/moorecam/Desktop/simdata/"
  dir2 <- "C:/Users/moorecam/Dropbox/Masters Paper/newsim_output/"

# settings for simulations
  mech<-"early"    # Can be "early" or "late"
  form <- "yi"     # Can be "yi" "yii" or "yiii"
  varsize <- "sml"  # Can be "med" or "sml" - corresponding datafiles
      must exist
  nSim <- 1000     # Number of datasets, must be consecutively numbered
  offset<-3      # Number 0 (no offset) to 3 (.2)

# FUNCTION FOR DETERMINING KSE DF FOR EACH DATASET

Simulation <- function(i){


  # Read in data and get dataset with one dropout time per subject

  file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",i,".dat",sep="")
  #file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",10,".dat",sep="")
  dat <- read.table(file.dat,row.names=NULL,na.strings=".")
  names(dat) <- c("patid", "alpha", "drptm", "b1", "b2",
                  "b2ui", "b2uii", "b2uiii", "t", "e", "yi", "yii", "yiii")

  #adjust problems with simulated data:
  dat$b2ui<- -1*dat$b2ui
  dat$b2uii<- ifelse(dat$drptm<2/3, -1*dat$b2uii, dat$b2uii)
  dat$yi<-(dat$b2ui*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e
  dat$yii<-(dat$b2uii*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e

  uu<-dat$drptm
  oneper <- dat[,c(1,3,6,7,8)]
  oneper <- unique.data.frame(oneper)
  drps <- sort(unique(oneper$drptm))
  dtimes<-seq(0,1,1/15)

  uuu<-oneper$drptm
  boundary<-range(uuu)

  lowerb<-offset/15
  upperb<-max(uuu)

  boundary[1]<-lowerb    # MOVES BOUNDARY IN TO MINBOUDARY POINT

  ok<-(uuu<=boundary[1])+(uuu>boundary[2])
  uuu<-uuu[ok==0]
  uuu.old<-oneper$drptm

  # Obtain true marginal slope and true dropout varying slopes for form
```

```
  if (form=="yi"){
    truslp<-mean(oneper$b2ui)
    trueslopes <- -exp(-4*dtimes)}
  if (form=="yii"){
    truslp<-mean(oneper$b2uii)
    trueslopes <- -exp(-4*dtimes)
    trueslopes[dtimes>=2/3] <- -exp(-4*2/3)}
  if (form=="yiii"){
    truslp<-mean(oneper$b2uiii)
    trueslopes <- as.numeric(dtimes>=2/3)}

#find model with minimum AIC - knot finding program

  #create dataframe to store knots and AIC values
  aic<-data.frame(df=double(0), knots = character(0), AIC = double(0),
      problems= double(0))

  #create set of candidate knots based on even spacing
  num_candidates<-9       #number of interior candidate knots
  step<-(upperb-lowerb)/(num_candidates+1)
  lowerc<-lowerb+step
  upperc<-upperb-step
  candidates<-seq(lowerc, upperc, step)

# Search for gs with smallest AIC

  GS <- 7 # gs (slope) number of parameters 1 (constant) to 7 (6 df)

  one <- rep(1,length(dat$t))

  for(qq in 1:GS) {
    if (qq==1){ns.gs <- one
      warning <- 0
      tryCatch(
        nsv <- lmer(formula(paste(form, " ~  t: ns.gs + (t|patid) ",sep=
            "")),REML=F,data=dat)
        ,warning = function(warn){warning <<-1}
      )
      aic<-rbind(aic, data.frame(df=1, knots=NA, AIC=ifelse(warning==1,
          NA,AIC(logLik(nsv))), problems=warning))

    }else if(qq>1) {subsets <- combn(candidates, (qq-2))
      for (j in 1:ncol(subsets)){col=subsets[,j]
      nsuu <-ns(uu, knots=col, Boundary.knots=boundary)    #b spline
          transformation
      ns.gs<-cbind(one, nsuu)              #include a 1 for main time
          effect

      warning <- 0
      tryCatch(
        nsv <- lmer(formula(paste(form, " ~  t: ns.gs + (t|patid) ",sep=
            "")),REML=F,data=dat)
        ,warning = function(warn){warning <<-1}
```

```
    )

    aic<-rbind(aic, data.frame(df=qq, knots=toString(subsets[,j]), AIC
        =ifelse(warning==1,NA,AIC(logLik(nsv))), problems=warning))

  }
}
}

aic$knots<-ifelse(aic$df==2, NA, aic$knots)

which.min.aic <- which(aic$AIC==min(aic$AIC, na.rm=T))

# Case with only 1 min.aic
if (length(which.min.aic)==1){
  gs.knots<-aic$knots[which.min.aic]
  gs.df<-aic$df[which.min.aic]  } else {


# Case of min.aic ties
  min.aic<-aic[which.min.aic,]
  sort.aic<-min.aic[with(min.aic, order(df)),]
  min.gs <- sort.aic[1,] #Takes case w/ min param
  gs.knots<-min.gs$knots
  gs.df<-min.gs$df

}



uu.knots<-as.numeric(unlist(strsplit(as.character(gs.knots), split=",
    ")))

lmerwarns <- sum(aic$problems)

# Chosen model: get splines bases & means needed for marginal slope

if(gs.df==1)  {ns.gs <- rep(1,length(dat$t))}

if(gs.df==2)  {ns.gs<-ns(uu,df=1,Boundary.knots=boundary)
  spline<-ns(uu,df=1,Boundary.knots=boundary)
  oneper.nsgs <- as.matrix(dat[,c(1,3)])
  ns.gs.all<-cbind(oneper.nsgs,ns.gs)
  ns.gs.unq<-unique.data.frame(as.data.frame(ns.gs.all))
  ns.gs.unq<-as.matrix(ns.gs.unq[,-c(1,2)])

  ns.gs.one <- unique(cbind(1,ns.gs.unq)[order(oneper$drptm),])

  # take the mean of ns(drptm,gs) for estimate calculations
  mn.gs <- apply(ns.gs.unq,2,mean)

  # "stretch" spline matrix: give each pat appropriate spline
  #ns.gs <- cbind(1,stretchMat%*%ns.gs)
```

```
    ns.gs<-cbind(1,ns.gs)
  }

  if(gs.df>2) {
    ns.gs<-ns(uu,knots=uu.knots,Boundary.knots=boundary)
    spline<-ns(uu,knots=uu.knots,Boundary.knots=boundary)

    # Get unique spline values, in order of dropout times
    oneper.nsgs <- as.matrix(dat[,c(1,3)])
    ns.gs.all<-cbind(oneper.nsgs,ns.gs)
    ns.gs.unq<-unique.data.frame(as.data.frame(ns.gs.all))
    ns.gs.unq<-as.matrix(ns.gs.unq[,-c(1,2)])

    ns.gs.one <- unique(cbind(1,ns.gs.unq)[order(oneper$drptm),])

    # take the mean of ns(drptm,gs) for estimate calculations
    mn.gs <- apply(ns.gs.unq,2,mean)

    # "stretch" spline matrix: give each pat appropriate spline
    #ns.gs <- cbind(1,stretchMat%*%ns.gs)

    ns.gs<-cbind(1,ns.gs)
  }

  # Run Chosen NSV model; calculate marginal slope, mse, etc. and pull
      off AIC

  nsv <- lmer(formula(paste(form,"~ t: ns.gs + (t|patid)",sep="")),
      REML=T,data=dat)

  # slope estimates at each dropout time and the marginal estimate
  bb <- 1

  if(gs.df==1) {
    slopes<-rep((attributes(nsv)$fixef[-1] %*% bb),16)
  }
  if(gs.df>1) {
    bb <- c(bb,mn.gs)
    cc<-cbind(1,predict(spline, seq(0, 1, 1/15)))
    slopes <- apply(cc,1,function(f){attributes(nsv)$fixef[-1]%*%f})
  }

  # Marginal slope estimate
  slp <- attributes(nsv)$fixef[-1] %*% bb

  mse <- (trueslopes-slopes)^2
  aic<-AIC(logLik(nsv))

  # Marginal slope output, etc.
  return(list(slopes=slopes, slope=slp, aic=aic, mse=mse, gs=gs.df, tru=
      truslp, warn=lmerwarns, knots=uu.knots))
}

# PARALLEL PROCESSING SHELL
```

```
cl <- makeCluster(4)                # Set up for 4 cores or nodes
clusterExport(cl, c("dir","form","varsize", "mech", "nSim", "offset"))
     # Let each node know about global variables
clusterEvalQ(cl, library(lme4))          # Have each node load the
    library 'lme4'
clusterEvalQ(cl, library(splines))         # Have each node load the
    library 'splines'


  # Runs simulations using function 'Simulation'

#i<-as.list(data.frame(matrix(250:260,ncol=11)))

   i<-as.list(data.frame(matrix(1:nSim,ncol=nSim)))
   system.time(
   results <- parLapply(cl, i, Simulation)
   )

# Stop parallel processing
stopCluster(cl)

# EXTRACT ALL RESULTS AND WRITE TO TABLE

slope.i <- sapply(results, function(f) {f$slope})
slopes.i <- sapply(results, function(f) {f$slopes})
slope.se.i <- sapply(results, function(f) {f$slope.se})
gs.i <- sapply(results, function(f) {f$gs})
mse.i <- sapply(results, function(f) {f$mse})
tru.i <- sapply(results, function(f) {f$tru})
warn.i <- sapply(results, function(f) {f$warn})
aic.i<- sapply(results, function(f) {f$aic})
knots.i<- sapply(results, function(f) {as.list(f$knots)})

mn.slopes<-apply(slopes.i,1,mean)

write.table(cbind(1:nSim,aic.i,gs.i,warn.i,tru.i,slope.i,t(slopes.i),t(
    mse.i)),
            paste(dir2,"o",offset,"_kse_",mech,"_",varsize,"_",form,"_6
                df_sim_results.txt",sep=""))

write.table(mn.slopes,paste(dir2,"o",offset,"_kse_",mech,"_",varsize,"_"
    ,form,"_6df_dvslp_plot.txt",sep=""),row.names=F,col.names=F)


# FUNCTION FOR DETERMINING KSQ DF FOR EACH DATASET

Simulation <- function(i){

  # Read in data and get dataset with one dropout time per subject

  file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",i,".dat",sep="")
  #file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",3,".dat",sep="")
  dat <- read.table(file.dat,row.names=NULL,na.strings=".")
```

```
names(dat) <- c("patid", "alpha", "drptm", "b1", "b2",
              "b2ui", "b2uii", "b2uiii", "t", "e", "yi", "yii", "yiii")

#adjust problems with simulated data:
dat$b2ui<- -1*dat$b2ui
dat$b2uii<- ifelse(dat$drptm<2/3, -1*dat$b2uii, dat$b2uii)
dat$yi<-(dat$b2ui*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e
dat$yii<-(dat$b2uii*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e

uu<-dat$drptm
oneper <- dat[,c(1,3,6,7,8)]
oneper <- unique.data.frame(oneper)
drps <- sort(unique(oneper$drptm))
dtimes<-seq(0,1,1/15)

uuu<-oneper$drptm
boundary<-range(uuu)

lowerb<-offset/15
upperb<-max(uuu)

boundary[1]<-lowerb

ok<-(uuu<=boundary[1])+(uuu>boundary[2])
uuu<-uuu[ok==0]
uuu.old<-oneper$drptm

# Obtain true marginal slope and true dropout varying slopes for form

if (form=="yi"){
  truslp<-mean(oneper$b2ui)
  trueslopes <- -exp(-4*dtimes)}
if (form=="yii"){
  truslp<-mean(oneper$b2uii)
  trueslopes <- -exp(-4*dtimes)
  trueslopes[dtimes>=2/3] <- -exp(-4*2/3)}
if (form=="yiii"){
  truslp<-mean(oneper$b2uiii)
  trueslopes <- as.numeric(dtimes>=2/3)}

#find model with minimum AIC - knot finding program

#create dataframe to store knots and AIC values
aic<-data.frame(df=double(0), knots = character(0), AIC = double(0),
    problems= double(0))

#create set of candidate knots based on quantiles
num_candidates<-9        #number of interior candidate knots
lower<-1/(num_candidates+1) #quantile of lowest possible interior knot
upper<-num_candidates*lower #quantile of highest possible interior
    knot
candidates<-unique(quantile(uuu, probs = seq(lower, upper,lower ), na.
    rm = FALSE, names = TRUE, type = 7))
candidates<-subset(candidates, candidates>lowerb & candidates<max(uuu)
```

```
    )

# Search for gi, gs combination with smallest AIC

  GS <- 7 # gs (slope) number of parameters 1 (constant) to 7 (6 df)

  one <- rep(1,length(dat$t))

  for(qq in 1:GS) {
    if (qq==1){ns.gs <- one
      warning <- 0
      tryCatch(
        nsv <- lmer(formula(paste(form, " ~  t: ns.gs + (t|patid) ",sep=
            "")),REML=F,data=dat)
        ,warning = function(warn){warning <<-1}
      )
      aic<-rbind(aic, data.frame(df=1, knots=NA, AIC=ifelse(warning==1,
          NA,AIC(logLik(nsv))), problems=warning))

    }else if(qq>1) {subsets <- combn(candidates, (qq-2))
      for (j in 1:ncol(subsets)){col=subsets[,j]
      nsuu <-ns(uu, knots=col, Boundary.knots=boundary)    #b spline
          transformation
      ns.gs<-cbind(one, nsuu)                #include a 1 for main time
          effect

      warning <- 0
      tryCatch(
        nsv <- lmer(formula(paste(form, " ~  t: ns.gs + (t|patid) ",sep=
            "")),REML=F,data=dat)
        ,warning = function(warn){warning <<-1}
      )

      aic<-rbind(aic, data.frame(df=qq, knots=toString(subsets[,j]), AIC
          =ifelse(warning==1,NA,AIC(logLik(nsv))), problems=warning))

    }
  }
  }

  aic$knots<-ifelse(aic$df==2, NA, aic$knots)

  which.min.aic <- which(aic$AIC==min(aic$AIC, na.rm=T))

  # Case with only 1 min.aic
  if (length(which.min.aic)==1){
    gs.knots<-aic$knots[which.min.aic]
    gs.df<-aic$df[which.min.aic]  } else {


  # Case of min.aic ties
    min.aic<-aic[which.min.aic,]
    sort.aic<-min.aic[with(min.aic, order(df)),]
    min.gs <- sort.aic[1,] #Takes case w/ min param
```

```
  gs.knots<-min.gs$knots
  gs.df<-min.gs$df

}



uu.knots<-as.numeric(unlist(strsplit(as.character(gs.knots), split=",
    ")))

lmerwarns <- sum(aic$problems)

# Chosen model: get splines bases & means needed for marginal slope

if(gs.df==1)  {ns.gs <- rep(1,length(dat$t))}

if(gs.df==2)  {ns.gs<-ns(uu,df=1,Boundary.knots=boundary)
  spline<-ns(uu,df=1,Boundary.knots=boundary)
  oneper.nsgs <- as.matrix(dat[,c(1,3)])
  ns.gs.all<-cbind(oneper.nsgs,ns.gs)
  ns.gs.unq<-unique.data.frame(as.data.frame(ns.gs.all))
  ns.gs.unq<-as.matrix(ns.gs.unq[,-c(1,2)])

  ns.gs.one <- unique(cbind(1,ns.gs.unq)[order(oneper$drptm),])

  # take the mean of ns(drptm,gs) for estimate calculations
  mn.gs <- apply(ns.gs.unq,2,mean)

  # "stretch" spline matrix: give each pat appropriate spline
  #ns.gs <- cbind(1,stretchMat%*%ns.gs)

  ns.gs<-cbind(1,ns.gs)
}

if(gs.df>2) {
  ns.gs<-ns(uu,knots=uu.knots,Boundary.knots=boundary)
  spline<-ns(uu,knots=uu.knots,Boundary.knots=boundary)

  # Get unique spline values, in order of dropout times
  oneper.nsgs <- as.matrix(dat[,c(1,3)])
  ns.gs.all<-cbind(oneper.nsgs,ns.gs)
  ns.gs.unq<-unique.data.frame(as.data.frame(ns.gs.all))
  ns.gs.unq<-as.matrix(ns.gs.unq[,-c(1,2)])

  ns.gs.one <- unique(cbind(1,ns.gs.unq)[order(oneper$drptm),])

  # take the mean of ns(drptm,gs) for estimate calculations
  mn.gs <- apply(ns.gs.unq,2,mean)

  # "stretch" spline matrix: give each pat appropriate spline
  #ns.gs <- cbind(1,stretchMat%*%ns.gs)

  ns.gs<-cbind(1,ns.gs)
}
```

```
  # Run Chosen NSV model; calculate marginal slope, mse, etc. and pull
      off AIC

  nsv <- lmer(formula(paste(form, " ~ t: ns.gs + (t|patid) ",sep="")),
      REML=T, data=dat)

  # slope estimates at each dropout time and the marginal estimate
  bb <- 1

  if(gs.df==1)  {
    slopes<-rep((attributes(nsv)$fixef[-1] %*% bb),16)
  }
  if(gs.df>1) {
    bb <- c(bb,mn.gs)
    cc<-cbind(1,predict(spline, seq(0, 1, 1/15)))
    slopes <- apply(cc,1,function(f){attributes(nsv)$fixef[-1]%*%f})
  }

  # Marginal slope estimate
  slp <- attributes(nsv)$fixef[-1] %*% bb

  mse <- (trueslopes-slopes)^2
  aic<-AIC(logLik(nsv))

  # Marginal slope output, etc.
  return(list(slopes=slopes, slope=slp, aic=aic, mse=mse, gs=gs.df, tru=
      truslp, warn=lmerwarns, knots=uu.knots))
}


# PARALLEL PROCESSING SHELL

cl <- makeCluster(4)              # Set up for 4 cores or nodes
clusterExport(cl, c("dir","form","varsize", "mech", "nSim", "offset"))
     # Let each node know about global variables
clusterEvalQ(cl, library(lme4))          # Have each node load the
    library 'lme4'
clusterEvalQ(cl, library(splines))         # Have each node load the
    library 'splines'


  # Runs simulations using function 'Simulation'

#i<-as.list(data.frame(matrix(250:260,ncol=11)))

  i<-as.list(data.frame(matrix(1:nSim,ncol=nSim)))
  system.time(
  results <- parLapply(cl, i, Simulation)
  )

# Stop parallel processing
stopCluster(cl)
```

```
# EXTRACT ALL RESULTS AND WRITE TO TABLE

slope.i <- sapply(results, function(f) {f$slope})
slopes.i <- sapply(results, function(f) {f$slopes})
slope.se.i <- sapply(results, function(f) {f$slope.se})
gs.i <- sapply(results, function(f) {f$gs})
mse.i <- sapply(results, function(f) {f$mse})
tru.i <- sapply(results, function(f) {f$tru})
warn.i <- sapply(results, function(f) {f$warn})
aic.i<- sapply(results, function(f) {f$aic})
knots.i<- sapply(results, function(f) {as.list(f$knots)})

mn.slopes<-apply(slopes.i,1,mean)

write.table(cbind(1:nSim,aic.i,gs.i,warn.i,tru.i,slope.i,t(slopes.i),t(
    mse.i)),
            paste(dir2,"o",offset,"_ksq_",mech,"_",varsize,"_",form,"_6
                df_sim_results.txt",sep=""))

write.table(mn.slopes,paste(dir2,"o",offset,"_ksq_",mech,"_",varsize,"_"
    ,form,"_6df_dvslp_plot.txt",sep=""),row.names=F,col.names=F)

#Function for NSV
Simulation <- function(i){

  # Read in data and get dataset with one dropout time per subject

  file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",i,".dat",sep="")
  #file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",1,".dat",sep="")
  dat <- read.table(file.dat,row.names=NULL,na.strings=".")
  names(dat) <- c("patid", "alpha", "drptm", "b1", "b2",
                  "b2ui", "b2uii", "b2uiii", "t", "e", "yi", "yii", "yiii")

  #adjust problems with simulated data:
  dat$b2ui<- -1*dat$b2ui
  dat$b2uii<- ifelse(dat$drptm<2/3, -1*dat$b2uii, dat$b2uii)
  dat$yi<-(dat$b2ui*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e
  dat$yii<-(dat$b2uii*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e

  uu<-dat$drptm
  oneper <- dat[,c(1,3,6,7,8)]
  oneper <- unique.data.frame(oneper)
  drps <- sort(unique(oneper$drptm))
  dtimes<-seq(0,1,1/15)

  uuu<-oneper$drptm
  boundary<-range(uuu)

  MINBOUNDARY<-offset/15

  boundary[1]<-MINBOUNDARY      # MOVES BOUNDARY IN TO MINBOUDARY POINT

  ok<-(uuu<=boundary[1])+(uuu>boundary[2])
  uuu<-uuu[ok==0]
```

```
# Obtain true marginal slope and true dropout varying slopes for form

if (form=="yi"){
  truslp<-mean(oneper$b2ui)
  trueslopes <- -exp(-4*dtimes)}
if (form=="yii"){
  truslp<-mean(oneper$b2uii)
  trueslopes <- -exp(-4*dtimes)
  trueslopes[dtimes>=2/3] <- -exp(-4*2/3)}
if (form=="yiii"){
  truslp<-mean(oneper$b2uiii)
  trueslopes <- as.numeric(dtimes>=2/3)}


# Search for gi, gs combination with smallest AIC

GI <- 1 # gi (intercept) number of parameters 1 (constant = no vc) to
    7 (6 df)
GS <- 7 # gs (slope) number of parameters 1 (constant) to 7 (6 df)

one <- rep(1,length(dat$t))
aic <- bic <- index <- problems <- NULL

for(qq in 1:GS) {
  ns.gs <- one
  if(qq>1) {  #ns.gs <- cbind(1,stretchMat%*%ns(oneper$drptm,qq-1)) #
      can't use stretchMat
       knots.gs<-attributes(ns(uuu,(qq-1),Boundary.knots=boundary))$
           knots
       ns.gs<-cbind(one,ns(uu,(qq-1),Boundary.knots=boundary,knots=
           knots.gs))}

  for(ss in 1:GI) {
    ns.gi <- one
    if(ss>1) {  #ns.gi <- cbind(1,stretchMat%*%ns(oneper$drptm,ss-1))#
        can't use stretchMat
        knots.gi<-attributes(ns(uuu,(ss-1),Boundary.knots=boundary))$
            knots
        ns.gi<-cbind(one,ns(uu,(ss-1),Boundary.knots=boundary,knots=
            knots.gi))}


    warning <- 0
    tryCatch(
      nsv <- lmer(formula(paste(form, " ~ ns.gi + t: ns.gs + (t|patid)
          - 1",sep="")),REML=F,data=dat)
     ,warning = function(warn){warning <<-1}
    )
    problems <- c(problems,warning)
    aic <- c(aic,ifelse(warning==1,NA,AIC(logLik(nsv))))
    #bic <- c(bic,ifelse(warning==1,NA,BIC(logLik(nsv))))
    index <- rbind(index,c(qq,ss))
    nsv<-NULL
```

```
  }
}

which.min.aic <- which(aic==min(aic, na.rm=T))

# Case with only 1 min.aic
if (length(which.min.aic)==1){
  min.gi <- index[which.min.aic,2]
  min.gs <- index[which.min.aic,1]} else {

# Case of min.aic ties
  params <- (index[,1]+index[,2])[which.min.aic] # Params in tied
      cases
  min.gi <- index[which.min.aic[order(params)][1],2] #Takes case w/
      min param
  min.gs <- index[which.min.aic[order(params)][1],1] #Takes case w/
      min param
}

lmerwarns <- sum(problems)

# Chosen model: get splines bases & means needed for marginal slope

gi <- min.gi-1
gs <- min.gs-1

if(min.gi==1) {ns.gi <- rep(1,length(dat$t))}
if(min.gs==1) {ns.gs <- rep(1,length(dat$t))}

if(min.gi>1)  {

  ns.gi.tmp <- ns(oneper$drptm,gi)

  knots.gi<-attributes(ns(uuu,gi,Boundary.knots=boundary))$knots
  ns.gi<-ns(uu,gi,Boundary.knots=boundary,knots=knots.gi)

  # take the mean of ns(drptm,gi) for estimate calculations

  oneper.nsgi <- as.matrix(dat[,c(1,3)])
  ns.gi.all<-cbind(oneper.nsgi,ns.gi)
  ns.gi.unq<-unique.data.frame(as.data.frame(ns.gi.all))
  ns.gi.unq<-as.matrix(ns.gi.unq[,-c(1,2)])

  mn.gi <- apply(ns.gi.unq,2,mean)

  # "stretch" spline matrix: give each pat appropriate spline
  #ns.gi <- cbind(1,stretchMat%*%ns.gi)
  ns.gi<-cbind(1,ns.gi)
}
if(min.gs>1)  {
  ns.gs <- ns(oneper$drptm,gs)

  knots.gs<-attributes(ns(uuu,gs,Boundary.knots=boundary))$knots
  ns.gs<-ns(uu,gs,Boundary.knots=boundary,knots=knots.gs)
```

```
    # Get unique spline values, in order of dropout times
    oneper.nsgs <- as.matrix(dat[,c(1,3)])
    ns.gs.all<-cbind(oneper.nsgs,ns.gs)
    ns.gs.unq<-unique.data.frame(as.data.frame(ns.gs.all))
    ns.gs.unq<-as.matrix(ns.gs.unq[,-c(1,2)])

    ns.gs.one <- unique(cbind(1,ns.gs.unq)[order(oneper$drptm),])

    # take the mean of ns(drptm,gs) for estimate calculations
    mn.gs <- apply(ns.gs.unq,2,mean)

    # "stretch" spline matrix: give each pat appropriate spline
    #ns.gs <- cbind(1,stretchMat%*%ns.gs)

    ns.gs<-cbind(1,ns.gs)
  }

  # Run Chosen NSV model; calculate marginal slope, mse, etc. and pull
      off AIC

  nsv <- lmer(formula(paste(form, " ~ ns.gi + t:ns.gs + (t|patid) - 1",
      sep="")),REML=T,data=dat)

  # slope estimates at each dropout time and the marginal estimate
  bb <- c(rep(0,min.gi),1)
  if(min.gs==1) {
    slopes<-rep((attributes(nsv)$fixef %*% bb),16)
  }
  if(min.gs>1)  {
    bb <- c(bb,mn.gs)
    cc <- cbind(rep(0,16),rep(1,16),ns(seq(0,1,1/15),gs,Boundary.knots=
        boundary,knots=knots.gs))
    slopes <- apply(cc,1,function(f){attributes(nsv)$fixef%*%f})
  }

  # Marginal slope estimate
  slp <- attributes(nsv)$fixef %*% bb

  mse <- (trueslopes-slopes)^2
  aic<-AIC(logLik(nsv))

  # Marginal slope output, etc.
  return(list(slopes=slopes, slope=slp, aic=aic, mse=mse, gi=gi, gs=gs,
      tru=truslp, warn=lmerwarns))
}

# PARALLEL PROCESSING SHELL

cl <- makeCluster(4)              # Set up for 4 cores or nodes
clusterExport(cl, c("dir","form","varsize", "mech", "nSim", "offset"))
     # Let each node know about global variables
clusterEvalQ(cl, library(lme4))          # Have each node load the
```

```
    library 'lme4'
clusterEvalQ(cl, library(splines))          # Have each node load the
    library 'splines'


  # Runs simulations using function 'Simulation'

#i<-as.list(data.frame(matrix(250:260,ncol=11)))

  i<-as.list(data.frame(matrix(1:nSim,ncol=nSim)))
  system.time(
  results <- parLapply(cl, i, Simulation)
  )

# Stop parallel processing
stopCluster(cl)

# EXTRACT ALL RESULTS AND WRITE TO TABLE

slope.i <- sapply(results, function(f) {f$slope})
slopes.i <- sapply(results, function(f) {f$slopes})
slope.se.i <- sapply(results, function(f) {f$slope.se})
gi.i <- sapply(results, function(f) {f$gi})
gs.i <- sapply(results, function(f) {f$gs})
mse.i <- sapply(results, function(f) {f$mse})
tru.i <- sapply(results, function(f) {f$tru})
warn.i <- sapply(results, function(f) {f$warn})
aic.i<- sapply(results, function(f) {f$aic})

mn.slopes<-apply(slopes.i,1,mean)

write.table(cbind(1:nSim,aic.i,gi.i,gs.i,warn.i,tru.i,slope.i,t(slopes.i
    ),t(mse.i)),
            paste(dir2,"o",offset,"_nsv_",mech,"_",varsize,"_",form,"_B4
                _6df_sim_results.txt",sep=""))

write.table(mn.slopes,paste(dir2,"o",offset,"_nsv_",mech,"_",varsize,"_"
    ,form,"_B4_6df_dvslp_plot.txt",sep=""),row.names=F,col.names=F)
```

R Code for CLM

```
# Run CLM Best for all datasets, variances, and slope forms

# GENERAL SETTINGS: cleanup, libraries,
  remove(list=ls())
  library(lme4)
  library(snow)

# files

#dir <- "C:/Documents and Settings/HernandE/Desktop/Erika Methods/data/"
#dir3 <- "C:/Documents and Settings/HernandE/Desktop/Erika Methods/clm_
    simulations_r_erika_mod/MySim/"
```

```
dir <- "C:/Users/moorecam/Desktop/simdata/"
dir2 <- "C:/Users/moorecam/Dropbox/Masters Paper/newsim_output/"

# settings for simulations

form <- "yiii"     # Can be "yi" "yii" or "yiii"
varsize <- "sml"     # Can be "med" or "sml" - corresponding datafiles
    must exist
mech<-'late'     #can be 'early' 'late' or 'reg'
offset<-3      # #/15
nSim <- 1000     # Number of datasets, must be consecutively numbered

if (mech == 'reg'){
  dir <- "C:/Users/moorecam/Desktop/sim/data/"}

# FUNCTION FOR DETERMINING BEST CLM FOR EACH DATASET

Simulation <- function(i){

  # Read in data and calculate dropout-time^2 and dropout-time^3

if (mech == 'reg'){
  file.dat <- paste(dir,"sim_",varsize,"_",i,".dat",sep="")
#file.dat <- paste(dir,"sim_",varsize,"_1.dat",sep="")


  dat <- read.table(file.dat,row.names=NULL,na.strings=".")
  names(dat) <- c("patid", "alpha", "drptm", "b1", "b2", "b2ui", "b2uii"
      , "b2uiii",
         "t", "e", "yi", "yii", "yiii")
} else if (mech != 'reg'){
  file.dat <- paste(dir,"sim_",mech,"_",varsize,"_",i,".dat",sep="")
  dat <- read.table(file.dat,row.names=NULL,na.strings=".")
  names(dat) <- c("patid", "alpha", "drptm", "b1", "b2",
              "b2ui", "b2uii", "b2uiii", "t", "e", "yi", "yii", "yiii")

  #adjust problems with simulated data:
  dat$b2ui<- -1*dat$b2ui
  dat$b2uii<- ifelse(dat$drptm<2/3, -1*dat$b2uii, dat$b2uii)
  dat$yi<-(dat$b2ui*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e
  dat$yii<-(dat$b2uii*dat$t)+dat$b1+(dat$b2*dat$t)+dat$e
}



  dat$drp.sqr <- dat$drptm^2
  dat$drp.cub <- dat$drptm^3

  # Fit linear, quadratic, and cubic CLM's
  # LL, SS, CC, LS, SL, CS, SC, LC, CL

  clm <- clm2 <- clm3 <- clm4 <- clm5 <- clm6 <- clm7 <- clm8 <-
      clm9 <- NULL
```

```
form1 <- formula(paste(form, " ~ t + t: drptm + (t|patid)",sep=""))
form2 <- formula(paste(form, " ~ t + t:drptm + t:drp.sqr + (t|patid)",
    sep=""))
form3 <- formula(paste(form, " ~ t + t:drptm + t:drp.sqr + t:drp.cub +
     (t|patid)",sep=""))

clm <- lmer(form1, REML=F, data=dat)
clm2 <- lmer(form2, REML=F, data=dat)
clm3 <- lmer(form3, REML=F, data=dat)

# Choose CLM with best fit
ano <- anova(clm,clm2,clm3)

if(ano$"Pr"[3]<0.05){Best <- 3; clmBest <-clm3}
if(ano$"Pr"[3]>=0.05 & ano$"Pr"[2]<0.05){Best <- 2; clmBest <-clm2}
if(ano$"Pr"[3]>=0.05 & ano$"Pr"[2]>0.05){Best <- 1; clmBest <-clm}

# Calculations associated with best CLM fit

oneper <- dat[,c(1,3,6,14,15)]
oneper <- unique.data.frame(oneper)
mndrp <- mean(oneper$drptm)
mndrps <- mean(oneper$drp.sqr)
mndrpc <- mean(oneper$drp.cub)

drps <- c
    (0,0.06666667,0.13333333,0.20000000,0.26666667,0.33333333,0.4,
       0.46666667,0.53333333,0.6,0.66666667,0.73333333,0.8,
       0.86666667,0.93333333,1)
drps.sqr <- drps^2
drps.cub <- drps^3


# Make vectors compatible with all 3 scenarios
b <- c(0,1,mndrp,mndrps,mndrpc)
drp <- cbind(0,1,drps,drps.sqr,drps.cub)
vb <- rbind(c(1, 2*mndrp, 2*mndrps,        2*mndrpc        ),
     c(0, mndrp^2, 2*mndrp*mndrps, 2*mndrp*mndrpc ),
     c(0, 0,       mndrps^2,       2*mndrps*mndrpc),
     c(0, 0,       0,              mndrpc^2        ))

# Pare down vectors to correspond to Best CLM
b <- b[1:(2+Best)]
drp <- drp[,1:(2+Best)]
vb <- vb[1:(1+Best),1:(1+Best)]

# Calculate drpt-varying slopes, marginal slope, and std. err. for
    marginal slope
slopes <- apply(drp,1,function(f){attributes(clmBest)$fixef%*%f})
slp <- attributes(clmBest)$fixef %*% b
vc <- vcov(clmBest)[-1,-1]
slp.se <- sqrt(sum(vb*vc))
```

```
  # MSE calculations and slopes output to file

  dslp <- data.frame(drps,slopes)

  if (form=="yi"){
    dslp$true <- -exp(-4*dslp$drps)}
  if (form=="yii"){
    dslp$true <- -exp(-4*dslp$drps)
    dslp[dslp$drps>=2/3,]$true <- -exp(-4*2/3)}
  if (form=="yiii"){
    dslp$true <- as.numeric(dslp$drps>=2/3)}

  dslp$mse <- (dslp$true-dslp$slopes)^2

  #file.sas <- paste(dir2,"drpslp_clm_",varsize,"_",form,"_",i,".dat",
      sep="")
  #write(t(dslp),file.sas,ncol=4)

  # Marginal slope output, etc.
  return(list(slopes=slopes, slope=slp, slope.se=slp.se, mse=dslp$mse,
      best=Best))
}

# PARALLEL PROCESSING SHELL

cl <- makeCluster(4)                # Set up for 4 cores or nodes
clusterExport(cl, c("dir","dir2","form","varsize", "offset", "mech"))
   # Let each node know about global variables
clusterEvalQ(cl, library(lme4))          # Have each node load the
   library 'lme4'


  # Runs simulations using function 'Simulation'

  i<-as.list(data.frame(matrix(1:nSim,ncol=nSim)))
  system.time(
  results <- parLapply(cl, i, Simulation)
  )

# Stop parallel processing
stopCluster(cl)

# EXTRACT ALL RESULTS AND WRITE TO TABLE

slope.i <- sapply(results, function(f) {f$slope})
slopes.i <- sapply(results, function(f) {f$slopes})
slope.se.i <- sapply(results, function(f) {f$slope.se})
model.i <- sapply(results, function(f) {f$best})
mse.i <- sapply(results, function(f) {f$mse})
mn.slopes<-apply(slopes.i,1,mean)

write.table(cbind(1:nSim,model.i,slope.i,slope.se.i,t(slopes.i),t(mse.i)
    ),
            paste(dir2,"o",offset,"_clm_",mech,"_",varsize,"_",form,"_
```

```
                    sim_results.txt",sep=""))

write.table(mn.slopes,paste(dir2,"o",offset,"_clm_",mech,"_",varsize,"_"
    ,form,"_B4_6df_dvslp_plot.txt",sep=""),row.names=F,col.names=F)
```

APPENDIX B

R CODE FOR AIEDRP ANALYSIS

DESCRIPTIVE STATISTICS AND FITTING THE KSE, KSQ, NSV AND CLM

```
#Final Models
  library(lme4)
  library(sampling)
  library(snow)
  library(nlme)


  dir <- "/Users/camille/Dropbox/Masters Paper/aiedrp/"
  dir2 <- "/Users/camille/Desktop/bootdata/"

  # make changes when necessary
  dir3 <- "C:/Users/moorecam/"
  dir4 <- "C:/Users/moorecam/"

#Descriptive Statistics
oneper<-unique(cbind(dat$patid, dat$age, dat$nadir, dat$uu, dat$female,
    dat$minority, dat$acute, dat$NEWtrt, dat$idu))
nidu<-oneper[oneper[,9]==0,]
idu<-oneper[oneper[,9]==1,]
summary(oneper)
summary(nidu)
summary(idu)
#Geometric Means
exp(mean((oneper[,3])))
exp(sd((oneper[,3])))
exp(mean(log(oneper[,4])))
exp(sd(log(oneper[,4])))
exp(mean((nidu[,3])))
exp(sd((nidu[,3])))
exp(mean(log(nidu[,4])))
exp(sd(log(nidu[,4])))
exp(mean((idu[,3])))
exp(sd((idu[,3])))
exp(mean(log(idu[,4])))
exp(sd(log(idu[,4])))

#boxplot of dropout times
boxplot(oneper[,4]~oneper[,8]*oneper[,9])

#For KSE
#read in dataset
file.dat<-paste(dir,"aiedrp_final.dat",sep="")
dat<-read.table(file.dat, header=T)
dat$group1<-ifelse(dat$idu==0 & dat$NEWtrt==1, 1, 0)
dat$group2<-ifelse(dat$idu==1 & dat$NEWtrt==1, 1, 0)
dat$group3<-ifelse(dat$idu==0 & dat$NEWtrt==0, 1, 0)
dat$group4<-ifelse(dat$idu==1 & dat$NEWtrt==0, 1, 0)
```

```
dat$finid<-dat$patid

  #Subset data by group
  set1<-subset(dat, dat$NEWtrt==0)
  set2<-subset(dat, dat$NEWtrt==1)

  #Determine Knots for Each Set
  df.list<-rep(NA,2)
  uu.knots.list<-rep(NA,2)
  boundary.list<-cbind(rep(NA,2), rep(NA,2))

  for (k in 1:2){set<-get(paste("set",k, sep=""))

  oneper <- cbind(set$finid, set$uu)
  oneper <- unique.data.frame(oneper)
  uuu<-oneper[,2]


  boundary<-range(uuu)
  lowerb<-168
  upperb<-max(uuu)

  boundary[1]<-lowerb

  ok<-(uuu<=boundary[1])+(uuu>boundary[2])
  uuu<-uuu[ok==0]

  #create dataframe to store knots and AIC values
  aic<-data.frame(knots = character(0), AIC = double(0), problems=
      double(0))

  #create set of candidate knots based on even spacing
  num_candidates<-9       #number of interior candidate knots
  step<-(upperb-lowerb)/(num_candidates+1)
  lowerc<-lowerb+step
  upperc<-upperb-step
  candidates<-seq(lowerc, upperc, step)

  # Search for gi, gs combination with smallest AIC

  GS <- 7 # gs (slope) number of parameters 1 (constant) to 7 (6 df)
  one <- rep(1,length(set$day))

  for(qq in 1:GS) {
    if (qq==1){ns.gs <- one
      error<-0
      warning <- 0
      tryCatch(
        nsv <-lme(logcd4 ~ nadir + acute + minority + age + female + idu
            + idu:day + nadir:day + acute:day + day:ns.gs, random=~day|
            finid,method='ML',data=set)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
      )
```

```
      probs<-warning+error
      aic<-rbind(aic, data.frame(df=1, knots=NA, AIC=ifelse(probs!=0,NA,
          AIC(logLik(nsv))), problems=probs))

    }else if(qq>1) {subsets <- combn(candidates, (qq-2))
      for (j in 1:ncol(subsets)){col=subsets[,j]
      nsuu <-ns(set$uu, knots=col, Boundary.knots=boundary)    #b spline
          transformation
      ns.gs<-cbind(one, nsuu)                #include a 1 for main time
          effect

      error<-0
      warning <- 0
      tryCatch(
        nsv <-lme(logcd4 ~ nadir + acute + minority + age + female + idu
            + idu:day + nadir:day + acute:day + day:ns.gs, random=~day|
            finid,method='ML',data=set)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
      )
      probs<-warning+error
      aic<-rbind(aic, data.frame(df=qq, knots=toString(subsets[,j]), AIC
          =ifelse(probs !=0,NA,AIC(logLik(nsv))), problems=probs))

    }
  }
  }

  aic$knots<-ifelse(aic$df==2, NA, aic$knots)

  which.min.aic <- which(aic$AIC==min(aic$AIC, na.rm=T))

  # Case with only 1 min.aic
  if (length(which.min.aic)==1){
    gs.knots<-aic$knots[which.min.aic]
    gs.df<-aic$df[which.min.aic]   } else {


  # Case of min.aic ties
    min.aic<-aic[which.min.aic,]
    sort.aic<-min.aic[with(min.aic, order(df)),]
    min.gs <- sort.aic[1,] #Takes case w/ min param
    gs.knots<-min.gs$knots
    gs.df<-min.gs$df

  }

uu.knots.list[i]<-gs.knots
boundary.list[i,]<-boundary
df.list[i]<-gs.df
}

#Fit the final model with all the covariates
uu.knots1<-(uu.knots.list[1])
```

```
uu.knots2<-(uu.knots.list[2])

#uu.knots1<-260.8
#uu.knots1<-351.6

if (is.na(uu.knots1)==T | is.na(uu.knots2)==T){coeff=rep(NA, 19)
  pvalues=rep(NA,19)
  m.st.nidu=NA
  m.st.idu= NA
  m.lost.nidu= NA
  m.lost.idu= NA
  diff.st= NA
  diff.lost= NA
  knots.st=NA
  knots.lost=NA
  mnadir=NA
  uu.st.nidu=NA
  uu.st.idu=NA
  uu.lost.nidu=NA
  uu.lost.idu=NA

  }else{one<-rep(1, length(dat$uu))
gs.1<-ns(dat$uu, knots=uu.knots1, Boundary.knots=boundary.list[1,])  #
gs.1<-cbind(one, gs.1)
gs.1[,1]<-ifelse(dat$NEWtrt==0, gs.1[,1], 0)
gs.1[,2]<-ifelse(dat$NEWtrt==0, gs.1[,2], 0)
gs.1[,3]<-ifelse(dat$NEWtrt==0, gs.1[,3], 0)

gs.2<-ns(dat$uu, knots=uu.knots2, Boundary.knots=boundary.list[2,])
gs.2<-cbind(one, gs.2)
gs.2[,1]<-ifelse(dat$NEWtrt==1, gs.2[,1], 0)
gs.2[,2]<-ifelse(dat$NEWtrt==1, gs.2[,2], 0)
gs.2[,3]<-ifelse(dat$NEWtrt==1, gs.2[,3], 0)

warning<-0
error<-0

tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
    acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
      )
       probs<-warning+error

if (probs>0){warning<-0
error<-0
  tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female
      + group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
      acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat,
      control=lmeControl(opt = "optim"))
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
       )
```

```
        probs<-warning+error
}

if (probs==0){
#Estimate the marginal slopes for the 4 groups...need to put in the
    appropriate median nadir CD4?
#NIDU Started
mnadir<-median(unique(cbind(dat$finid, dat$nadir))[,2])

g1.oneper<-cbind(dat$finid, dat$group1, dat$uu, gs.2)
g1.oneper<-unique(subset(g1.oneper, g1.oneper[,2]==1))
uu.st.nidu<-mean(g1.oneper[,3])
g1.oneper<-cbind(rep(mnadir, length(g1.oneper[,1])), g1.oneper)

#IDU Started
g2.oneper<-cbind(dat$finid, dat$group2, dat$uu, gs.2)
g2.oneper<-unique(subset(g2.oneper, g2.oneper[,2]==1))
uu.st.idu<-mean(g2.oneper[,3])
g2.oneper<-cbind(rep(mnadir, length(g2.oneper[,1])), g2.oneper)

#NIDU lost
g3.oneper<-cbind(dat$finid, dat$group3, dat$uu, gs.1)
g3.oneper<-unique(subset(g3.oneper, g3.oneper[,2]==1))
uu.lost.nidu<-mean(g3.oneper[,3])
g3.oneper<-cbind(rep(mnadir, length(g3.oneper[,1])), g3.oneper)

#IDU Lost
g4.oneper<-cbind(dat$finid, dat$group4, dat$uu, gs.1)
g4.oneper<-unique(subset(g4.oneper, g4.oneper[,2]==1))
uu.lost.idu<-mean(g4.oneper[,3])
g4.oneper<-cbind(rep(mnadir, length(g4.oneper[,1])), g4.oneper)


coeff<-model$coefficients$fixed

fix<-coeff[names(coeff)%in%c('nadir:day', 'day:gs.1one', 'day:gs.11', '
    day:gs.12','day:gs.2one', 'day:gs.21', 'day:gs.22')]

m.st.nidu<-mean(g1.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])
m.st.idu<-mean(g2.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])+coeff[10]
diff.st<-m.st.idu-m.st.nidu

m.lost.nidu<-mean(g3.oneper[,-c(2, 3,4)]%*%fix[1:4])
m.lost.idu<-mean(g4.oneper[,-c(2, 3,4)]%*%fix[1:4])+coeff[11]
diff.lost<-m.lost.idu-m.lost.nidu

#get p vals

pvalues<-summary(model)$tTable[,5]}

m.idu<-(27/78)*m.st.idu + (51/78)*m.lost.idu
m.nidu<-(381/996)*m.st.nidu + (615/996)*m.lost.nidu

i.idu<-(27/78)*(coeff[1]+coeff[7]) + (51/78)*(coeff[1]+coeff[8])
```

```
i.nidu<-(381/996)*(coeff[1]+coeff[9]) + (615/996)*coeff[1]

#Make Plots
coeff.reason<-c(coeff[1:9], coeff[13], m.lost.nidu, m.lost.idu, m.st.
    nidu, m.st.idu)
coeff.marg<-c(coeff[2:6], coeff[13], i.nidu, i.idu, m.nidu, m.idu)

mage<-median(unique(cbind(dat$finid, dat$age))[,2])

days<-seq(0, 1100, 50)

#Recents
pred.lost.nidu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23),
    acute=rep(0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep
    (0, 23), g2=rep(0, 23), g4=rep(0, 23), g1=rep(0, 23), acute_day=rep
    (0, 23), lost.nidu=days, lost.idu=rep(0, 23), st.nidu=rep(0, 23), st
    .idu=rep(0, 23))

pred.lost.idu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute
    =rep(0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(0, 23), g4=rep(1, 23), g1=rep(0, 23), acute_day=rep(0,
    23), lost.nidu=rep(0, 23), lost.idu=days, st.nidu=rep(0, 23), st.idu
    =rep(0, 23))

pred.st.nidu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=
    rep(0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(0, 23), g4=rep(0, 23), g1=rep(1, 23), acute_day=rep(0,
    23), lost.nidu=rep(0, 23), lost.idu=rep(0,23), st.nidu=days, st.idu=
    rep(0, 23))

pred.st.idu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=
    rep(0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(1, 23), g4=rep(0, 23), g1=rep(0, 23), acute_day=rep(0,
    23), lost.nidu=rep(0, 23), lost.idu=rep(0,23), st.nidu=rep(0, 23),
    st.idu=days)

#Acutes
pred.lost.nidu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23),
    acute=rep(1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep
    (0, 23), g2=rep(0, 23), g4=rep(0, 23), g1=rep(0, 23), acute_day=days
    , lost.nidu=days, lost.idu=rep(0, 23), st.nidu=rep(0, 23), st.idu=
    rep(0, 23))

pred.lost.idu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute
    =rep(1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(0, 23), g4=rep(1, 23), g1=rep(0, 23), acute_day=days,
    lost.nidu=rep(0, 23), lost.idu=days, st.nidu=rep(0, 23), st.idu=rep
    (0, 23))

pred.st.nidu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=
    rep(1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(0, 23), g4=rep(0, 23), g1=rep(1, 23), acute_day=days,
    lost.nidu=rep(0, 23), lost.idu=rep(0, 23), st.nidu=days, st.idu=rep
    (0, 23))
```

```
pred.st.idu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=
    rep(1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0,
    23), g2=rep(1, 23), g4=rep(0, 23), g1=rep(0, 23), acute_day=days,
    lost.nidu=rep(0, 23), lost.idu=rep(0, 23), st.nidu=rep(0,23), st.idu
    =days)

pred.data<-rbind(pred.lost.nidu.r, pred.lost.idu.r, pred.st.nidu.r, pred
    .st.idu.r, pred.lost.nidu.a, pred.lost.idu.a, pred.st.nidu.a, pred.
    st.idu.a)

graph.data<-as.matrix(pred.data)%*%coeff.reason

pred.marg<-cbind(pred.data[,2:6], pred.data[,10], pred.data[,1]-pred.
    data[,7]-pred.data[,8], pred.data[,7]+pred.data[,8], pred.data[,11]+
    pred.data[,13], pred.data[,12]+pred.data[,14])

marg.data<-as.matrix(pred.marg)%*%coeff.marg

#Graph for Recents
par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))


plot(x=days/365, y=graph.data[1:23], xlab='Years', ylab='Log(CD4)', type
    ='l', lty=3, col='blue', ylim=range(graph.data))
lines(x=days/365, y=graph.data[24:46], lty=3, col='red')
lines(x=days/365, y=graph.data[47:69], lty=4, col='blue')
lines(x=days/365, y=graph.data[70:92], lty=4, col='red')
lines(x=days/365, y=marg.data[1:23], lty=1, lwd=2, col='blue')
lines(x=days/365, y=marg.data[24:46], lty=1, lwd=2, col='red')

par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.5,0.2, c('NIDU',"IDU", 'NIDU, Lost', 'IDU, Lost', 'NIDU, ST',
    'IDU, ST'), lty=c(1,1,3,3,4,4), col=c("blue", 'red', 'blue', 'red',
    'blue', 'red'),lwd=c(2,2,1,1,1,1), box.col=NA, ncol=3, cex=1.4)


#Exp Graph Recents

par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))


plot(x=days/365, y=exp(graph.data[1:23]), xlab='Years', ylab='CD4', type
    ='l', lty=3, col='blue', ylim=range(exp(graph.data)))
lines(x=days/365, y=exp(graph.data[24:46]), lty=3, col='red')
lines(x=days/365, y=exp(graph.data[47:69]), lty=4, col='blue')
lines(x=days/365, y=exp(graph.data[70:92]), lty=4, col='red')
lines(x=days/365, y=exp(marg.data[1:23]), lty=1, lwd=2, col='blue')
lines(x=days/365, y=exp(marg.data[24:46]), lty=1, lwd=2, col='red')

par(op)
```

```
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.5,0.2, c('NIDU',"IDU", 'NIDU, Lost', 'IDU, Lost', 'NIDU, ST',
    'IDU, ST'), lty=c(1,1,3,3,4,4), col=c("blue", 'red', 'blue', 'red',
    'blue', 'red'),lwd=c(2,2,1,1,1,1), box.col=NA, ncol=3, cex=1.4)


#Graph for Acutes
par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))


plot(x=days/365, y=graph.data[93:115], xlab='Years', ylab='Log(CD4)',
    type='l', lty=3, col='blue', ylim=range(graph.data))
lines(x=days/365, y=graph.data[116:138], lty=3, col='red')
lines(x=days/365, y=graph.data[139:161], lty=4, col='blue')
lines(x=days/365, y=graph.data[162:184], lty=4, col='red')
lines(x=days/365, y=marg.data[93:115], lty=1, lwd=2, col='blue')
lines(x=days/365, y=marg.data[116:138], lty=1, lwd=2, col='red')

par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.5,0.2, c('NIDU',"IDU", 'NIDU, Lost', 'IDU, Lost', 'NIDU, ST',
    'IDU, ST'), lty=c(1,1,3,3,4,4), col=c("blue", 'red', 'blue', 'red',
    'blue', 'red'),lwd=c(2,2,1,1,1,1), box.col=NA, ncol=3, cex=1.4)

#Exp Graph for Acutes
par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))


plot(x=days/365, y=exp(graph.data[93:115]), xlab='Years', ylab='CD4',
    type='l', lty=3, col='blue', ylim=range(exp(graph.data)))
lines(x=days/365, y=exp(graph.data[116:138]), lty=3, col='red')
lines(x=days/365, y=exp(graph.data[139:161]), lty=4, col='blue')
lines(x=days/365, y=exp(graph.data[162:184]), lty=4, col='red')
lines(x=days/365, y=exp(marg.data[93:115]), lty=1, lwd=2, col='blue')
lines(x=days/365, y=exp(marg.data[116:138]), lty=1, lwd=2, col='red')

par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.75,0.2, c('NIDU',"IDU", 'NIDU, Lost', 'IDU, Lost', 'NIDU, ST',
     'IDU, ST'), lty=c(1,1,3,3,4,4), col=c("blue", 'red', 'blue', 'red',
     'blue', 'red'),lwd=c(2,2,1,1,1,1), box.col=NA, ncol=3, cex=1.4)

kse.model<-model
kse.graph<-graph.data
kse.marg<-marg.data

kse.reason<-c(m.lost.nidu, m.lost.idu, m.st.nidu, m.st.idu, diff.lost,
    diff.st)
kse.oa<-c(m.nidu, m.idu)

###for KSQ
```

```
#read in dataset
file.dat<-paste(dir,"aiedrp_final.dat",sep="")
dat<-read.table(file.dat, header=T)
dat$group1<-ifelse(dat$idu==0 & dat$NEWtrt==1, 1, 0)
dat$group2<-ifelse(dat$idu==1 & dat$NEWtrt==1, 1, 0)
dat$group3<-ifelse(dat$idu==0 & dat$NEWtrt==0, 1, 0)
dat$group4<-ifelse(dat$idu==1 & dat$NEWtrt==0, 1, 0)
dat$finid<-dat$patid

  #Subset data by group
  set1<-subset(dat, dat$NEWtrt==0)
  set2<-subset(dat, dat$NEWtrt==1)

  #Determine Knots for Each Set
  df.list<-rep(NA,2)
  uu.knots.list<-rep(NA,2)
  boundary.list<-cbind(rep(NA,2), rep(NA,2))

  for (k in 1:2){set<-get(paste("set",k, sep=""))

  oneper <- cbind(set$finid, set$uu)
  oneper <- unique.data.frame(oneper)
  uuu<-oneper[,2]


  boundary<-range(uuu)
  lowerb<-168
  upperb<-max(uuu)

  boundary[1]<-lowerb

  ok<-(uuu<=boundary[1])+(uuu>boundary[2])
  uuu<-uuu[ok==0]

  #create dataframe to store knots and AIC values
  aic<-data.frame(knots = character(0), AIC = double(0), problems=
      double(0))

  #create set of candidate knots based on quantiles
  num_candidates<-9        #number of interior candidate knots
  lower<-1/(num_candidates+1) #quantile of lowest possible interior knot
  upper<-num_candidates*lower #quantile of highest possible interior
      knot
  candidates<-unique(quantile(uuu, probs = seq(lower, upper,lower ), na.
      rm = FALSE, names = TRUE, type = 7))
  candidates<-subset(candidates, candidates>lowerb & candidates<max(uuu)
      )

  # Search for gi, gs combination with smallest AIC

  GS <- 7 # gs (slope) number of parameters 1 (constant) to 7 (6 df)
  one <- rep(1,length(set$day))

  for(qq in 1:GS) {
```

```
  if (qq==1){ns.gs <- one
    error<-0
    warning <- 0
    tryCatch(
      nsv <-lme(logcd4 ~ nadir + acute + minority + age + female + idu
          + idu:day + nadir:day + acute:day + day:ns.gs, random=~day|
          finid,method='ML',data=set)
      ,warning = function(warn){warning <<-1}
      , error=function(err){error<<-1}
    )
    probs<-warning+error
    aic<-rbind(aic, data.frame(df=1, knots=NA, AIC=ifelse(probs!=0,NA,
        AIC(logLik(nsv))), problems=probs))

  }else if(qq>1) {subsets <- combn(candidates, (qq-2))
    for (j in 1:ncol(subsets)){col=subsets[,j]
    nsuu <-ns(set$uu, knots=col, Boundary.knots=boundary)   #b spline
        transformation
    ns.gs<-cbind(one, nsuu)              #include a 1 for main time
        effect

    error<-0
    warning <- 0
    tryCatch(
      nsv <-lme(logcd4 ~ nadir + acute + minority + age + female + idu
          + idu:day + nadir:day + acute:day + day:ns.gs, random=~day|
          finid,method='ML',data=set)
      ,warning = function(warn){warning <<-1}
      , error=function(err){error<<-1}
    )
    probs<-warning+error
    aic<-rbind(aic, data.frame(df=qq, knots=toString(subsets[,j]), AIC
        =ifelse(probs !=0,NA,AIC(logLik(nsv))), problems=probs))

  }
}
}

aic$knots<-ifelse(aic$df==2, NA, aic$knots)

which.min.aic <- which(aic$AIC==min(aic$AIC, na.rm=T))

# Case with only 1 min.aic
if (length(which.min.aic)==1){
  gs.knots<-aic$knots[which.min.aic]
  gs.df<-aic$df[which.min.aic]  } else {


# Case of min.aic ties
  min.aic<-aic[which.min.aic,]
  sort.aic<-min.aic[with(min.aic, order(df)),]
  min.gs <- sort.aic[1,] #Takes case w/ min param
  gs.knots<-min.gs$knots
  gs.df<-min.gs$df
```

```
  }

uu.knots.list[i]<-gs.knots
boundary.list[i,]<-boundary
df.list[i]<-gs.df
}

#Fit the final model with all the covariates
uu.knots1<-(uu.knots.list[1])
uu.knots2<-(uu.knots.list[2])

#uu.knots1<-260.8
#uu.knots1<-351.6

if (is.na(uu.knots1)==T | is.na(uu.knots2)==T){coeff=rep(NA, 19)
  pvalues=rep(NA,19)
  m.st.nidu=NA
  m.st.idu= NA
  m.lost.nidu= NA
  m.lost.idu= NA
  diff.st= NA
  diff.lost= NA
  knots.st=NA
  knots.lost=NA
  mnadir=NA
  uu.st.nidu=NA
  uu.st.idu=NA
  uu.lost.nidu=NA
  uu.lost.idu=NA

  }else{one<-rep(1, length(dat$uu))
gs.1<-ns(dat$uu, knots=uu.knots1, Boundary.knots=boundary.list[1,])  #
gs.1<-cbind(one, gs.1)
gs.1[,1]<-ifelse(dat$NEWtrt==0, gs.1[,1], 0)
gs.1[,2]<-ifelse(dat$NEWtrt==0, gs.1[,2], 0)
gs.1[,3]<-ifelse(dat$NEWtrt==0, gs.1[,3], 0)

gs.2<-ns(dat$uu, knots=uu.knots2, Boundary.knots=boundary.list[2,])
gs.2<-cbind(one, gs.2)
gs.2[,1]<-ifelse(dat$NEWtrt==1, gs.2[,1], 0)
gs.2[,2]<-ifelse(dat$NEWtrt==1, gs.2[,2], 0)
gs.2[,3]<-ifelse(dat$NEWtrt==1, gs.2[,3], 0)

warning<-0
error<-0

tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
    acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
      )
      probs<-warning+error
```

```
if (probs>0){warning<-0
error<-0
  tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female
      + group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
      acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat,
      control=lmeControl(opt = "optim"))
         ,warning = function(warn){warning <<-1}
         , error=function(err){error<<-1}
       )
       probs<-warning+error
}

if (probs==0){
#Estimate the marginal slopes for the 4 groups...need to put in the
    appropriate median nadir CD4?
#NIDU Started
mnadir<-median(unique(cbind(dat$finid, dat$nadir))[,2])

g1.oneper<-cbind(dat$finid, dat$group1, dat$uu, gs.2)
g1.oneper<-unique(subset(g1.oneper, g1.oneper[,2]==1))
uu.st.nidu<-mean(g1.oneper[,3])
g1.oneper<-cbind(rep(mnadir, length(g1.oneper[,1])), g1.oneper)

#IDU Started
g2.oneper<-cbind(dat$finid, dat$group2, dat$uu, gs.2)
g2.oneper<-unique(subset(g2.oneper, g2.oneper[,2]==1))
uu.st.idu<-mean(g2.oneper[,3])
g2.oneper<-cbind(rep(mnadir, length(g2.oneper[,1])), g2.oneper)

#NIDU lost
g3.oneper<-cbind(dat$finid, dat$group3, dat$uu, gs.1)
g3.oneper<-unique(subset(g3.oneper, g3.oneper[,2]==1))
uu.lost.nidu<-mean(g3.oneper[,3])
g3.oneper<-cbind(rep(mnadir, length(g3.oneper[,1])), g3.oneper)

#IDU Lost
g4.oneper<-cbind(dat$finid, dat$group4, dat$uu, gs.1)
g4.oneper<-unique(subset(g4.oneper, g4.oneper[,2]==1))
uu.lost.idu<-mean(g4.oneper[,3])
g4.oneper<-cbind(rep(mnadir, length(g4.oneper[,1])), g4.oneper)


coeff<-model$coefficients$fixed

fix<-coeff[names(coeff)%in%c('nadir:day', 'day:gs.1one', 'day:gs.11', '
    day:gs.12','day:gs.2one', 'day:gs.21', 'day:gs.22')]

m.st.nidu<-mean(g1.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])
m.st.idu<-mean(g2.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])+coeff[10]
diff.st<-m.st.idu-m.st.nidu

m.lost.nidu<-mean(g3.oneper[,-c(2, 3,4)]%*%fix[1:4])
m.lost.idu<-mean(g4.oneper[,-c(2, 3,4)]%*%fix[1:4])+coeff[11]
```

```
diff.lost<-m.lost.idu-m.lost.nidu

#get p vals

pvalues<-summary(model)$tTable[,5]}

m.idu<-(27/78)*m.st.idu + (51/78)*m.lost.idu
m.nidu<-(381/996)*m.st.nidu + (615/996)*m.lost.nidu

i.idu<-(27/78)*(coeff[1]+coeff[7]) + (51/78)*(coeff[1]+coeff[8])
i.nidu<-(381/996)*(coeff[1]+coeff[9]) + (615/996)*coeff[1]

#Fit NSV Model

for (i in 1:2){set<-get(paste("set",i, sep=""))

oneper <- cbind(set$patid, set$uu)
oneper <- unique.data.frame(oneper)
uuu<-oneper[,2]
uu<-set$uu
boundary<-range(uuu)      # NEW NEW NEW NEW NEW
lowerb<-168 # NEW NEW NEW NEW NEW bound_2=0.0666667, bound_3=0.13333333,
    bound_4=0.2
upperb<-max(uuu)
boundary[1]<-lowerb
ok<-(uuu<=boundary[1])+(uuu>boundary[2])
  uuu<-uuu[ok==0]
#create dataframe to store knots and AIC values
  ------------------------------------------------------------
  GI <- 1 # gi (intercept) number of parameters 1 (constant = no vc) to
     7 (6 df)
  GS <- 4 # gs (slope) number of parameters 1 (constant) to 7 (6 df)

  one <- rep(1,length(set$day))
  aic <- bic <- index <- problems <- NULL

  for(qq in 1:GS) {
    ns.gs <- one
    if(qq>1) {  #ns.gs <- cbind(1,stretchMat%*%ns(oneper$drptm,qq-1)) #
        can't use stretchMat
        knots.gs<-attributes(ns(uuu,(qq-1),Boundary.knots=boundary))$
            knots
        ns.gs<-cbind(one,ns(uu,Boundary.knots=boundary,knots=knots.gs))}

      ns.gi <- one

      warning <- 0
      tryCatch(
        #nsv <- lme(logcd4 ~ nadir + acute + female + minority + age +
            female:day  + acute:day + day:nadir + day: ns.gs + (day|
            finid),REML=F, data=set)
        nsv <- lme(logcd4 ~ nadir + acute + female + minority + age +
            idu + idu:day + acute:day + day:nadir + day: ns.gs, random=~
            day|finid,method='ML',data=set)
```

```
        #nsv <- lmer(logcd4 ~ nadir + day:nadir + day: ns.gs + (day|
            patid),REML=F, data=set)
         ,warning = function(warn){warning <<-1}
       )
       problems <- c(problems,warning)
       aic <- c(aic,ifelse(warning==1,NA,AIC(logLik(nsv))))
       #bic <- c(bic,ifelse(warning==1,NA,BIC(logLik(nsv))))
       index <- rbind(index,c(qq,ss))
       nsv<-NULL
     }


   which.min.aic <- which(aic==min(aic, na.rm=T))

   # Case with only 1 min.aic
   if (length(which.min.aic)==1){
     min.gi <- index[which.min.aic,2]
     min.gs <- index[which.min.aic,1]} else {

   # Case of min.aic ties
     params <- (index[,1]+index[,2])[which.min.aic] # Params in tied
         cases
     min.gi <- index[which.min.aic[order(params)][1],2] #Takes case w/
         min param
     min.gs <- index[which.min.aic[order(params)][1],1] #Takes case w/
         min param
   }
   df.list[i]<-min.gs }

   uu.knots1<-701
   uu.knots2<-453

   if (is.na(uu.knots1)==T | is.na(uu.knots2)==T){coeff=rep(NA, 19)
   pvalues=rep(NA,19)
   m.st.nidu=NA
   m.st.idu= NA
   m.lost.nidu= NA
   m.lost.idu= NA
   diff.st= NA
   diff.lost= NA
   knots.st=NA
   knots.lost=NA
   mnadir=NA
   uu.st.nidu=NA
   uu.st.idu=NA
   uu.lost.nidu=NA
   uu.lost.idu=NA

   }else{one<-rep(1, length(dat$uu))
gs.1<-ns(dat$uu, knots=uu.knots1, Boundary.knots=boundary.list[1,])  #
gs.1<-cbind(one, gs.1)
gs.1[,1]<-ifelse(dat$NEWtrt==0, gs.1[,1], 0)
gs.1[,2]<-ifelse(dat$NEWtrt==0, gs.1[,2], 0)
gs.1[,3]<-ifelse(dat$NEWtrt==0, gs.1[,3], 0)
```

```
gs.2<-ns(dat$uu, knots=uu.knots2, Boundary.knots=boundary.list[2,])
gs.2<-cbind(one, gs.2)
gs.2[,1]<-ifelse(dat$NEWtrt==1, gs.2[,1], 0)
gs.2[,2]<-ifelse(dat$NEWtrt==1, gs.2[,2], 0)
gs.2[,3]<-ifelse(dat$NEWtrt==1, gs.2[,3], 0)

warning<-0
error<-0

tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
    acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
       )
       probs<-warning+error

if (probs>0){warning<-0
error<-0
  tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female
     + group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
     acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat,
     control=lmeControl(opt = "optim"))
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
       )
       probs<-warning+error
}

if (probs==0){
#Estimate the marginal slopes for the 4 groups...need to put in the
    appropriate median nadir CD4?
#NIDU Started
mnadir<-median(unique(cbind(dat$finid, dat$nadir))[,2])

g1.oneper<-cbind(dat$finid, dat$group1, dat$uu, gs.2)
g1.oneper<-unique(subset(g1.oneper, g1.oneper[,2]==1))
uu.st.nidu<-mean(g1.oneper[,3])
g1.oneper<-cbind(rep(mnadir, length(g1.oneper[,1])), g1.oneper)

#IDU Started
g2.oneper<-cbind(dat$finid, dat$group2, dat$uu, gs.2)
g2.oneper<-unique(subset(g2.oneper, g2.oneper[,2]==1))
uu.st.idu<-mean(g2.oneper[,3])
g2.oneper<-cbind(rep(mnadir, length(g2.oneper[,1])), g2.oneper)

#NIDU lost
g3.oneper<-cbind(dat$finid, dat$group3, dat$uu, gs.1)
g3.oneper<-unique(subset(g3.oneper, g3.oneper[,2]==1))
uu.lost.nidu<-mean(g3.oneper[,3])
g3.oneper<-cbind(rep(mnadir, length(g3.oneper[,1])), g3.oneper)

#IDU Lost
```

```
g4.oneper<-cbind(dat$finid, dat$group4, dat$uu, gs.1)
g4.oneper<-unique(subset(g4.oneper, g4.oneper[,2]==1))
uu.lost.idu<-mean(g4.oneper[,3])
g4.oneper<-cbind(rep(mnadir, length(g4.oneper[,1])), g4.oneper)


coeff<-model$coefficients$fixed

fix<-coeff[names(coeff)%in%c('nadir:day', 'day:gs.1one', 'day:gs.11', '
    day:gs.12','day:gs.2one', 'day:gs.21', 'day:gs.22')]

m.st.nidu<-mean(g1.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])
m.st.idu<-mean(g2.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])+coeff[10]
diff.st<-m.st.idu-m.st.nidu

m.lost.nidu<-mean(g3.oneper[,-c(2, 3,4)]%*%fix[1:4])
m.lost.idu<-mean(g4.oneper[,-c(2, 3,4)]%*%fix[1:4])+coeff[11]
diff.lost<-m.lost.idu-m.lost.nidu

#get p vals

pvalues<-summary(model)$tTable[,5]}

m.idu<-(27/78)*m.st.idu + (51/78)*m.lost.idu
m.nidu<-(381/996)*m.st.nidu + (615/996)*m.lost.nidu

i.idu<-(27/78)*(coeff[1]+coeff[7]) + (51/78)*(coeff[1]+coeff[8])
i.nidu<-(381/996)*(coeff[1]+coeff[9]) + (615/996)*coeff[1]

#Random Effects Model
re.model<-lme(logcd4 ~ nadir + acute + minority + age + female +idu+ day
    + day:idu + nadir:day + acute:day, random=~day|finid,data=dat)

coeff<-re.model$coefficients$fixed

pred.nidu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=rep
    (0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0, 23),
    idu=rep(0, 23), day=days, idu_day=rep(0,23), nadir_day=mnadir*days,
    acute_day=rep(0, 23))

pred.idu.r<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=rep
    (0, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0, 23),
    idu=rep(1, 23), day=days, idu_day=days, nadir_day=mnadir*days, acute
    _day=rep(0, 23))


#Acutes
pred.nidu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=rep
    (1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0, 23),
    idu=rep(0, 23), day=days, idu_day=rep(0,23), nadir_day=mnadir*days,
    acute_day=days)

pred.idu.a<-data.frame(int=rep(1, 23), nadir=rep(mnadir, 23), acute=rep
    (1, 23), minority=rep(0, 23), age=rep(mage, 23), female=rep(0, 23),
```

```
    idu=rep(1, 23), day=days, idu_day=days, nadir_day=mnadir*days, acute
    _day=days)


pred.data<-rbind(pred.nidu.r, pred.idu.r, pred.nidu.a, pred.idu.a)

marg.data<-as.matrix(pred.data)%*%coeff

re.marg.data<-marg.data

m.nidu<-coeff[8]+coeff[10]*mnadir
m.idu<-coeff[8]+coeff[9]+coeff[10]*mnadir
re.oa<-c(m.nidu, m.idu, m.idu-m.nidu)


#CLM
clm.model<-lme(logcd4 ~ nadir + acute + minority + age + female + group2
     + group4 + group1+ day:group1 + day:group2 + day:group3 + day:
    group4 + nadir:day + acute:day + day:uu + NEWtrt:day:uu, random=~day
    |finid,data=dat)

uus<-dat$uu^2
uuc<-dat$uu^3

clm.model.s<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group1 + day:group2 + day:group3 + day
    :group4 + nadir:day + acute:day + day:uu + NEWtrt:day:uu+ day:uus +
    NEWtrt:day:uus, random=~day|finid,data=dat)

clm.model.c<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group1 + day:group2 + day:group3 + day
    :group4 + nadir:day + acute:day + day:uu + NEWtrt:day:uu+ day:uus +
    NEWtrt:day:uus+ day:uuc + NEWtrt:day:uuc, random=~day|finid,data=dat
    )

coeff<-clm.model$coefficients$fixed

#Estimate the marginal slopes for the 4 groups
#NIDU Started
mnadir<-median(unique(cbind(dat$finid, dat$nadir))[,2])

g1.oneper<-cbind(dat$finid, dat$group1, dat$uu)
g1.oneper<-unique(subset(g1.oneper, g1.oneper[,2]==1))
uu.st.nidu<-mean(g1.oneper[,3])
g1.oneper<-cbind(rep(mnadir, length(g1.oneper[,1])), g1.oneper)

#IDU Started
g2.oneper<-cbind(dat$finid, dat$group2, dat$uu)
g2.oneper<-unique(subset(g2.oneper, g2.oneper[,2]==1))
uu.st.idu<-mean(g2.oneper[,3])
g2.oneper<-cbind(rep(mnadir, length(g2.oneper[,1])), g2.oneper)

#NIDU lost
g3.oneper<-cbind(dat$finid, dat$group3, dat$uu)
```

```
g3.oneper<-unique(subset(g3.oneper, g3.oneper[,2]==1))
uu.lost.nidu<-mean(g3.oneper[,3])
g3.oneper<-cbind(rep(mnadir, length(g3.oneper[,1])), g3.oneper)

#IDU Lost
g4.oneper<-cbind(dat$finid, dat$group4, dat$uu)
g4.oneper<-unique(subset(g4.oneper, g4.oneper[,2]==1))
uu.lost.idu<-mean(g4.oneper[,3])
g4.oneper<-cbind(rep(mnadir, length(g4.oneper[,1])), g4.oneper)


coeff<-clm.model$coefficients$fixed

m.st.nidu<-coeff[10]+mnadir*coeff[14]+mean(g1.oneper[,4]*(coeff[16]+
    coeff[17]))
m.st.idu<-coeff[11]+mnadir*coeff[14]+mean(g2.oneper[,4]*(coeff[16]+coeff
    [17]))
diff.st<-m.st.idu-m.st.nidu

m.lost.nidu<-coeff[12]+mnadir*coeff[14]+mean(g3.oneper[,4]*(coeff[16]))
m.lost.idu<-coeff[13]+mnadir*coeff[14]+mean(g4.oneper[,4]*(coeff[16]))
diff.lost<-m.lost.idu-m.lost.nidu

m.idu<-(27/78)*m.st.idu + (51/78)*m.lost.idu
m.nidu<-(381/996)*m.st.nidu + (615/996)*m.lost.nidu

i.idu<-(27/78)*(coeff[1]+coeff[7]) + (51/78)*(coeff[1]+coeff[8])
i.nidu<-(381/996)*(coeff[1]+coeff[9]) + (615/996)*coeff[1]

coeff.marg<-c(coeff[2:6], coeff[14], i.nidu, i.idu, m.nidu, m.idu)
marg.data<-as.matrix(pred.marg)%*%coeff.marg
clm.marg.data<-marg.data
clm.reason<-c(m.lost.nidu, m.lost.idu, m.st.nidu, m.st.idu, diff.lost,
    diff.st)
clm.oa<-c(m.nidu, m.idu)

#Plot Marginals for Different Methods
#Recents
par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))
plot(x=days/365, y=exp(kse.marg[1:23]), xlab='Years', ylab='CD4', type='
    l', lty=1, col='blue', ylim=range(exp(c(clm.marg.data, nsv.marg, re.
    marg.data, kse.marg))))
lines(x=days/365, y=exp(kse.marg[24:46]), lty=4, col='blue')
lines(x=days/365, y=exp(clm.marg.data[1:23]), lty=1, col='red')
lines(x=days/365, y=exp(clm.marg.data[24:46]), lty=4, col='red')
lines(x=days/365, y=exp(re.marg.data[1:23]), lty=1, col='black')
lines(x=days/365, y=exp(re.marg.data[24:46]), lty=4, col='black')
par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.6,0.2, c('KSE: NIDU','KSE: IDU', "CLM: NIDU","CLM: IDU", 'RE:
    NIDU', 'RE: IDU'), lty=c(1,4,1,4,1,4), col=c("blue", 'blue', 'red',
    'red', 'black', 'black'),lwd=c(1,1,1,1,1,1), box.col=NA, ncol=3, cex
    =1.4)
```

```
par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))
plot(x=days/365, y=kse.marg[1:23], xlab='Years', ylab='Log(CD4)', type='
    l', lty=1, col='blue', ylim=range((c(clm.marg.data, nsv.marg, re.
    marg.data, kse.marg))))
lines(x=days/365, y=kse.marg[24:46], lty=4, col='blue')
lines(x=days/365, y=clm.marg.data[1:23], lty=1, col='red')
lines(x=days/365, y=clm.marg.data[24:46], lty=4, col='red')
lines(x=days/365, y=re.marg.data[1:23], lty=1, col='black')
lines(x=days/365, y=re.marg.data[24:46], lty=4, col='black')
par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.6,0.2, c('KSE: NIDU','KSE: IDU', "CLM: NIDU","CLM: IDU", 'RE:
    NIDU', 'RE: IDU'), lty=c(1,4,1,4,1,4), col=c("blue", 'blue', 'red',
    'red', 'black', 'black'),lwd=c(1,1,1,1,1,1), box.col=NA, ncol=3, cex
    =1.4)

par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))
plot(x=days/365, y=exp(kse.marg[1:23]), xlab='Years', ylab='CD4', type='
    l', lty=1, col='blue', ylim=range(exp(c(clm.marg.data, nsv.marg, re.
    marg.data, kse.marg))))
lines(x=days/365, y=exp(kse.marg[24:46]), lty=4, col='blue')
lines(x=days/365, y=exp(re.marg.data[1:23]), lty=1, col='black')
lines(x=days/365, y=exp(re.marg.data[24:46]), lty=4, col='black')
par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.3,0.2, c('KSE: NIDU','KSE: IDU', 'RE: NIDU', 'RE: IDU'), lty=c
    (1,4,1,4), col=c("blue", 'blue', 'black', 'black'),lwd=c(1,1,1,1),
    box.col=NA, ncol=2, cex=1.4)

par(tcl=0.5, mgp=c(1,0,0), mar=c(2,2,2,1), mfrow=c(3,3), family='serif')
op <- par(oma=c(3,0,0,0),mfrow=c(1,1))
plot(x=days/365, y=kse.marg[1:23], xlab='Years', ylab='Log(CD4)', type='
    l', lty=1, col='blue', ylim=range((c(clm.marg.data, nsv.marg, re.
    marg.data, kse.marg.graph))))
lines(x=days/365, y=kse.marg[24:46], lty=4, col='blue')
lines(x=days/365, y=re.marg.data[1:23], lty=1, col='black')
lines(x=days/365, y=re.marg.data[24:46], lty=4, col='black')
par(op)
op <- par(usr=c(0,1,0,1),xpd=NA)
legend(-1.3,0.2, c('KSE: NIDU','KSE: IDU', 'RE: NIDU', 'RE: IDU'), lty=c
    (1,4,1,4), col=c("blue", 'blue', 'black', 'black'),lwd=c(1,1,1,1),
    box.col=NA, ncol=2, cex=1.4)


###Tables#########################################
log.reason.comp<-365*cbind(kse.reason, ksq.reason, nsv.reason, clm.
    reason)
colnames(log.reason.comp)<-c('KSE', 'KSQ', 'NSV', 'CLM')
rownames(log.reason.comp)<-c('Lost, NIDU', 'Lost, IDU', 'ST, NIDU', 'ST,
     IDU', 'Lost: IDU-NIDU', 'ST: IDU-NIDU')
latex(round(log.reason.comp, 3), file='')
```

```
exp.reason.comp<-(exp(log.reason.comp)-1)*100
colnames(log.reason.comp)<-c('KSE', 'KSQ', 'NSV', 'CLM')
rownames(log.reason.comp)<-c('Lost, NIDU', 'Lost, IDU', 'ST, NIDU', 'ST,
    IDU', 'Lost: IDU-NIDU', 'ST: IDU-NIDU')
latex(round(exp.reason.comp, 2), file='')

log.oa<-cbind(kse.oa, ksq.oa, nsv.oa, clm.oa, re.oa[1:2])
log.oa<-rbind(log.oa, log.oa[2,]-log.oa[1,])
colnames(log.oa)<-c('KSE', 'KSQ', 'NSV', 'CLM', 'RE')
rownames(log.oa)<-c('NIDU', 'IDU', 'IDU-NIDU')
log.oa<-365*log.oa
latex(round(log.oa, 3), file='')

exp.oa<-(exp(log.oa)-1)*100
latex(round(exp.oa, 2), file='')

##############random effects confidence intervals##############
k.nidu<-c(rep(0, 7), 1, 0, mnadir, 0)
k.idu<-c(rep(0, 7), 1, 1, mnadir, 0)
k.diff<-c(rep(0, 7), 0, 1, 0, 0)

nidu<-estimable(re.model, k.nidu, conf.int=0.95)
idu<-estimable(re.model, k.idu, conf.int=0.95)
d<-estimable(re.model, k.diff, conf.int=0.95)

re.table<-rbind(nidu, idu, d)
re.table<-cbind(365*re.table[,1:2], 365*re.table[,6:7], re.table[,c(3,5)
    ])
rownames(re.table)<-c('NIDU', 'IDU', 'IDU-NIDU')
colnames(re.table)<-c('Estimate', 'SE', "LCI", 'UCI', 'T Value', 'P
    Value')
latex(round(re.table, 4), file='')

re.exp.table<-(exp(re.table[,c(1, 3, 4)])-1)*100
latex(round(re.exp.table,2), file='')
```

BOOTSTRAP

```
# Originally written by Jeri Forster (appboot.r)
# To program the bootstrap for the slope difference
# Modified by Camille for MS Thesis
# Started: 01/28/2013

# GENERAL SETTINGS: cleanup, libraries,
  remove(list=ls())
  library(lme4)
  library(sampling)
  library(snow)
  library(nlme)
  dir <- "/Users/camille/Dropbox/Masters Paper/aiedrp/"
  dir2 <- "/Users/camille/Desktop/bootdata/"
  # make changes when necessary
```

```
  dir3 <- "C:/Users/moorecam/"
  dir4 <- "C:/Users/moorecam/"
  set.seed(1234)

#read in dataset
file.dat<-paste(dir,"aiedrp_final.dat",sep="")
dat<-read.table(file.dat, header=T)
dat$group1<-ifelse(dat$idu==0 & dat$NEWtrt==1, 1, 0)
dat$group2<-ifelse(dat$idu==1 & dat$NEWtrt==1, 1, 0)
dat$group3<-ifelse(dat$idu==0 & dat$NEWtrt==0, 1, 0)
dat$group4<-ifelse(dat$idu==1 & dat$NEWtrt==0, 1, 0)

# number of bootstraps

  nBoot <- 1000   # Number of datasets, must be consecutively numbered
  n_lost_idu<- length(unique(subset(dat, dat$group4==1)$patid))#Number
      of patients lost to follow up that are IDU
  n_lost_nidu<- length(unique(subset(dat, dat$group3==1)$patid))#Number
      of patients lost to follow up that are NIDU
  n_started_idu<- length(unique(subset(dat, dat$group2==1)$patid))#
      Number of patients that started treatment and are IDU
  n_started_nidu<- length(unique(subset(dat, dat$group1==1)$patid))#
      Number of patients that started treatment and are NIDU

CustomBoot<- function(i){


  # Read in data -- IDU
  lostidu <- subset(dat, dat$group4==1)        # 853 24 (3 years, 64
      patients)
  lostnidu<-subset(dat, dat$group3==1)
  startedidu <- subset(dat, dat$group2==1)
  startednidu<- subset(dat, dat$group1==1)


  file.out <- paste(dir2,"sampling2_",i,".dat",sep="")

  #file.out <- paste(dir2,"sampling2_4.dat",sep="")

  # dataset

  nidu.cl <- cluster(data=lostnidu,clustername=c("patid"),size=n_lost_
      nidu,method="srswr")
  idu.cl <- cluster(data=lostidu,clustername=c("patid"),size=n_lost_idu,
      method="srswr")
  stidu.cl <- cluster(data=startedidu,clustername=c("patid"),size=n_
      started_idu,method="srswr")
  stnidu.cl <- cluster(data=startednidu,clustername=c("patid"),size=n_
      started_nidu,method="srswr")

  # cluster returns: 1) the selected cluster (patid)
  #         2) the identifier of the units in the selected clusters (ID_
      unit)
  #         3) the final inclusion probabilities for these units (Prob).
```

```
      They are equal for the units included in the same cluster
#          4) if method is 'srswr', the number of replicates is also
    given (Replicates)

nidu.dat <- getdata(lostnidu,nidu.cl)
idu.dat <- getdata(lostidu,idu.cl)
stidu.dat <- getdata(startedidu,stidu.cl)
stnidu.dat <- getdata(startednidu,stnidu.cl)

# getdata: extracts the observed data from a data frame

#get nidu data
  # getdata: extracts the observed data from a data frame

nidu.dat <- nidu.dat[order(nidu.dat$patid,nidu.dat$day),]

nidu.dat.e <- nidu.dat[rep(row.names(nidu.dat), nidu.dat$Replicates),
    1:24]          # 749 27
                                                  # 489 rows in idu.dat
temp<-NULL
for (z in 1:nrow(nidu.dat)){temp<-c(temp, seq(1, nidu.dat$Replicates[z])
  )}
nidu.dat.e$fin <- temp/1000                                          #
nidu.dat.e$finid <- 1000*(nidu.dat.e$patid + nidu.dat.e$fin)
            # 64 unique finid (41 unique patid)
nidu.dat.e <- nidu.dat.e[order(nidu.dat.e$finid,nidu.dat.e$day),]

#get idu data
idu.dat <- idu.dat[order(idu.dat$patid,idu.dat$day),]

idu.dat.e <- idu.dat[rep(row.names(idu.dat), idu.dat$Replicates),
    1:24]          # 749 27


temp<-NULL
for (z in 1:nrow(idu.dat)){temp<-c(temp, seq(1, idu.dat$Replicates[z])
    )}
idu.dat.e$fin <- temp/1000                                           #
idu.dat.e$finid <- 1000*(idu.dat.e$patid + idu.dat.e$fin)
        # 64 unique finid (41 unique patid)
idu.dat.e <- idu.dat.e[order(idu.dat.e$finid,idu.dat.e$day),]
                                                  # 489 rows in idu.dat

#get started treatment idu data
  # getdata: extracts the observed data from a data frame

stidu.dat <- stidu.dat[order(stidu.dat$patid,stidu.dat$day),]

stidu.dat.e <- stidu.dat[rep(row.names(stidu.dat), stidu.dat$
    Replicates), 1:24]          # 749 27

temp<-NULL
for (z in 1:nrow(stidu.dat)){temp<-c(temp, seq(1, stidu.dat$Replicates
    [z]))}
```

```
  stidu.dat.e$fin <- temp/1000                                    #
  stidu.dat.e$finid <- 1000*(stidu.dat.e$patid + stidu.dat.e$fin)
              # 64 unique finid (41 unique patid)
  stidu.dat.e <- stidu.dat.e[order(stidu.dat.e$finid,stidu.dat.e$day),]

  #get started treatment nidu data
    # getdata: extracts the observed data from a data frame

  stnidu.dat <- stnidu.dat[order(stnidu.dat$patid,stnidu.dat$day),]

  stnidu.dat.e <- stnidu.dat[rep(row.names(stnidu.dat), stnidu.dat$
      Replicates), 1:24]          # 749 27
                                              # 489 rows in idu.dat

  temp<-NULL
  for (z in 1:nrow(stnidu.dat)){temp<-c(temp, seq(1, stnidu.dat$
      Replicates[z]))}
  stnidu.dat.e$fin <- temp/1000                                   #
  stnidu.dat.e$finid <- 1000*(stnidu.dat.e$patid + stnidu.dat.e$fin)
                    # 64 unique finid (41 unique patid)
  stnidu.dat.e <- stnidu.dat.e[order(stnidu.dat.e$finid,stnidu.dat.e$day
      ),]

  write.table(rbind(nidu.dat.e, idu.dat.e, stidu.dat.e, stnidu.dat.e),
      file.out,sep='\t')

  }

  # Simulate 1000 datasets
  set.seed(213)
  nBoot <- 1000

  # Data simulation
  for (i in 1:nBoot) {
    CustomBoot(i)
  }


#Analyze the Bootstrap Datasets

rslt<-NULL

for (i in 1:1000){
  #Get Data
  dat<-read.table(paste(dir2,'sampling2_',i,'.dat', sep=''), header=T)
  #dat<-read.table(paste(dir2,'sampling2_2.dat', sep=''), header=T)

  #Subset data by group
  set1<-subset(dat, dat$NEWtrt==0)
  set2<-subset(dat, dat$NEWtrt==1)

  #Determine Knots for Each Set, 1 knot for Lost NIDU, 2 knots of Lost
      IDU, 1 knot for ST
  df.list<-rep(3,2)
```

```
uu.knots.list<-rep(NA,2)
boundary.list<-cbind(rep(NA,2), rep(NA,2))


for (k in 1:2){set<-get(paste("set",k, sep=""))

oneper <- cbind(set$finid, set$uu)
oneper <- unique.data.frame(oneper)
uuu<-oneper[,2]


boundary<-range(uuu)      # NEW NEW NEW NEW NEW
lowerb<-168 # NEW NEW NEW NEW NEW bound_2=0.0666667, bound_
    3=0.13333333, bound_4=0.2
upperb<-max(uuu)

boundary[1]<-lowerb

ok<-(uuu<=boundary[1])+(uuu>boundary[2])
uuu<-uuu[ok==0]

#create dataframe to store knots and AIC values aic<-data.frame(knots
    = character(0), AIC = double(0), problems= double(0))

#create set of candidate knots based on even spacing
num_candidates<-9        #number of interior candidate knots
step<-(upperb-lowerb)/(num_candidates+1)
lowerc<-lowerb+step
upperc<-upperb-step
candidates<-seq(lowerc, upperc, step)

# Search for gi, gs combination with smallest AIC

GS <- df.list[k]  # gs (slope) number of parameters 1 (constant) to 7
    (6 df)

one <- rep(1,length(set$day))

subsets <- combn(candidates, (GS-2))
     for (j in 1:ncol(subsets)){col=subsets[,j]
     nsuu <-ns(set$uu, knots=col, Boundary.knots=boundary)   #b spline
         transformation
     ns.gs<-cbind(one, nsuu)
     error<-0
     warning <- 0
     tryCatch(
       nsv<-lme(logcd4 ~ nadir + acute + minority + age + female + idu
           + idu:day + nadir:day + acute:day + day:ns.gs, random=~day|
           finid,method='ML',data=set)
       #nsv <-lmer(logcd4 ~ nadir + acute + minority + age + female +
           idu + idu:day + nadir:day + acute:day + day:ns.gs + (day|
           finid),REML=F, data=set)
       ,warning = function(warn){warning <<-1}
       , error=function(err){error<<-1}
```

```
        )
        probs<-warning+error
        aic<-rbind(aic, data.frame(knots=subsets[,j], AIC=ifelse(probs !=
            0,NA,AIC(logLik(nsv))), problems=probs))

      }

    if (length(subset(aic, is.na(aic$AIC)==T)$AIC)<9){
    which.min.aic <- which(aic$AIC==min(aic$AIC, na.rm=T))


    # Case with only 1 min.aic
    if (length(which.min.aic)==1){
      gs.knots<-aic$knots[which.min.aic]
      gs.df<-aic$df[which.min.aic]   } else {


    # Case of min.aic ties
      min.aic<-aic[which.min.aic,]
      sort.aic<-min.aic[with(min.aic, order(df)),]
      min.gs <- sort.aic[1,] #Takes case w/ min param
      gs.knots<-min.gs$knots
      gs.df<-min.gs$df   }

    uu.knots.list[k]<-gs.knots
    boundary.list[k,]<-boundary
    }}

#Fit the final model with all the covariates
uu.knots1<-(uu.knots.list[1])
uu.knots2<-(uu.knots.list[2])

if (is.na(uu.knots1)==T | is.na(uu.knots2)==T){coeff=rep(NA, 19)
   pvalues=rep(NA,19)
   m.st.nidu=NA
   m.st.idu= NA
   m.lost.nidu= NA
   m.lost.idu= NA
   diff.st= NA
   diff.lost= NA
   knots.st=NA
   knots.lost=NA
   mnadir=NA
   uu.st.nidu=NA
   uu.st.idu=NA
   uu.lost.nidu=NA
   uu.lost.idu=NA

   }else{one<-rep(1, length(dat$uu))
gs.1<-ns(dat$uu, knots=uu.knots1, Boundary.knots=boundary.list[1,])  #
gs.1<-cbind(one, gs.1)
gs.1[,1]<-ifelse(dat$NEWtrt==0, gs.1[,1], 0)
gs.1[,2]<-ifelse(dat$NEWtrt==0, gs.1[,2], 0)
gs.1[,3]<-ifelse(dat$NEWtrt==0, gs.1[,3], 0)
```

```
gs.2<-ns(dat$uu, knots=uu.knots2, Boundary.knots=boundary.list[2,])
gs.2<-cbind(one, gs.2)
gs.2[,1]<-ifelse(dat$NEWtrt==1, gs.2[,1], 0)
gs.2[,2]<-ifelse(dat$NEWtrt==1, gs.2[,2], 0)
gs.2[,3]<-ifelse(dat$NEWtrt==1, gs.2[,3], 0)

warning<-0
error<-0

tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female +
    group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
    acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat)
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
       )
       probs<-warning+error

if (probs>0){warning<-0
error<-0
  tryCatch( model<-lme(logcd4 ~ nadir + acute + minority + age + female
     + group2 + group4 + group1+ day:group2 + day:group4 + nadir:day +
      acute:day + day:gs.1 + day:gs.2, random=~day|finid,data=dat,
      control=lmeControl(opt = "optim"))
        ,warning = function(warn){warning <<-1}
        , error=function(err){error<<-1}
       )
       probs<-warning+error
}

if (probs==0){
#Estimate the marginal slopes for the 4 groups...need to put in the
    appropriate median nadir CD4?
#NIDU Started
mnadir<-median(unique(cbind(dat$finid, dat$nadir))[,2])

g1.oneper<-cbind(dat$finid, dat$group1, dat$uu, gs.2)
g1.oneper<-unique(subset(g1.oneper, g1.oneper[,2]==1))
uu.st.nidu<-mean(g1.oneper[,3])
g1.oneper<-cbind(rep(mnadir, length(g1.oneper[,1])), g1.oneper)

#IDU Started
g2.oneper<-cbind(dat$finid, dat$group2, dat$uu, gs.2)
g2.oneper<-unique(subset(g2.oneper, g2.oneper[,2]==1))
uu.st.idu<-mean(g2.oneper[,3])
g2.oneper<-cbind(rep(mnadir, length(g2.oneper[,1])), g2.oneper)

#NIDU lost
g3.oneper<-cbind(dat$finid, dat$group3, dat$uu, gs.1)
g3.oneper<-unique(subset(g3.oneper, g3.oneper[,2]==1))
uu.lost.nidu<-mean(g3.oneper[,3])
g3.oneper<-cbind(rep(mnadir, length(g3.oneper[,1])), g3.oneper)

#IDU Lost
```

```
g4.oneper<-cbind(dat$finid, dat$group4, dat$uu, gs.1)
g4.oneper<-unique(subset(g4.oneper, g4.oneper[,2]==1))
uu.lost.idu<-mean(g4.oneper[,3])
g4.oneper<-cbind(rep(mnadir, length(g4.oneper[,1])), g4.oneper)


coeff<-model$coefficients$fixed

fix<-coeff[names(coeff)%in%c('nadir:day', 'day:gs.1one', 'day:gs.11', '
    day:gs.12','day:gs.2one', 'day:gs.21', 'day:gs.22')]

m.st.nidu<-mean(g1.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])
m.st.idu<-mean(g2.oneper[,-c(2, 3,4)]%*%fix[c(1,5,6,7)])+coeff[10]
diff.st<-m.st.idu-m.st.nidu

m.lost.nidu<-mean(g3.oneper[,-c(2, 3,4)]%*%fix[1:4])
m.lost.idu<-mean(g4.oneper[,-c(2, 3,4)]%*%fix[1:4])+coeff[11]
diff.lost<-m.lost.idu-m.lost.nidu

#get p vals

pvalues<-summary(model)$tTable[,5]}

if (probs>0){coeff=rep(NA, 19)
  pvalues=rep(NA,19)
  m.st.nidu=NA
  m.st.idu= NA
  m.lost.nidu= NA
  m.lost.idu= NA
  diff.st= NA
  diff.lost= NA
  knots.st=NA
  knots.lost=NA
  mnadir=NA
  uu.st.nidu=NA
  uu.st.idu=NA
  uu.lost.nidu=NA
  uu.lost.idu=NA}}

#Save the marginals

rslt<-rbind(rslt, c(m.st.nidu, m.st.idu, m.lost.nidu, m.lost.idu, diff.
    st, diff.lost, uu.knots2, uu.knots1, mnadir, uu.st.nidu, uu.st.idu,
    uu.lost.nidu, uu.lost.idu, coeff, pvalues))


#bootstrap confidence intervals for KSE model
write.table(rslt,paste(dir2,"boot_results.txt",sep=""))

boot.nidu.lost<-c(sd(rslt[,3]), quantile(rslt[,3], probs=c(0.025, 0.975)
    ))
boot.idu.lost<-c(sd(rslt[,4]), quantile(rslt[,4], probs=c(0.025, 0.975))
    )
boot.nidu.st<-c(sd(rslt[,1]), quantile(rslt[,1], probs=c(0.025, 0.975)))
```

```
boot.idu.st<-c(sd(rslt[,2]), quantile(rslt[,2], probs=c(0.025, 0.975)))
boot.diff.lost<-c(sd(rslt[,6]), quantile(rslt[,6], probs=c(0.025, 0.975)
    ))
boot.diff.st<-c(sd(rslt[,5]), quantile(rslt[,5], probs=c(0.025, 0.975)))
boot.diff.idu<-c(sd(rslt[,2]-rslt[,4]), quantile(rslt[,2]-rslt[,4],
    probs=c(0.025, 0.975)))
boot.diff.nidu<-c(sd(rslt[,1]-rslt[,3]), quantile(rslt[,1]-rslt[,3],
    probs=c(0.025, 0.975)))
idu<-(27/78)*rslt[,2] + (51/78)*rslt[,4]
nidu<-(381/996)*rslt[,1] + (615/996)*rslt[,3]
diff<-idu-nidu
boot.m.nidu<-c(sd(nidu), quantile(nidu, probs=c(0.025, 0.975)))
boot.m.idu<-c(sd(idu), quantile(idu, probs=c(0.025, 0.975)))
boot.m.diff<-c(sd(diff), quantile(diff, probs=c(0.025, 0.975)))

boot.rslts<-rbind(boot.nidu.lost, boot.idu.lost, boot.nidu.st, boot.idu.
    st, boot.diff.lost, boot.diff.st, boot.diff.nidu, boot.diff.idu)

boot.m.results<-rbind(boot.m.nidu, boot.m.idu, boot.m.diff)

kse.reason<-c(kse.reason, kse.reason[3]-kse.reason[1], kse.reason[4]-kse
    .reason[1])

kse.oa<-c(kse.oa, kse.oa[2]-kse.oa[1])

final.rslts<-cbind(kse.reason, boot.rslts)

final.m.rslts<-cbind(kse.oa, boot.m.results)

rownames(final.rslts)<-c('Lost, NIDU', 'Lost, IDU', 'ST, NIDU', 'ST, IDU
    ', 'Lost IDU - Lost NIDU', 'ST IDU - ST NIDU', 'ST NIDU - Lost NIDU'
    , 'ST IDU - Lost IDU')
colnames(final.rslts)<-c('Marginal Slope', 'Standard Error', 'Lower 95%
    CI', 'Upper 95% CI')
final.rslts<-as.data.frame(final.rslts)
final.rslts$tvalue<-final.rslts[,1]/final.rslts[,2]
final.rslts$p<-(pnorm(final.rslts$tvalue)*2)
final.rslts.day<-final.rslts
final.rslts.year<-cbind(final.rslts[,1:4]*365, final.rslts$tvalue, final
    .rslts$p)
final.rslts.pct<-(exp(final.rslts[,c(1,3,4)]*365)-1)*100

latex(round(final.rslts.year, 3), file='')
latex(round(final.rslts.pct, 2), file='')


rownames(final.m.rslts)<-c('NIDU', 'IDU', 'IDU-NIDU')
colnames(final.m.rslts)<-c('Marginal Slope', 'Standard Error', 'Lower
    95% CI', 'Upper 95% CI')
final.m.rslts<-as.data.frame(final.m.rslts)
final.m.rslts$tvalue<-final.m.rslts[,1]/final.m.rslts[,2]
final.m.rslts$p<-(pnorm(final.m.rslts$tvalue)*2)
final.m.rslts.year<-cbind(final.m.rslts[,1:4]*365, final.m.rslts$tvalue,
     final.m.rslts$p)
```

```
final.m.rslts.pct<-(exp(final.m.rslts[,c(1,3,4)]*365)-1)*100

latex(round(final.m.rslts.year, 3), file='')
latex(round(final.m.rslts.pct, 2), file='')
```

APPENDIX C

AIEDRP MODELS

Table C.1: KSE Model Fit

Variable	Value	Std. Error	DF	T Value	P Value
(Intercept)	1.536330	0.126446	6277	12.15	< 0.0001
Nadir CD4	0.779886	0.019066	1065	40.91	< 0.0001
Acute	0.065047	0.023497	1065	2.77	0.0057
Minority	−0.032833	0.017984	1065	−1.83	0.0682
Age	−0.000779	0.000819	1065	−0.95	0.3416
Female	0.067455	0.030795	1065	2.19	0.0287
IDU, Started Tratment	−0.043924	0.051347	1065	−0.86	0.3925
IDU, Lost	0.003352	0.035735	1065	0.09	0.9253
NIDU, Started Treatment	−0.010536	0.018324	1065	−0.57	0.5654
IDU, Started Tratment*day	−0.000386	0.000190	6277	−2.03	0.0419
IDU, Lost*day	−0.000192	0.000090	6277	−2.12	0.0337
Nadir CD4*day	−0.000149	0.000053	6277	−2.84	0.0046
Acute*day	−0.000208	0.000065	6277	−3.21	0.0013
Lost spline term 1*day	0.000159	0.000356	6277	0.45	0.6563
Lost spline term 2*day	0.000830	0.000241	6277	3.44	0.0006
Lost spline term 3*day	0.000374	0.000078	6277	4.77	< 0.0001
Started treatment spline term 1*day	−0.000744	0.000365	6277	−2.04	0.0413
Started treatment spline term 2*day	0.001613	0.000312	6277	5.17	< 0.0001
Started treatment spline term 3*day	0.000655	0.000135	6277	4.87	< 0.0001

Table C.2: KSQ Model Fit

Variable	Value	Std.Error	DF	T Value	P Value
(Intercept)	1.536116	0.126442	6277	12.148817	< 0.0001
Nadir CD4	0.779923	0.019065	1065	40.909034	< 0.0001
Acute	0.065065	0.023497	1065	2.769100	0.005719
Minority	−0.032833	0.017984	1065	−1.825704	0.068175
Age	−0.000779	0.000819	1065	−0.951858	0.341385
Female	0.067457	0.030795	1065	2.190482	0.028705
IDU, Started Treatment	−0.044023	0.051343	1065	−0.857436	0.391397
IDU, Lost	0.003361	0.035736	1065	0.094045	0.925091
NIDU, Started Treatment	−0.010622	0.018317	1065	−0.579859	0.562132
IDU, Started Treatment*day	−0.000386	0.000190	6277	−2.035828	0.041810
IDU, Lost*day	−0.000192	0.000090	6277	−2.123502	0.033751
Nadir CD4*day	−0.000149	0.000053	6277	−2.835216	0.004594
Acute*day	−0.000208	0.000065	6277	−3.213318	0.001319
Lost spline term 1*day	0.000158	0.000357	6277	0.443128	0.657688
Lost spline term 2*day	0.000830	0.000242	6277	3.434435	0.000598
Lost spline term 3*day	0.000375	0.000079	6277	4.771388	< 0.0001
Started Treatment spline term 1*day	−0.000742	0.000364	6277	−2.036810	0.041711
Started Treatment spline term 2*day	0.001614	0.000310	6277	5.199219	< 0.0001
Started Treatment spline term 3*day	0.000647	0.000134	6277	4.818244	< 0.0001

Table C.3: NSV Model Fit

Variable	Value	Std. Error	DF	T Value	P Value
(Intercept)	1.530941	0.126359	6277	12.12	< 0.0001
Nadir CD4	0.780416	0.019057	1065	40.95	< 0.0001
Acute	0.065091	0.023495	1065	2.77	0.0057
Minority	−0.032852	0.017986	1065	−1.83	0.0681
Age	−0.000757	0.000818	1065	−0.93	0.3551
Female	0.067311	0.030799	1065	2.19	0.0291
IDU, Started Treatment	−0.043440	0.051297	1065	−0.85	0.3973
IDU, Lost	0.002887	0.035731	1065	0.08	0.9356
NIDU, Started Treatment	−0.009916	0.018221	1065	−0.54	0.5864
IDU, Started Treatment*day	−0.000388	0.000190	6277	−2.05	0.0409
IDU, Lost*day	−0.000193	0.000090	6277	−2.13	0.0330
Nadir CD4*day	−0.000149	0.000053	6277	−2.82	0.0047
Acute*day	−0.000208	0.000065	6277	−3.21	0.0013
Lost spline term 1*day	0.000220	0.000351	6277	0.63	0.5302
Lost spline term 2*day	0.000804	0.000217	6277	3.70	0.0002
Lost spline term 3*day	0.000237	0.000056	6277	4.24	< 0.0001
Started Treatment spline term 1*day	−0.000725	0.000363	6277	−2.00	0.0455
Started Treatment spline term 2*day	0.001613	0.000301	6277	5.36	< 0.0001
Started Treatment spline term 3*day	0.000593	0.000134	6277	4.44	< 0.0001

Table C.4: RE Model Fit

Variable	Value	Std. Error	DF	T Value	P Value
(Intercept)	1.388842	0.122816	6283	11.31	< 0.0001
Nadir CD4	0.799022	0.018624	1067	42.90	< 0.0001
Acute	0.068645	0.023633	1067	2.90	0.0038
Minority	−0.034492	0.018420	1067	−1.87	0.0614
Age	−0.000971	0.000840	1067	−1.16	0.2479
Female	0.071304	0.031547	1067	2.26	0.0240
IDU	−0.012507	0.029389	1067	−0.43	0.6705
Day	−0.000236	0.000374	6283	−0.63	0.5283
IDU*day	−0.000192	0.000093	6283	−2.07	0.0384
Nadir CD4*day	−0.000032	0.000059	6283	−0.54	0.5903
Acute*day	−0.000272	0.000073	6283	−3.71	0.0002

Table C.5: CLM Model Fit

Variable	Value	Std. Error	DF	T Value	P Value
(Intercept)	1.4971	0.1262	6279	11.8665	< 0.0001
Nadir CD4	0.7847	0.0190	1065	41.2159	< 0.0001
Acute	0.0656	0.0235	1065	2.7917	0.0053
Minority	−0.0342	0.0180	1065	−1.8978	0.0580
Age	−0.0007	0.0008	1065	−0.8270	0.4084
Female	0.0665	0.0309	1065	2.1551	0.0314
IDU, Started Treatment	−0.0588	0.0509	1065	−1.1560	0.2480
IDU, Lost	0.0014	0.0358	1065	0.0380	0.9697
NIDU, Started Treatment	−0.0181	0.0175	1065	−1.0335	0.3016
IDU, Started Treatment*day	−0.0008	0.0004	6279	−2.0872	0.0369
IDU, Lost*day	0.0002	0.0003	6279	0.7176	0.4730
NIDU, Started Treatment*day	−0.0005	0.0003	6279	−1.3228	0.1860
NIDU, Lost*day	0.0004	0.0003	6279	1.2563	0.2090
Droptime*day	0.0000	0.0000	6279	4.1542	< 0.0001
Nadir CD4*day	−0.0002	0.0001	6279	−3.0555	0.0023
Acute*day	−0.0002	0.0001	6279	−3.2100	0.0013
Started Treatment*Droptime*day	0.0000	0.0000	6279	3.1109	0.0019