SEMIPARAMETRIC ESTIMATION PROCEDURES USING LOCAL POLYNOMIAL SMOOTHING FOR INCONSISTENTLY MEASURED LONGITUDINAL DATA

by

Lei Ye

BMed, China Medical University, China, 2007

Submitted to the Graduate Faculty of

the Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Lei Ye

It was defended on

December 4, 2014

and approved by

Susan M. Sereika, PhD, Professor, Director, Center for Research and Evaluation, Health and Community Systems, School of Nursing, University of Pittsburgh

Lora Burke, PhD, MPH, RN, FAAN, Professor, Health and Community Systems, School of Nursing, University of Pittsburgh

Stewart J. Anderson, PhD, Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Dissertation Advisor: Ada O. Youk, PhD, Assistant Professor , Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Ada O. Youk, PhD

SEMIPARAMETRIC ESTIMATION PROCEDURES USING LOCAL POLYNOMIAL SMOOTHING FOR INCONSISTENTLY MEASURED LONGITUDINAL DATA

Lei Ye, PhD

University of Pittsburgh, 2014

ABSTRACT

For longitudinal data analyses, existing statistical methods can be used when the independent and dependent variables are measured at the same frequency. In Part 1 of this dissertation, we propose a three-step estimation procedure using local polynomial smoothing for longitudinal data where the variables to be handled are repeatedly measured with different frequencies within the same time period. We first inserted pseudo data for the less frequently measured variable. Then, standard linear regressions were fitted at each time point to obtain raw estimates. Lastly, local polynomial smoothing with analytical weights was applied to smooth the raw estimates. The results showed using analytic weights instead of a kernel function during smoothing is critical when the raw estimates have extreme values, or the association between the dependent and independent variables is nonlinear. In Part 2 of this dissertation, we propose another semiparametric estimation procedure to solve the same problem. After imputing pseudo data for the less frequently measured variable, local polynomial smoothing with analytical weights was applied to smooth the pseudo data for one subject at a time. Then, a suitable parametric mixed-effects model was applied. The results showed that using different types of analytic weights during smoothing produced similar results. Both proposed methods work better when the variances of the repeated measures are small and the within-subjects correlations are high.

Statement of Public Health Relevance: The proposed methods are good tools for exploring inconsistently measured longitudinal data. They provide estimation without losing information that has been collected. It is important to biomedical studies especially when many researchers are using diary-based methods to improve the data collection process. For example, paper diaries, personal digital assistants (PDA) and smart phones have been used in the weight loss clinical trials to collect intensive longitudinal data that reflect subjects' real life experiences and behaviors. The proposed methods can be used when the inconsistent measure problem is present in a longitudinal study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

ACKNOWLEDGMENTS: This work was supported by NIH grants NIH/NIDDK R01-DK071817, R01-DK071817-04S1, R01-DK071817-05S1, and NIH/NINR K24-NR010742, PI: LE Burke. The conduct of the study was also supported by the Data Management Core of the Center for Research in Chronic Disorders at the University of Pittsburgh School of Nursing (NIH-NINR P30-NR03924), the General Clinical Research Center (NIH-NCRR-GCRC 5M01-RR000056) and the Clinical Translational Research Center (NIH/NCRR/CTSA Grant UL1 RR024153) at the University of Pittsburgh.

1.0 INTRODUCTION

Longitudinal data such as repeated measurements are collected frequently in clinical trials and other scientific areas. Many researchers use diary-based methods to collect data. These methods started with simple paper diaries, later electronic devices such as personal digital assistants (PDA) or smartphones began to be used to improve the data collection process. These techniques enable one to record detailed information that is difficult to recall or is subject to reporting bias. Therefore this type of longitudinal data is usually intensively measured and has complicated patterns.

There are other types of longitudinal data that cannot be collected frequently because the processes are invasive or are time consuming, for example, lab tests or questionnaires. As a result, problems in existing statistical methodology arise when we examine the association between intensively measured longitudinal variables and those variable less intensively measured.

The motivation for this work was based on the longitudinal data collected in the Self-Monitoring And Recording using Technology (SMART) Trial, a single-center, 24-month clinical trial of overweight and obese adults seeking weight loss treatment(Burke et al., 2009). One of the study aims of the study was to compare weight loss between participants who were adherent to self-monitoring and participants who were not adherent to self-monitoring. The SMART Trial had 210 participants randomized to three treatment groups: Paper Diary, PDA, or PDA with

daily tailored feedback (PDA+FB). The traditional weight-loss intervention of group-based standard behavioral treatment (SBT) sessions was provided to each treatment group. SBT meetings were held weekly for the first 4 months, biweekly for 8 months and then monthly for 12 months.

The only difference among the three treatment groups was the tools used for self-monitoring. Participants in the paper diary group were given standard paper diaries, and were asked to record all foods eaten and their calorie and fat content and to calculate subtotals using a nutritional reference book. Participants in the PDA group were given a PDA to self-monitor diet. The PDA+FB group had feedback software that provided a daily message based on participants' entries. The PDAs had a database of foods and nutrient contents so the participant had to only search and select the food and enter the portion size. Subtotals were automatically calculated throughout the day.

Participants turned in their paper diaries at each intervention session; those with the PDA had the data uploaded to a desktop computer during the group session. Adherence to self-monitoring was defined as whether participants recorded at least 50% of their daily calorie goal in the diaries. If participants failed to turn in diaries, nonadherence for self-nomitoring was assumed. As a result there were 43 repeated measurements of adherence to self-monitoring for each participant.

The primary outcome was the participants' weight assessed at baseline and semi-annually. At each assessment, the participants' weight was measured by study staff using a digital scale and following an overnight fast. Therefore, there were five repeated measurements of weight for each participant.

Using parametric mixed-effects model to examine the effect of adherence to self-monitoring on weight involves reducing the dimension of the adherence data (Burke et al., 2012). However, the advantages of collecting detailed longitudinal data through the diaries are not fully exploited when using the aggregated data. Thus, the main objective of this dissertation is to propose statistical methods that can fully use both weight (five measurements) and adherence data (43 measurements) and to estimate the association between these two variables. Specific aims include:

Aim 1: We propose a three-step estimation procedure using local polynomial smoothing with analytical weights to analyze inconsistently measured longitudinal data.

Aim 2: We propose another semiparametric estimation procedure for consistently measured longitudinal data to model inconsistently measured longitudinal data.

In Chapter 2, we give a critical review of the literature related to parametric mixed-effects modeling, local polynomial smoothing techniques, semiparametric mixed-effects modeling for consistently measured longitudinal data. in Chapter 3, we present the completed work on Aim 1 and in Chapter 4, we present the completed work on Aim 2.

2.0 LITERATURE REVIEW

2.1 INTRODUCTION

In longitudinal studies, when variables are repeatedly measured on each study participant at different frequencies during the same time period, the dimensions of the two variables are different. This unbalanced structure causes problems for modelling the relationship between these variables. We review here the literature for analyzing consistently measured longitudinal data as these are the methods we wish to extend. These include parametric mixed-effects models, local polynomial smoothing, semiparametric mixed-effects models and two-step estimation of functional linear models.

2.2 PARAMETRIC MIXED-EFFECTS MODELS

In the longitudinal studies, participants are measured repeatedly over time which allows researchers to study their experiences and behaviors directly. Because the cluster of observations for each subject are correlated, the within-subjects correlation must be taken into account to draw valid scientific inferences. Parametric mixed-effects models, including linear and nonlinear mixed-effects models, are powerful tools for modeling longitudinal data. Linear mixed-effects

models are used when the association between a longitudinal response variable and its covariates are linear with normally distributed errors, while nonlinear mixed-effects models are used when the relationship is not linear and may have errors do not follow a normal distribution. (Demidenko, 2004; Diggle, 1988).

2.2.1 Notation

Let t_{ij}, $j = 1, 2, \cdots, T_i$; $i = 1, 2, \cdots, n$, be the distinct time points for subject i where data were collected. Let Y_{ij} be the response variable and $\mathbf{X}_{ij}$ be the covariates for the i^{th} subject at time t_{ij}. The data have the form:

$$\left(t_{ij}, \mathbf{X}_{ij}, y_{ij}\right), j = 1, \cdots, T_i; i = 1, 2, \cdots, n,$$

where $\mathbf{X}_{ij} = (X_{ij1}, X_{ij2}, \cdots, X_{ijd})^T$ are the d covariates measured at time t_{ij}. The interest is to examine the association between the response and its covariates.

2.2.2 Linear mixed-effects models

Linear mixed-effects models were introduced to capture the change in the response variable and account for the within-subjects correlation (Harville, 1976, 1977; Laird & Ware, 1982). Assuming subjects are independent from each other, the linear mixed-effects model can be written as:

$$\boldsymbol{y}_i = \mathbf{X}_i\beta + \boldsymbol{Z}_i b_i + \epsilon_i, \tag{2.2.1}$$

$$b_i \sim N(0, \boldsymbol{D}),\ \epsilon_i \sim N(0, \boldsymbol{R}_i),\ i = 1, 2, \cdots, n,$$

where $\boldsymbol{y}_i = [y_{i1}, \cdots, y_{iT_i}]^T$, $\mathbf{X}_i = [\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{iT_i}]^T$, $\mathbf{Z}_i = [\boldsymbol{z}_{i1}, \cdots, \boldsymbol{z}_{iT_i}]^T$ and $\epsilon_i = [\epsilon_{i1}, \cdots, \epsilon_{iT_i}]^T$. The

y_{ij} and ϵ_{ij} are the response and the error of the i^{th} subject's j^{th} measurement, respectively. The fixed effects parameter β $(d \times 1)$ and random effects parameter b_i $(q \times 1)$ need to be estimated, and $\mathbf{X}_i$ and $\boldsymbol{Z}_i$ are the corresponding fixed effects and random effects covariates. In model (2.2.1), $\boldsymbol{D}$ and $\boldsymbol{R}_i$ are the variance components, and b_i and ϵ_i are assumed to be independent with normal distributions. The correlation among the repeated measurements is introduced through the between-subject variation term $\mathbf{Z}_i \boldsymbol{D}\, \mathbf{Z}_i{}^T$ and the within-subject variation matrix $\boldsymbol{R}_i$.

2.2.3 Nonlinear mixed-effects models

A nonlinear mixed-effects model is generalized from a linear mixed-effects model when a longitudinal response variable cannot be written as a linear function of its covariates. The model has the form (Davidian & Giltinan, 1995; Vonesh & Chinchilli, 1996):

$$\boldsymbol{y}_i = f(\mathbf{X}_i, \beta_i) + \epsilon_i,\ \boldsymbol{\beta}_i = d(a_i, \boldsymbol{\beta}, \boldsymbol{b}_i), \qquad (2.2.3)$$

$$\boldsymbol{b}_i \sim N(0, \boldsymbol{D}),\ \epsilon_i \sim N(0, \boldsymbol{R}_i),\ i = 1, 2, \cdots, n,$$

where $f(\mathbf{X}_i, \boldsymbol{\beta}_i)$ is a known function of design matrix $\mathbf{X}_i$ and subject-specific parameters $\boldsymbol{\beta}_i$, and $d(\boldsymbol{a}_i, \boldsymbol{\beta}, \boldsymbol{b}_i)$ is a known function of the fixed effect vector β, the random effect vector b_i and between-subjects covariate vector $\boldsymbol{a}_i$.

Parameters of the mixed-effects models can be estimated using Maximum Likelihood Estimation (MLE) or Restricted Maximum Likelihood Estimation (REML) (Dempster, Rubin, & Tsutakawa, 1981; Laird, Lange, & Stram, 1987; Vonesh & Chinchilli, 1996). The Akaike Information Criteria (AIC) and the Bayesian Information Criteria (BIC) can be used as information criterias of goodness of fit (Ngo & Brand, 2002; Pinheiro & Bates, 2000). However, the successful application of a parametric mixed-effects model to longitudinal data depends

heavily on the assumptions of the model. Sometimes these assumptions may not be met. In this case, the parametric models are extended to nonparametric models that do not have distribution assumptions.

2.3 LOCAL POLYNOMIAL SMOOTHING

Parametric mixed-effects models are useful tools for modeling the relationship between a response variable and its covariates in longitudinal studies. However, in some applications, parametric models can be too restrictive or limiting, because they assume that the outcome has a certain distribution and the underlying regression function is known. To overcome this difficulty, nonparametric and semiparametric models have been proposed for longitudinal data. There are many existing smoothing techniques including local polynomial smoothing, locally weighted scatter plot smoothing (LOWESS) and splines. The basic idea of these techniques is to let the data determine the most suitable function forms. All of these nonparametric methods can be incorporated into longitudinal data analysis. We focus here on local polynomial smoothing (Fan & Gijbels, 1992; Muller, 1987; Wand & Jones, 1995) and LOWESS (Cleveland, 1979).

2.3.1 Notation

Let (t_i, y_i), $i = 1, 2, \cdots, n$, be an independent and identically distributed observations from a population (T, Y), where t_i are equally spaced time points in an interval of interest. Our interest is to estimate the conditional expectation of y_i

$$f(t) = \mathrm{E}(y_i|t_i = t),\ i = 1, 2, \cdots, n, \tag{2.3.1}$$

and the derivatives $f'(t)$, $f''(t)$,$\cdots$, and $f^p(t)$ can also be estimated.

2.3.2 General degree local polynomial smoothing

The main idea of local polynomial smoothing is to locally approximate the $f(t)$ in (2.3.1) by a polynomial of certain degree. Suppose that the $(p+1)^{\text{th}}$ derivative of $f(t)$ at the point t_0 exists, for t in a local neighborhood of t_0, a Taylor series gives

$$f(t) \approx f(t_0) + f'(t_0)\,(t - t_0) + \frac{f''(t_0)}{2!}(t - t_0)^2 + \cdots + \frac{f^p(t_0)}{p!}(t - t_0)^p. \tag{2.3.2}$$

The polynomial is fitted locally by a weighted least squares regression by minimizing

$$\sum_{i=1}^{n}[y_i - \sum_{j=0}^{p}\beta_j\,(t_i - t_0)^j]^2\,K_h(t_i - t_0), \tag{2.3.3}$$

where $K_h \geq 0$ is a kernel function re-scaled by a constant h ($h > 0$) called the bandwidth that controls the size of the local neighborhood $I_h\,(t_0)$ where the local smoothing is conduct. Let

$$I_h\,(t_0) = [t_0 - h, t_0 + h]. \tag{2.3.4}$$

The kernel function, K_h, determines how much the observations in the local neighborhood $I_h\,(t_0)$ contribute to the fit at t_0.

Let $\hat{\beta}_j$, $j = 0,\cdots,p$, be the solution to the weighted least squares problem (2.3.3), and $\hat{f}_h^{(j)}(t_0)$, be the estimate of the r^{th} derivative $f^{(j)}(t_0)$. Then

$$\hat{f}_h^{(j)}(t_0) = j!\,\hat{\boldsymbol{\beta}}_j\ ,\ j = 0,1,\cdots,p.$$

Therefore the p^{th} degree local polynomial kernel smoothing estimate of $f(t_0)$ is $\hat{\beta}_0$. We get a smoothed line after solving the weighted least squared problem (2.3.3) for all points in the domain of interest.

The design matrix of model (2.3.3) can be written as:

$$\boldsymbol{X} = \begin{bmatrix} 1 & (t_1 - t_0) & \cdots & (t_1 - t_0)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (t_n - t_0) & \cdots & (t_n - t_0)^p \end{bmatrix},$$

let

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \widehat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$$

and

$$\mathbf{W} = diag\,[K_h(t_1 - t_0), \cdots, K_h(t_n - t_0)].$$

Then the weighted least squares problem (2.2.3) can be expressed as

$$(\boldsymbol{y} - \boldsymbol{X\beta})^{\boldsymbol{T}} \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{X\beta}) \tag{2.3.5}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)^T$. The solution vector is

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X^T W X})^{-1} \boldsymbol{X^T W Y}. \tag{2.3.6}$$

2.3.3 Order of polynomial fit

A local constant or linear fit is appropriate in a flat neighborhood, but higher order fits are preferable at peaks and valleys. The local constant smoother called Nadaraya-Watson estimator is a local polynomial smoother with $p = 0$. It fits the data in the local neighborhoods with constants that minimize

$$\sum_{i=1}^{n} (y_i - \beta_0)^2 K_h(t_i - t_0).$$

It has a slower convergence rate at the region boundary than on the interior of the region because fewer data points are in the defined local neighborhoods near the boundary (Cheng, Fan, & Marron, 1993).

The local linear smoother is a local polynomial smoother with $p = 1$ (Fan & Gijbels, 1992). It fits the data within local neighborhoods with constants that minimize

$$\sum_{i=1}^{n}[y_i - \beta_0 - (t_i - t_0)\beta_1]^2 K_h(t_i - t_0).$$

This smoother does not have a boundary effect and convergence rate is the same at any point in the region (Cheng et al., 1993).

Usually the choice of the local polynomial fitting degree p is not as critical as the choice of the bandwidth, h. However, an odd p is better for curve estimation for the following reasons. Even order approximations have the same asymptotic variance as their consecutive odd order approximations, but the asymptotic variances are smaller for odd order approximations as compared to consecutive even order approximations (Ruppert & Wand, 1994). As the order of the approximation increases, the bias decreases, but the asymptotic variance and computational time increases (Fan, 1992; Hastie & Loader, 1993). Therefore, a low odd order approximation has been recommended (Wand & Jones, 1995).

2.3.4 Kernel function

The kernel function, K, in the local polynomial smoothing is usually a symmetric probability density function used to decide how much the observations contribute to the fit at t_0 by assigning weights to each observation in the local neighborhood I_h (t_0). Widely used kernel functions include the uniform kernel, Gaussian Kernel, Epanechnikov kernel, Biweight kernel and Triweight kernel. These kernels are all members of the symmetric Beta family.

When the uniform kernel is used, all of the observations within the local neighborhood contribute equally by assigning the same weight. When the Gaussian Kernel, Epanechnikov kernel, Biweight kernel or Triweight kernel functions are used, the contributions of the

observations are determined by the distance, between t_i and t_0. The shorter the distance the bigger the contribution will be (higher weights). The choice of a kernel is usually not crucial, but the Epanechnikov kernel is known as the "ideal" kernel function for local polynomial smoothing (Fan, Gijbels, Hu, & Huang, 1996).

2.3.5 Bandwidth selection

The bandwidth, h, specifies the size of the local neighborhood. A good choice of bandwidth produces small prediction error that is quantified by Mean Squared Error (MSE) of the local polynomial smoother (Fan et al., 1996). The MSE combines both variance and bias of the estimates

$$MSE\ [(\hat{f}_h(t_0)] = Bias^2[(\hat{f}_h(t_0)] + Var\ [(\hat{f}_h(t_0)]$$

$$Bias\ [(\hat{f}_h(t_0)] = E[\hat{f}_h(t_0)] - f(t_0)$$

$$Var\ [(\hat{f}_h(t_0)] = E\{\hat{f}_h(t_0) - \mathrm{E}[\hat{f}_h(t_0)]\}^2.$$

The bandwidth controls both the bias and the variance of the local polynomial smoother. When h is small, few observations fall within the local neighborhood, so $\hat{f}_h(t_0)$ is well estimated with a small bias, but small numbers of observations produce a large variance. For a similar reason, when h is large, many observations fall in the local neighborhood so that $\hat{f}_h(t_0)$ is estimated with a large bias but with a small variance.

Sometimes it is appropriate to choose the bandwidth subjectively by looking at several smoothed lines with different bandwidths. The bandwidth can also be selected automatically from the data by minimizing MSE of $\hat{f}_h(t_0)$. This bandwidth selector is typically based on cross-validation ideas (Wand & Jones, 1995).

2.3.6 Locally Weighted Scatter plot Smoothing

Local polynomial smoothing can be influenced by extreme observations in the response variable that produce an undesirable smoothed line. In this situation, a Locally Weighted Scatter plot Smoothing (LOWESS) is preferred, because it is robust against extreme observations (Cleveland, 1979; Cleveland & Devlin, 1988). During the fitting of LOWESS, residuals of a local polynomial smoother are evaluated, and robust weights are assigned to each residual with large residuals given small robust weights. Then the local polynomial smoothing is fitted again with new weights, which are the product of the kernel weights at the first fit and the robust weights assigned to each residual. Therefore, observations with large residuals during the first fit are down weighted in the second fit. This procedure is done iteratively until the results become stable.

The local polynomials fitted in local neighborhoods are usually first or second order because higher order polynomials tend to over fit the data and are numerically unstable. The kernel function usually used in LOWESS is the tri-cube kernel:

$$K(t) = \frac{70}{81}(1 - |t|^3)^3 I_{[|t|\leq 1]}.$$

It assigns higher weight to the observations that are close to the point where the response is being smoothed. The weight is calculated by scaling the distance between each observation and the point of estimation (d_i) to the maximum absolute distance (d_q) in the local neighborhood

$$w_i = \begin{cases} [1 - (d_i/d_i)^3]^3, & d_i < d_q \\ 0, & d_i \geq d_q \end{cases}$$

The local neighborhoods are defined by a nearest neighbor bandwidth α ($0 < \alpha \leq 1$). The data used in each local fit contain n×α observations that are closest to the point where the response is being smoothed. The larger the α, the smoother the fit but the greater the bias.

Reasonable values of the smoothing parameter α lie between 0.2 and 0.8 for most LOWESS applications (Cleveland, 1979).

Robust weights are calculated using the residuals. Let r_i, i = 1, 2, ⋯, n, be the residual of the i^{th} observation, and M be the median of the absolute values of the residuals in the first local polynomial fit. Robust weights are given by the bisquare function as

$$rw_i = \begin{cases} [1 - (r_i/6M)^2]^2, & |r_i| < 6M \\ 0, & |r_i| \geq 6M \end{cases}$$

New weights, the product of kernel weights and robust weights, are used in the next iteration of local polynomial fit. This procedure is repeated N times. Cleveland (1979) recommends N=3.

2.4 SEMIPARAMETRIC MIXED-EFFECTS MODELS

Parametric models have restrictive assumptions, but they are efficient if models are correctly specified. In contrast, nonparametric models are robust against model assumptions, but the fitting procedure is complicated. Semiparametric models are a compromise that have features of both parametric and nonparametric models.

2.4.1 Notation

Let t_{ij}, $j = 1, 2, \cdots, T_i$; $i = 1, 2, \cdots, n$, be the distinct time points where data were collected for subject i. Let Y_{ij} be the response variable and $\mathbf{X}_{ij}$ be the covariates for the i^{th} subject at time t_{ij}. The data have the form:

$$(t_{ij}, \mathbf{X}_{ij}, y_{ij}), j = 1, \cdots, T_i; i = 1, 2, \cdots, n,$$

where $\mathbf{X}_{ij}$= (X_{ij1}, X_{ij2},⋯, X_{ijd})^T are the d covariates measured at time t_{ij}. Our interest is to examine the association between the response and covariates.

2.4.2 Model specification

There are parametric components and nonparametric components in the semiparametric mixed-effects models. The parametric components are used to model factors that affect the response parametrically while the nonparametric components are used to model factors that affect the response nonparametrically (Ruppert, Wand, & Carroll, 2003; Zeger & Diggle, 1994; Zhang, Lin, Raz, & Sowers, 1998). The model has the form

$$y_{ij} = \boldsymbol{X}_{ij}^T\boldsymbol{\alpha} + \eta(t_{ij}) + \boldsymbol{h}_{ij}^T\boldsymbol{a}_i + \boldsymbol{v}_i(t_{ij}) + \epsilon_{ij}, \qquad (2.4.1)$$

$$j = 1, 2, \cdots, T_i;\ i = 1, 2, \cdots, n,$$

where $\boldsymbol{X}_{ij}^T\boldsymbol{\alpha}$ and $\eta(t_{ij})$ are the parametric and nonparametric fixed effects components respectively, and $\boldsymbol{h}_{ij}^T\boldsymbol{a}_i$ and $\boldsymbol{v}_i(t_{ij})$ are their corresponding random components that incorporate the within-subjects correlation. The response at time t_{ij} depends on time nonparametrically via a smoothing function $\eta(t)$, and parametrically on covariates $\mathbf{X}_{ij}$. Similarly the random effect at time t_{ij} depends on time nonparametrically via a smoothing function $\boldsymbol{v}_i(t)$, and parametrically on covariates $\mathbf{h}_{ij}$. Vector $\boldsymbol{\alpha}$ contains coefficients that need to be estimated, and ϵ_{ij} is the error term. It is assumed that

$$\boldsymbol{a}_i \sim N(0, \boldsymbol{D}_a),\ v_i(t) \sim GP(\mu, \gamma),\ E\,[\boldsymbol{a}_i\boldsymbol{v}_i(t)\,] = \gamma_a(t),\ \epsilon_i = [\epsilon_{i1}, \cdots, \epsilon_{iT_i}]^T \sim N(0, \boldsymbol{R}_i),$$

where $GP(0, \gamma)$ is a Gaussian process with mean function $\mu(t)$ and covariance function $\gamma(s, t)$.

When the nonparametric fixed effects and random effects components are dropped, the model (2.4.1) becomes the usual linear mixed-effects model $y_{ij} = \boldsymbol{X}_{ij}^T\alpha + \boldsymbol{h}_{ij}^T\boldsymbol{a}_i + \epsilon_{ij}$. Other

models can also be obtained for specific research questions by dropping one or two components from the model (2.4.1).

2.4.3 Local polynomial approximation

Local polynomial smoothing can be used to estimate $\eta(t_{ij})$ and $v_i(t_{ij})$ in model (2.4.1). Assuming the functions have a $(p+1)^{\text{th}}$ derivative at each time point, by Taylor series, $\eta(t_{ij})$ and $v_i(t_{ij})$ can be estimated by a p^{th} degree polynomial within a neighborhood of t_0:

$$\eta(t_{ij}) \approx \eta(t_0) + \eta'(t_0)\,(t_{ij} - t_0) + \frac{\eta''(t_0)}{2!}(t_{ij} - t_0)^2 + \cdots + \frac{\eta^{(p)}(t_0)}{p!}(t_{ij} - t_0)^p = \boldsymbol{k}_{ij}^T\boldsymbol{\beta},$$

$$v_i(t_{ij}) \approx v_i(t_0) + v_i'(t_0)\,(t_{ij} - t_0) + \frac{v_i''(t_0)}{2!}(t_{ij} - t_0)^2 + \cdots + \frac{v_i^{(p)}(t_0)}{p!}(t_{ij} - t_0)^p = \boldsymbol{k}_{ij}^T\boldsymbol{b}_i,$$

where $\boldsymbol{k}_{ij} = [1, t_{ij} - t_0, \cdots, (t_{ij} - t_0)^p]^T$, $j = 1, 2, \cdots, T_i$; $i = 1, 2, \cdots, n$, and

$$\boldsymbol{\beta} = [\eta(t_0), \eta'(t_0), \cdots, \frac{\eta^{(p)}(t_0)}{p!}]^T,$$

$$\boldsymbol{b}_i = [v_i(t_0), v_i'(t_0), \cdots, \frac{v_i^{(p)}(t_0)}{p!}]^T.$$

Within a local neighborhood of t_0, the model (2.4.1) can be reasonably approximated by:

$$y_{ij} = \boldsymbol{X}_{ij}^T\boldsymbol{\alpha} + \boldsymbol{k}_{ij}^T\boldsymbol{\beta} + \boldsymbol{h}_{ij}^T\boldsymbol{a}_i + \boldsymbol{k}_{ij}^T\boldsymbol{b}_i + \boldsymbol{\epsilon}_{ij}, \qquad (2.4.2)$$

$$j = 1, 2, \cdots, T_i;\ i = 1, 2, \cdots, n,$$

and $\boldsymbol{b}_i \sim N(0, \boldsymbol{D}_b)$. The fixed effect $\boldsymbol{\beta}$ and the covariance matrix $\boldsymbol{D}_b$ are functions of t_0.

A bandwidth h needs to be carefully chosen for the semiparametric mixed-effects model. When the bandwidth is too big, $\eta(t_{ij})$ and $v_i(t_{ij})$ can over smooth and lose important information. When the bandwidth is too small, $\eta(t_{ij})$ and $v_i(t_{ij})$ can have large variances. There are criterias that can be used to select proper bandwidth for a semiparametric mixed-effects

model, for example, the “leave-one-subject-out” cross-validation and the “leave-one-point-out” cross-validation (Ruppert, Sheather, & Wand, 1995).

Because model (2.4.1) has a nonparametric mixed-effects model $[\eta(t_{ij}) + v_i(t_{ij})]$ part and a linear mixed-effects model $(\boldsymbol{X}_{ij}^T\boldsymbol{\alpha} + \boldsymbol{h}_{ij}^T\boldsymbol{a}_i)$ part, backfitting strategies (Hastie & Tibshirani, 1990) can be employed to estimate the parameters by iteratively fitting a standard linear mixed-effects model and a nonparametric mixed-effects model. The process is: Given the current estimate of a and a_i, fit model (2.4.1) and obtain the estimates of $[\eta(t_{ij}) + v_i(t_{ij})]$, then given the current estimates of $[\eta(t_{ij}) + v_i(t_{ij})]$, fit (2.4.1) using a standard linear mixed-effects model and obtain $(\boldsymbol{X}_{ij}^T\boldsymbol{\alpha} + \boldsymbol{h}_{ij}^T\boldsymbol{a}_i)$. This fitting procedure is simple, but inferences about the model are not easy to draw.

Semiparametric mixed-effects models are flexible as they can employ any existing smoothing techniques to estimate $\eta(t_{ij})$ and $v_i(t_{ij})$. Local polynomial smoothing works better when the data have a small range of values, otherwise smoothing splines can be used (Wu & Zhang, 2006).

2.5 TWO-STEP ESTIMATION OF FUNCTIONAL LINEAR MODELS

Parametric, nonparametric and semiparametric mixed-effects models are popular methods for analyzing longitudinal data. The selection of an appropriate parametric mixed-effects model depends heavily on whether data meet the model assumptions. As a result, smoothing techniques, including splines and local polynomial smoothing, have been proposed to estimate coefficients nonparametrically, but these methods are computationally intensive especially when the number

of covariates is large. A two-step procedure has been proposed to overcome this computational disadvantage (Fan & Zhang, 2000). The method is suitable for longitudinal data where the response variable and associated covariates are collected at the same scheduled time points for all subjects.

2.5.1 Notation

Let $t_{ij}, j = 1, 2, \cdots, T_i; i = 1, 2, \cdots, n,$ be the distinct time points where data were collected. Let Y_{ij} be the response variable and $\mathbf{X}_{ij}$ be the covariates for the i^{th} subject at time t_{ij}. The data have the form of

$$\left(t_{ij}, \mathbf{X}_{ij}, y_{ij}\right), j = 1, \cdots, T_i; i = 1, 2, \cdots, n,$$

Where $\mathbf{X}_{ij}$= (X_{ij1}, X_{ij2},$\cdots$, X_{ijd})^T are the d covariates measured at time t_{ij}. The interest is to examine the association between the response variable and covariates as well as the change of the association over time.

2.5.2 Raw estimates

At each given time t_j, let N_j be the number of subjects who have both observations of Y_{ij} and $\mathbf{X}_{ij}$. Let $\widetilde{\mathbf{X}}_j$ be the design matrix and $\widetilde{\mathbf{Y}}_j$ be the response vector. Then the standard linear model at time t_j is

$$\widetilde{\mathbf{Y}}_j = \widetilde{\mathbf{X}}_j \beta(t_j) + \tilde{e}_j, \tag{2.5.1}$$

where $\tilde{e}_j$ is the error term, and

$$E(\tilde{e}_j) = 0, cov(\tilde{e}_j) = r(t_j, t_j) I_{n_j},$$

where n_j is the number of elements in N_j. If $\widetilde{\mathbf{X}}_j$ has full rank d, then the standard least squares estimator of $\beta(t_j)$ is $\mathrm{b}(t_j) = (\widetilde{\mathbf{X}}_j{}^T\widetilde{\mathbf{X}}_j)^{-1}\widetilde{\mathbf{X}}_j{}^T\widetilde{\mathbf{Y}}_j$, with $E(b(t_j)) = \beta(t_j)$ and $cov(b(t_j) = r(t_j, t_j)(\widetilde{\mathbf{X}}_j{}^T\widetilde{\mathbf{X}}_j)^{-1}$. Let $b_r(t_j)$ be the r^{th} component of $\mathrm{b}(t_j)$ then

$$cov((b_r), b_r(t_k)|\mathrm{D}) = r(t_j, t_k)e_{r,d}^T(\widetilde{\mathbf{X}}_j{}^T\widetilde{\mathbf{X}}_j)^{-1}\widetilde{\mathbf{X}}_j{}^T M_{jk}\widetilde{\mathbf{X}}_k(\widetilde{\mathbf{X}}_k{}^T\widetilde{\mathbf{X}}_k)^{-1}e_{r,d}, \qquad (2.5.2)$$

where $\mathrm{D} = \{(\mathbf{X}_{ij}, t_j), j = 1, 2, \cdots, T; i = 1, 2, \cdots, n\}$ and $e_{r,d}$ is a d-dimension unit vector with one at its r^{th} entry. M_{jk} is an identity matrix that if the α^{th} entry of $\widetilde{\mathbf{Y}}_j$ and the β^{th} entry of $\widetilde{\mathbf{Y}}_k$ come from the same subject, the $(\alpha,\beta)^{\text{th}}$ entry of M_{jk} is one otherwise zero.

To estimate the covariance of $b_r(t_j), b_r(t_k)$, $r(t_j, t_k)$ needs to be estimated. Let $\widehat{\tilde{e}_j} = (I_{n_j} - p_j)\,\widetilde{\mathbf{Y}}_j$ be the residuals from the linear regressions where $p_j = \widetilde{\mathbf{X}}_j(\widetilde{\mathbf{X}}_j{}^T\widetilde{\mathbf{X}}_j)^{-1}\widetilde{\mathbf{X}}_j{}^T$. It follows that

$$E(tr(\hat{\tilde{e}}_j\hat{\tilde{e}}_k^T)) = tr\{(I_{n_k} - p_k)M_{jk}{}^T(I_{n_j} - p_j)^T\}\, r(t_j, t_k).$$

If $tr\{(I_{n_k} - p_k)M_{jk}{}^T(I_{n_j} - p_j)^T\} \neq 0$, then the estimator for $r(t_j, t_k)$ is

$$\hat{r}(t_j, t_k) = tr\{\hat{\tilde{e}}_j\hat{\tilde{e}}_k^T\}/\, tr\{(I_{n_k} - p_k)M_{jk}{}^T(I_{n_j} - p_j)^T\}. \qquad (2.5.3)$$

2.5.3 Refine the raw estimates

The raw estimates obtained in (2.5.1) need to be refined, because these raw estimates are not smooth and information from the neighboring time points has not been considered. An easy way to refine the raw estimates is to smooth the coefficient $\hat{\beta}_r(t)$ over time using one of existing smoothing techniques. Suppose $(p + 1)^{\text{th}}$ derivative of $\hat{\beta}_r(t)$ exists at any given time point and the q^{th} $(0 \leq q < p + 1)$ derivative can be estimated. Then a typical estimator is

$$\widehat{\beta_r^{(q)}}(\mathrm{t}) = \sum_{j=1}^T w_r(t_j, t)b_r(t_j)$$

Smoothing techniques like splines or local polynomial smoothing can be used to construct the weights $w_r(t_j, t)$. Also

$$E\left(\widehat{\beta_r^{(q)}}(\mathrm{t})\middle|\mathrm{D}\right) = \sum_{j=1}^{T} w_r(t_j, t)\beta_r(t_j),$$

$$Var\left(\widehat{\beta_r^{(q)}}(\mathrm{t})\middle|\mathrm{D}\right) = \sum_{j=1}^{T}\sum_{k=1}^{T} w_r(t_j, t)w_r(t_k, t)cov(b_r(t_j), b_r(t_j)|D).$$

by using (2.3.2) and (2.3.3), $cov(b_r(t_j), b_r(t_j)|D)$ can be estimated, and the ±2 standard error bands are be computed as

$$\widehat{\beta_r^{(q)}}(\mathrm{t}) \pm 2\{\widehat{Var}(\widehat{\beta_r^{(q)}}(\mathrm{t})\ |\mathrm{D})\}^{1/2}$$

Because the smoothing step only has one dimension, separate smoothing parameters can be used for different covariates. Visualization of the raw estimates can assist in picking appropriate smoothing parameters, and any existing smoothing parameter selector can also be employed.

Parametric mixed-effects model, semiparametric mixed-effects model and two-step estimation are powerful tools for examining associations between a response variable and its covariates when they are measured at the same frequency. However, there are times when the response variable and the covariates are measured at different frequencies during the same time period. In this case the existing methods for longitudinal data need to be modified to better solve the problem. In the next two chapters we will extend the semiparametric mixed-effects models and two-step estimation techniques to analyze inconsistently measured longitudinal data.

3.0 A THREE-STEP ESTIMATION PROCEDURE USING LOCAL POLYNOMIAL SMOOTHING FOR INCONSISTENTLY MEASURED LOGITUDINAL DATA

3.1 INTRODUCTION

We propose a nonparametric model approach to address the problem of modeling inconsistently measured longitudinal data. This method is based on the two–step estimation of functional linear models for longitudinal data where both response variable and its covariates are measured at the same scheduled time points for all subjects. We will extend this method by using local polynomial smoothing with analytical weights and apply the proposed method to a longitudinal weight loss trial where data are inconsistently measured. A simulation study will be used to assess our new approach.

3.2 PROPOSED METHOD

Let there be n subjects observed during time 0 to T. Y_{ij} is the outcome measured for the i-th subject at time t_{ij} $(0 \leq j \leq T)$. X_{ik} is the covariate for the i^{th} subject at time t_{ik} $(0 \leq k \leq T)$. Because the outcome is measured less frequently than the covariate, for each subject, when k=j both outcome and covariate are measured, and when there are no matching j for k only covariate is measured. The parametric or nonparametric mixed model cannot be applied to this type of data

directly because of the inconsistent measurement. We introduce a three-step estimation method to explore this type of data, which enables the use of all data with no loss of information by discarding or modifying the extra covariate measurements.

3.2.1 Step one – insert pseudo data

At time t_{ik} when only the covariate are measured, pseudo data points of the outcome will be inserted for every subject to create a new dataset. At times t_{iv} and t_{iu} $(0 \leq v < u \leq T)$ both the outcome and the covariate are measured, and between these two time points t_{ik} $(v < k < u)$, only covariate are measured. We assume that the change of outcome from time t_{iv} to t_{iu} is linear. By solving for a_i and b_i in function below we will get the straight line from Y_{iv} to Y_{iu} for subject i.

$$\begin{cases} Y_{iv} = a_i\, t_{iv} + b_i \\ Y_{iu} = a_i\, t_{iu} + b_i \end{cases} \qquad (3.2.1a)$$

$$Y_{ik} = a_i t_{ik} + b_i \qquad (3.2.1b)$$

Then by substituting t_{ik} $(v < k < u)$ into the function above we get pseudo data Y_{ik} that matches real measurements X_{ik} between times t_{iv} and $(t_{iu} 0 \leq v < u \leq T)$ for subject i. This procedure is done repeatedly in the same way between all of the adjacent time points t_{ij} for each subject to get all of the pseudo data of Y at the time points when only the covariate is recorded. After this step, a new dataset (t_{ik}, Y_{ik}, X_{ik}) is created.

Missing data are common in longitudinal studies. If data are missing for subject i at time t_{iv} or t_{iu} $(0 \leq v < u \leq T)$, there is no way to insert data for Y between these two time points using functions (3.4.1a) and (3.4.1b). Missing data will be inserted as pseudo data for Y in this situation.

3.2.2 Step two – raw estimates

For subjects at each time t_{ik} $(0 \leq k \leq T)$, a standard linear model (3.2.2) is fitted to get the raw estimates and standard errors (se_k) of $\boldsymbol{\beta}$

$$\boldsymbol{Y}(t_k) = (\boldsymbol{X}(t_k)\,\boldsymbol{\beta}(t_k) + e(t_k), \tag{3.2.2}$$

where $e(t_k)$ is the error term. Here, sample sizes of the local linear regressions will not be the same, due to missing data.

3.2.3 Step three – smooth raw estimates

Local polynomial smoothing will be used to smooth the raw estimates from step two. This step is a modification of local polynomial kernel smoothing. Rather than using kernel functions to assign weights during smoothing, only the analytical weights that indicate the importance of the raw estimates will be used.

Let p be the degree of the polynomial being fit. At time point $t_k(0 \leq k \leq T)$, the smoothed estimate $\boldsymbol{\beta}_{smooth(k)}$ are obtained by smoothing raw estimates in the local neighborhood $I_h\,(t_k) = [t_k - h, t_k + h]$. By using only analytical weight W, the smoothed estimate $\beta_{smooth(k)}$ at t_k is the value of estimated $\hat{\beta}_0$, where $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \cdots, \hat{\beta}_p)$ minimizes

$$\sum_{k=0}^{N}\left\{\beta_{raw(k)} - \beta_0 - \beta_1(t_k - t) - \cdots - \beta_p(t_k - t)^p\right\}^2 W.$$

Weighted least squares theory leads to the solution

$$\hat{\beta} = (\tilde{t}^T\, W\, \tilde{t}\,)^{-1}\, \tilde{t}^T\, W\, \tilde{\beta}_{raw}$$

where $\widetilde{\boldsymbol{\beta}}_{raw}$ is the vector of raw estimates from step two, and

$$\tilde{\boldsymbol{t}} = \begin{bmatrix} 1 & \mathrm{t}_1 - t_k & \cdots & (\mathrm{t}_1 - t_k)^{\mathrm{p}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathrm{t}_{\mathrm{n}} - t_k & \cdots & (\mathrm{t}_{\mathrm{n}} - t_k)^{\mathrm{p}} \end{bmatrix}$$

is an $n \times (p + 1)$ design matrix. $\boldsymbol{W}$ is an $n \times n$ diagonal matrix of analytical weights given by

$$\boldsymbol{W} = diag\{w_1, \ldots, w_n\}.$$

Because pseudo data were inserted in step one, the raw estimates $\beta_{raw\,(k)}$ do not have the same accuracy. The most accurate raw estimates will the ones that were estimated by fitting local linear regressions at t_k $(k = j)$ where Y_{ik} and X_{ij} were both measured, so the highest analytical weights are given to those raw estimates during smoothing. The raw estimates from the local linear regressions that used pseudo data but are close to the time of the real measurement are also given higher analytical weights than the raw estimates that are far from the time of real measurement. The measure of the time distance between a raw estimate at time $\mathrm{t_k}$ using pseudo data and the adjacent raw estimate using data of real measurement will be defined as

$$D_k = min\,(|t_k - t_{last\ real\ measure}| + 1, |t_k - \mathrm{t_{next\ real\ measure}}| + 1).$$

For example, at times t_{iv} and t_{iu} $(0 \le v < u \le T)$ both outcome and covariate are measured and between these two time points t_{ik} $(v < k < u)$, only the covariate is measured. D_v and D_u will equal 1 and D_k will be calculated as

$$D_k = min\,(|\mathrm{t_k} - \mathrm{t_v}| + 1, |t_k - t_u| + 1).$$

We define the analytical weights in four different ways using D_k and the standard error to reflect the importance of the raw estimate.

Type 1: $w_k = \frac{1}{\sqrt{D_k}}$

Type 2: $w_k = \frac{1}{D_k}$

Type 3: $w_k = \frac{1}{\sqrt{D_k}} + \frac{1}{\sqrt{se_k}}$

Type 4: $w_k = \frac{1}{D_k} + \frac{1}{se_k}$

The performance of local polynomial smoothing also depends on the values chosen for bandwidth h and order of polynomial p, but how to choose the best h and p for the proposed model is not the focus of this paper. We will use the value p=3 as recommended by Wand and Jones (1995) as this has been shown to be adequate. For data that are not equally spaced choosing certain bandwidth will result in less data that fall into the window where the data are sparser. In this situation, we will use a proportion of the data to which we will fit a local polynomial smoother like LOWESS. Because useful values of the smoothing parameter typically lie in the range 0.25 to 0.5 for most LOWESS applications, we will use 0.3.

3.3 APPLICATION TO LONGITUDINAL DATA

The improving Self-monitoring in Weight Loss with Technology (SMART) trial was a 2-year clinical weight loss trial where the longitudinal data were inconsistently measured. Two hundred and ten participants were randomized to 3 self-monitoring groups: paper diary or personal digital assistant with or without feedback. All of the participants were asked to self-monitor food intake. Adherence to self-monitoring was a binary variable (yes/no) and was measured weekly for the first 4 months, biweekly for 8 months and then monthly for 12 months. The primary outcome was subject weight (kg), which was measured objectively using a digital scale at five clinical visits: baseline, 6^{th}, 12^{th}, 18^{th}, and 24^{th} months. Weights were assessed less frequently than adherence to self-monitoring during the two years, and one of the research interests was the association between the participants' weight and adherence to self-monitoring.

The association between the participants' weight and adherence to self-monitoring was assessed using a linear mixed-effects model. The response variable was modified by taking average values of subject weight between baseline and the 6^{th} month, the 6^{th} and 12^{th} month, the 12^{th} and 18^{th} month, and the 18^{th} and 24^{th} month for each subject. After the data modification every subject had both weight data and proportion of time adherent to self-monitoring data at baseline, 6^{th}, 12^{th}, 18^{th} and 24^{th} month, so the linear mixed-effects model or marginal mixed model can be easily applied to the modified data set (Burke et al., 2012). The drawback of this approach is that the detailed information on the changes of the adherence to self-monitoring among the five time points is lost. Also using the proportion of the time that subjects were adherent to self-monitoring during a 6-month period is too general and more difficult to understand than a binary (yes/no) adherence variable.

To apply the proposed method on the SMART data we inserted pseudo weight data for all subjects between baseline and 6^{th} month, 6^{th} and 12^{th} month, 12^{th} and 18^{th} month, and 18^{th} and 24^{th} month according to the self-monitoring data collection schedule. After this step, both weight and adherence to self-monitoring had 43 matching data records. Local linear regressions were fitted at each of the 43 time points. The dependent variable was weight in kilograms, which is continuous. The independent variable was adherence to self-monitoring, which is binary (yes group was treated as reference group). The coefficients for adherence to self-monitoring indicate the weight differences between subjects who were not adherent to self-monitoring and subjects who were adherent to self-monitoring.

Local polynomial smoothing using analytical weights was applied to smooth the 43 raw estimates of intercepts and coefficients of adherence from the local linear regressions. The degree of the polynomial was chosen to be 3, and about one third of the raw data were used in

each local fit. Four types of analytical weights as defined earlier that reflected the importance of the raw estimators were used. Results from Local polynomial kernel smoothing using Epanechnikov kernel were compared with the results from the proposed method. Results are shown in Figures 1.a and 1.b.

The standard errors of the coefficients from local linear regressions were also smoothed in the same way using analytical weight type 1 ($w_k = \frac{1}{\sqrt{D_k}}$). In Figures 1.c and 1.d, at each time point, the smoothed intercept and adherence coefficient using analytical weight type 1 is shown with one departure of the smoothed standard error.

In Figure 1, we can see that the intercept and adherence coefficients were not the same over time. For example the adherence effect started with negative values, but from Figure1.d we know that the adherence effect was not statistically significant at least for the first 10 weeks, because the one standard error band crossed value zero.

After 10 weeks the weight difference between subjects who were not adherent to self-monitoring and subjects who were adherent to self-monitoring kept increasing and peaked at the 50^{th} week. After 50 weeks the adherence was less strong until the 70^{th} week and increased slightly afterwards. Because the standard errors were very large after 90 weeks the adherence effect may not be statistically significant (Figure 1.d).

Comparing the smoothing results using the 4 different types of analytical weights, we can see that before the 50th week the 4 smoothed lines were so close that they are not distinguishable. After the 50th week, the lines were more distinct. The possible reason for the phenomenon is that the values of raw estimates at the 48th, 52nd and 76th week were more extreme compared to the data around them. Also the raw estimates at the 48th week are very close to the time when the real measurement happened and the raw estimates at the 76th and the

52nd weeks were from local linear regression using real measured data. As a result, these two raw estimates were given higher analytical weight and based on the scale of the weight, the fitted lines were different. Another possible reason is that the association between subjects' weights and adherence to self-monitoring was linear before the 50th week, and became nonlinear afterwards.

The results of local kernel polynomial smoothing that used the Epanechnikov kernel function are also shown in Figures 1.a and 1.b. The kernel smoothing line was the same as the other 4 smoothed lines before the 50th week. However, after the 50th week the kernel smoothing line had a "w" shape, which mirrors the data flow. For example, between the 58th and 70th weeks, the estimates of the kernel smoothing were not very different from the raw estimates. Because these raw estimates are far from the time of the real measurements at the 52nd and the 76th week, the values should be adjusted up like the other 4 fitted lines.

3.4 SIMULATION STUDY

A simulation study was conducted to examine the performance and features of the proposed method. The model of our simulation study attempted to mimic the SMART data set and was designed as follows. We chose N =200 subjects, and two true coefficient functions were set to the values based on the results of the SMART trial:

$$\beta_0^{true} = 90.5\text{-}0.2t_j$$

$$\beta_1^{true} = -1.1 + 0.4t_j - 0.003{t_j}^2$$

The t_j in weeks ($1 \le t_j \le 96,\ 1 \le j \le 43$) was defined as the time when the measurement was taken, and the measurement was weekly for the first 4 months, biweekly for 8 weeks, then

monthly for 12 months. The intercept effect is a straight line over time, and the covariate effect is a curve over time both based on 43 time points.

Let $\boldsymbol{X}_0(t)$ be a vector of 1's (200 by 1), and $\boldsymbol{X}_1(t)$ be a binomial random variable with probability of success $p = 0.5$. We sampled the errors from a multivariate normal distribution with mean 0 and a compound symmetric covariance structure (43 by 43). Different variance with high correlation ($\rho = 0.7$), medium correlation ($\rho = 0.5$) or low correlation ($\rho = 0.3$) of the compound symmetric covariance structure was used and results from them were compared. The simulated full data were the sum of the errors and the underlying true coefficients at each time point

$$Y_{ij} = \boldsymbol{X}_0\beta_0^{true}(t_{ij}) + \boldsymbol{X}_1\beta_1^{true}(t_{ij}) + \epsilon_{ij}, \quad j = 1,2,\cdots,T;\ i = 1,2,\cdots,N$$

Using the simulated full data, we fit local linear regressions at each time point to get 43 estimated β_1 ($\hat{\beta}_1^{simulate}$).

Using the proposed method, Y_{ik} is inserted between time points $t_{ij} = 1$ and $t_{ij} = 24$, $t_{ij} = 24$ and $t_{ij} = 48$, $t_{t_{ij}=ij} = 48$ and $t_{ij} = 72$, $t_{ij} = 72$ and $t_{ij} = 96$ using functions (3.2.1a) and (3.2.1b). Local linear regressions are fitted at each time point to get 43 raw estimates of β_1 ($\hat{\beta}_1^{raw}$). Local polynomial smoothing using analytical weights was applied to estimate smoothed β_1 ($\hat{\beta}_1^{smooth}$). The degree of the polynomial was set to be 3, and about one third of the raw estimates were used in each local fit. For each different covariance structures of the error term and each different analytical weight the process was repeated with 5000 replications.

The performance of the proposed method was measured by the Averaged Deviation (AD) defined as

$$AD = \left\{\frac{1}{5000}\sum_{q=1}^{5000}\sum_{j=1}^{43}(\hat{\beta}_{1jq}^{simulate} - \beta_{1jq}^{true})^2\right\} - \left\{\frac{1}{5000}\sum_{q=1}^{5000}\sum_{j=1}^{43}(\hat{\beta}_{1jq}^{smooth} - \beta_{1jq}^{true})^2\right\}$$

The simulation results are shown in Table 1. Smaller absolute values of AD indicate a better fit.

When the variance of the error term at each time point was small and repeated measures were highly correlated to each other, the fit of the proposed model was better, and using each type of analytical weight did not make a big difference. However, when the variance was larger than 35^2, the type of the proposed analytical weights had different performances.

Because the covariance matrix of a real longitudinal data typically does not have compound symmetry structure, a more realistic covariance structure was also used in the simulation study. The subjects in the SMART study reported their weight at the same scheduled time as self-monitoring. These data were not as accurate as the weights measured at clinical visit and a lot more data were missing. We fit a linear mixed-effects model using self-report weights as the outcome and adherence to self-monitoring as the covariate to estimate an R matrix. The variance $\widehat{\sigma}_{jj}^2$ in the estimated R matrix ranged from 430 to 570 and the AD of the simulation study using different analytical weights defined before are AD_1= 65.61, AD_2= 65.17, AD_3= 68.29 and AD_4= 68.26. The performance was good and each type of analytical weight produced similar results.

3.5 DISCUSSION

We have demonstrated the utility of a three-step estimation via local polynomial smoothing for longitudinal data where the outcome was measured less frequently than its covariate. Our method also works when covariates were measured less frequently than outcome, because it involves smoothing coefficients of local linear regressions. For the same reason, any type of generalized linear regressions can be implemented at step two, and no matter how many covariates are modeled, the fitting process will be fast. The results of this process have straightforward

interpretations. Not only does the method show the association between the dependent and independent variables, it also displays the change of the association over time.

Compared to applying classical mixed-effects models to inconsistently measured longitudinal data, the proposed method does not require modifying the data that are measured more frequently. As a result less information will be lost during the data analysis.

The proposed method does have limitations. When the time distance and standard error of the raw estimates that are used for computing analytical weight do not have the same scale, further modification is necessary. If the variation of the errors of the local linear regression is big and the correlation between them is small, the proposed method is less accurate. During the local linear regression fitting, although one can use as many covariates as needed, when there are interactions among these covariates, the final results will be hard to understand. Because withdrawing prematurely is a common occurrence in a longitudinal study, each local linear model may not have the same power. Lastly, the proposed method cannot estimate a response curve for each subject.

As data collecting techniques becomes more improved in the scientific studies, participants' experiences can be recorded as they occur in daily life. These intensive longitudinal data can provide us detailed information to understand human behaviors. The proposed method in this paper will be a useful tool for exploring these type of data when they are correlated with other less intensively measured variables.

4.0 A SEMIPARAMETRIC ESTIMATION PROCEDURE USING LOCAL POLYNOMIAL SMOOTHING FOR INCONSISTENTLY MEASURED LONGITUDINAL DATA

4.1 INTRODUCTION

Parametric mixed-effects models are frequently used to analyze longitudinal data where information is collected repeatedly on the same subject over time. However when a response variable and its covariates are measured at different frequencies, parametric mixed-effects models cannot be applied directly. We propose a semiparametric estimation procedure to address the problem of modeling inconsistently measured longitudinal data. We will demonstrate the proposed method on the SMART Trial data. A simulation study will be conducted to assess the performance of our approach.

4.2 PROPOSED METHOD

Let there be N subjects observed during time 0 to T. Y_{ij} is the outcome measured for the i^{th} subject at time t_{ij} $(0 \leq j \leq T)$, and $\boldsymbol{X}_{ik}$ is its covariate vector measured for the i^{th} subject at time t_{ik} $(0 \leq k \leq T)$. We assume that the outcome is repeatedly measured less frequently than its covariates. However, when the covariates are repeatedly measured less frequently than the

outcome the proposed method is also applicable. For each subject, when $k = j$ both the outcome and its covariates are measured, and when there are no matching j for k only covariates are measured. The parametric mixed-effects model cannot be applied to this inconsistently measured longitudinal data directly. We introduce a semiparametric estimation procedure to explore this type of data that can use all available data with no information loss instead of reducing the dimension of the more frequently measured covariate.

4.2.1 Insert pseudo data

When only the covariates are more intensively measured, pseudo data for the outcome are inserted for every subject. At any two adjacent times points, t_{iv} and t_{iu} $(0 \leq v < u \leq T)$, both outcome and covariate are measured and between these two time points t_{ik} $(v < k < u)$ only covariate are measured. We assume that the change of outcome from time t_{iv} to t_{iu} is linear. By solving for a_i and b_i in function (3.2.1a) we can get a straight line that connects Y_{iv} and Y_{iu} for the i^{th} subject. Then, by substituting t_{ik} $(v < k < u)$ repeatedly into the function (3.2.1b) above, we will get pseudo data Y_{ik} $(v < k < u)$ that matches the real measurements $\boldsymbol{X}_{ik}$ between times t_{iv} and t_{iu} $(0 \leq v < u \leq T)$ for the i^{th} subject. This procedure is done repeatedly in the same way between any two adjacent time points of the outcome to insert pseudo data at the time points where only covariates are measured. After this step, a new dataset (t_{ik}, Y_{ik}, X_{ik}) is created.

Missing data are common in longitudinal studies. If outcome data are missing for i^{th} subject at time t_{iv} or t_{iu} $(0 \leq v < u \leq T)$, there is no way to insert data between these two adjacent time points using functions (3.2.1a) and (3.2.1b) for the outcome. Missing data are inserted as pseudo data for the outcome under this situation.

4.2.2 Smooth pseudo data and apply parametric mixed-effects model

Local polynomial smoothing is used to smooth the pseudo data for one subject at a time. Rather than using a kernel function to assign weights, only analytical weights that indicate the accuracy of the data are used. Let p be the degree of the polynomial being fit. For the i^{th} subject at time t_k $(0 \le k \le T)$, the smoothed estimates $Y_{smooth(ik)}$ are obtained by smoothing the data in the local neighborhood $I_h\ (t_k) = [t_k - h, t_k + h]$. By using analytical weights, the smoothed estimate $Y_{smooth(ik)}$ at time t_k is the value of estimated $\hat{\beta}_0$, where $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_0, \cdots, \hat{\beta}_p)$ minimizes

$$\sum_{k=0}^{N}\left\{Y_{ik} - \beta_0 - \beta_1(t_k - t) - \cdots - \beta_p(t_k - t)^p\right\}^2 W_k.$$

weighted least squares theory leads to the solution

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{t}^T\, \boldsymbol{W}\, \boldsymbol{t}\,)^{-1}\boldsymbol{t}^T\boldsymbol{W}\, \boldsymbol{Y}\,,$$

Where $\boldsymbol{Y}$ is a vector of Y_{ik} that falls in the local neighborhood, and

$$\boldsymbol{t} = \begin{bmatrix} 1 & \mathrm{t}_1 - t_k & \cdots & (\mathrm{t}_1 - t_k)^{\mathrm{p}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathrm{t}_{\mathrm{n}} - t_k & \cdots & (\mathrm{t}_{\mathrm{n}} - t_k)^{\mathrm{p}} \end{bmatrix}$$

is an $n \times (p + 1)$ design matrix. $\boldsymbol{W}$ is an $n \times n$ diagonal matrix of analytical weights given by

$$\boldsymbol{W} = diag\{w_1, \ldots, w_n\}.$$

Because pseudo data were inserted, the value of Y_{ik} do not all have the same accuracy. The most accurate Y_{ik} are the ones at t_k $(0 \le k \le T)$ where both the outcome and its covariates were scheduled to be measured, so the highest analytical weights are given to these Y_{ik} during smoothing. The pseudo data that are close to the nearest real measurements are given higher analytical weights than the pseudo data that are far from the nearest real measurements. The measure of the time distance between a pseudo data and the nearest real measurement is defined as

$$D_k = min\ (|t_k - t_{last\ real\ measure}| + 1, |t_k - t_{next\ real\ measure}| + 1).$$

For example, at any two adjacent time points t_{iv} and t_{iu} $(0 \le v < u \le T)$ both the outcome and its covariates are measured and between these two time points at t_{ik} $(v < k < u)$ only covariates are measured. D_v and D_u will equal 1 and D_k will be calculated as

$$D_k = min\ (|t_k - t_v| + 1, |t_k - t_u| + 1).$$

We define the analytical weights in two different ways using D_k to reflect the accuracy of the Y_{ik} after imputation.

$$\text{Type 1: } w_k = \frac{1}{\sqrt{D_k}}$$

$$\text{Type 2: } w_k = \frac{1}{D_k}$$

The performance of the local polynomial smoothing also depends on the values chosen for the bandwidth h and the fitting order p. The focus of this paper is not how to choose the best h and p for the proposed model so we will use the value $p = 3$ as recommended by Wand and Jones (1995) as this has been shown to be adequate for local polynomial smoothing. For the example data where the time points are not equally spaced, choosing a fixed bandwidth h results in fewer data falling into the local neighborhood when the data are sparser. In this situation, we will use a certain proportion of the data in the nearest neighborhood to which local polynomial smoothing with analytical weights is applied, which is similar to LOWESS. Because useful values of the smoothing parameter a typically lie in the range 0.25 to 0.5 for the most LOWESS applications, we will use 0.3.

Drop-out is a common situation in longitudinal studies. As a result there may not be enough data to smooth even after imputation for some subjects. In this situation, the unsmoothed pseudo data are kept for the parametric mixed-effects modeling. After extending the less frequently measured variable by inserting and smoothing pseudo data, the variables in the dataset

will have the same amount of measurement. Depending on the research question being asked, any appropriate parametric mixed-effects model can be easily applied.

4.3 APPLICATION USING REAL LONGITUDINAL DATA

We used the SMART Trail study data as described in 3.3 to apply our proposed method. As done in 3.3, we first inserted pseudo data for subjects' weights between baseline and 6th month, 6th and 12th month, 12th and 18th month, 18th and 24th month according to the measure schedule of adherence to self-monitoring. After this step, every subject had both weight data and adherence to self-monitoring data at all 43 time points.

Local polynomial smoothing using only analytical weights was repeatedly applied to smooth the cluster of data for each subject. The degree of the polynomial was 3, and about one third of the data were used in each local fit. Two types of analytical weights defined earlier in 3.2, that reflected the importance of the data were used. When subjects did not come in for weight assessment, pseudo weight data could not be imputed using functions 3.2.1a and 3.2.1b. After imputation 173 subjects had 43 values for weight data and 19 subjects had 20 to 38 values for weight data. For these subjects, data were smoothed as described. There are 18 subjects who had fewer than 8 values of weight data after imputation. For these subjects, the data were too sparse to smooth and the pseudo data were kept without smoothing.

After smoothing, the less frequently measured weight data were extended to have the same amount of measurements as adherence to self-monitoring data. A parametric mixed-effects model was used to estimate the effect of adherence to self-monitoring on weight. When type 1 analytical weights $w_k = \frac{1}{\sqrt{D_k}}$ were used during smoothing, the results showed that subjects who

were adherent to self-monitoring weighed 1.2050kg less than subjects who were not adherent (p<.001). When type 2 analytical weights $w_k = \frac{1}{D_k}$ were used, the results showed that subjects who were adherent to self-monitoring weighed 1.2109kg less than subjects who were not adherent (p<.001). Using the different analytical weights produced similar results for SMART trial.

4.4 SIMULATION STUDY

In order to examine the performance and features of the proposed method a simulation study was conducted. The data that were simulated mimic the SMART Trial data and the simulation was designed as follows. We chose $N = 200$ subjects, and the two true coefficient functions were given as modelled after the SMART trial:

$$\boldsymbol{\beta}_0^{true} = 90.5 - 0.2t_j$$

$$\boldsymbol{\beta}_1^{true} = 2.5$$

Outcomes were measured weekly for the first 4 months, biweekly for 8 weeks, then monthly for 12 months at time t_j ($1 \leq t_j \leq 96,\ 1 \leq j \leq 43$). The intercept was a straight decreasing line, and the coefficient of the covariate was a small positive number.

Let $\boldsymbol{X}_0(t)$ be a vector of 1's (200 by 1), and $\boldsymbol{X}_1(t)$ be a vector of binomial random variables with success rate of 0.5. We sampled the errors from a multivariate normal distribution with mean 0 and a compound symmetric covariance structure (43 by 43). Different variances with high correlation ($\rho = 0.7$), medium correlation ($\rho = 0.5$) or low correlation ($\rho = 0.3$) of the compound symmetric covariance structure were used and results from them were compared.

The simulated full data were the sum of the errors and the underlying true coefficients at each time point

$$Y_{ij} = X_0\boldsymbol{\beta}_0^{true}(t_{ij}) + X_1\boldsymbol{\beta}_1^{true} + \epsilon_{ij}, \quad j = 1,2,\cdots,T;\ i = 1,2,\cdots,N$$

Using the simulated full data, we fitted parametric mixed-effects model to estimate β_1($\hat{\beta}_1^{simulate}$). This is the coefficient estimated under the situation that the dependent and its covariates are measured the same number of times.

Using the proposed method, $y_{ij}^{pseudo}$was inserted between time points $t_{ij} = 1$ and $t_{ij} =$ 24, $t_{ij} = 24$ and t_{ij} =48, $t_{t_{ij}}$= 48 and $t_{ij} = 72$, t_{ij} =72 and t_{ij} =96 using functions (1a) and (1b) to create a new dataset. Local polynomial smoothing using only analytical weights was applied repeatedly to smooth the cluster of data for each subject. The degree of the polynomial was set to be 3, and about one third of the raw estimates were used in each local fit. A parametric mixed-effects models was used on the smoothed data to estimate β_1 ($\hat{\beta}_1^{estimate}$). This is the coefficient estimate using the proposed method under the situation that the dependent variable is measured less frequently than its covariate. For each different covariance structure of the error term and each different analytical weight, the process was repeated with 500 replications.

The performance of the proposed method was measured by the Averaged Deviation (AD) defined as

$$AD = \left\{\frac{1}{500}\sum_{q=1}^{500}(\hat{\beta}_{1q}^{simulate} - \beta_{1q}^{true})^2\right\} - \left\{\frac{1}{500}\sum_{q=1}^{500}(\hat{\beta}_{1q}^{estimate} - \beta_{1q}^{true})^2\right\}$$

The simulation results are shown in Table 2. Smaller absolute values of AD indicate a better fit. When the variations of the repeated measures were small and the within-subject correlations were high, the performance of the proposed model was better. However, using different types of analytical weights during smoothing produced similar results.

In real longitudinal studies, the repeated measurements rarely have a compound symmetric covariance structure, so a more realistic covariance matrix was also used in the simulation study. The subjects in the SMART study self-reported their weight at the same scheduled time as adherence to self-monitoring. These data were not as accurate as the subjects' weight measured at clinical visits and more data were missing. However, by fitting a mixed model using self-reported weights and adherence to self-monitoring an R matrix was estimated. The variance and covariance in the estimated R matrix ranged from 430 to 570. The 500 pairs of $\hat{\beta}_1^{estimate}$ and $\hat{\beta}_1^{simulate}$ are plotted in the Figures 2 and 3. In both figures the dots were gathered around a 45 degree line, which means the performance of the proposed method was good. Because the $\hat{\beta}_1^{estimate}$ in each figure were estimated with different analytical weights and the two figures looked very similar, it appears that using different types of analytical weights will not make a big difference.

4.5 DISCUSSION

The application of the proposed semiparametric estimation procedure using local polynomial smoothing has been demonstrated on inconsistently measured longitudinal data where the outcome was measured less frequently than its covariates. However, this method also works when covariates are measured less frequently than the outcome or when some covariates are measured less frequently than the other covariates or the outcome, because this method extends the less frequently measured variables to have the same amount of measurements as the more frequently measured variables. The last step of the proposed method is applying parametric mixed-effects models, so missing data caused by random reasons can be easily handled.

The proposed method does not require reducing the dimension of the more frequently measured data, so less information is lost and the results are easier to understand. Because local polynomial smoothing has only two smoothing parameters and is conducted repeatedly on one dimensional data, the computing time is short even when the total number of subjects is big. Based on the research question, any suitable parametric mixed-effects model can be chosen after smoothing which makes the proposed method flexible.

There are limitations of the proposed method. If the less frequently measured variable is a binary outcome or multinomial outcome, pseudo data imputation and local polynomial smoothing with analytical weights do not work. When the variation of the repeated measures are large and the within-subjects correlations are small, the proposed method is less precise. Because subjects withdrawing from the trial is a common situation in longitudinal studies, some subjects may not have enough data to be smoothed even after imputation. If the number of subjects who do not have enough data to be smoothed is large, the results will not be accurate.

With the fast development of technology, data collection is becoming increasingly efficient. Participants' experiences can be recorded in real time in great detail which provide us opportunities to better understand human experiences and behaviors. Finding the best way to utilize all of the collected information is important. The proposed method in this paper will be a useful tool for exploring intensive longitudinal data when they are to be correlated with other less intensively measured longitudinal variables.

5.0 DISCUSSION

The applications of the two proposed methods have been demonstrated using SMART data where the outcome was measured less frequently than its covariate. Both methods involve imputing pseudo data for the less frequently measured variable and using local polynomial smoothing with analytical weights to adjust results. The difference is that in the first method the adjustment happens at the parameter level while in the second method the adjustment happens at data level. As a result there are similarities and differences between the two methods.

Comparisons between the two methods are summarized in Table 3. Both methods are flexible because they can be used when the outcome or its covariates are measured less frequently. They are easy to apply and produce straightforward results. The form of the analytical weights does not make a big difference in either method. However, for some situations only one of the methods work. For example, if there are interactions between covariates that need to be estimated or individual profiles are of interest, the second method works better. If the effect of the covariates varies over time, the first method provides better estimation. There are situations when neither of the methods work. For example when the repeated measures have large variation and small correlation, both methods produce inaccurate results. When the less frequently measured variable is binary or binomial, the imputation cannot be done which makes the application impossible.

The Public Health significance of the proposed methods is that they are good tools for exploring inconsistently measured longitudinal data. They provide estimation without losing information that has been collected. This is important to biomedical studies because as the data collection methods become better, finding ways to use all of the data are necessary. Researchers can choose one of the proposed methods or both to solve problems when the inconsistent measure is present in a longitudinal study.

6.0 FUTURE DIRECTIONS

In semiparametric mixed-effects models, the parametric components are used to model factors that affect the responses parametrically and the nonparametric components are used to model factors that affect the response nonparametrically. Both parametric components and nonparametric components can have fixed effects and random effects to incorporate the within subjects correlation. However, semiparametric mixed-effects models are only suitable when the response and covariates are repeatedly measured with the same frequencies. In future work, we will extend semiparametric modelling techniques to model inconsistently measured longitudinal data.

6.1 MODEL FORMULATION

Let there be N subjects observed during time 0 to T. Y_{ij} is the response variable for the i^{th} subject at time t_{ij} $(0 \leq j \leq T)$, and $\boldsymbol{X}_{ik}$ is the covariate vector for the i^{th} subject at time t_{ik} $(0 \leq k \leq T)$. Here the outcome is measured less frequently than the covariates. For each subject, when $k = j$, both the outcome and its covariate are measured, and when there are no matching j for k only the covariate is measured.

Pseudo data for the outcome will be inserted to create a new dataset assuming that the outcome changes linearly between the adjacent time points. This step is exactly the same as what was described in the proposed methods 1 and 2 in Chapters 3 and 4, respectively. After imputation, the outcome and its covariates in the new dataset (t_{ik},Y_{ik},$\boldsymbol{X}_{ik}$) will have the same amount of measurement. The proposed semiparametric mixed-effects model will have the form

$$y_{ik} = \boldsymbol{X}_{ik}^T\boldsymbol{\alpha} + \eta(t_{ik}) + \boldsymbol{h}_{ik}^T\boldsymbol{a}_i + \boldsymbol{v}_i(t_{ik}) + \boldsymbol{\epsilon}_{ik},$$

$$k = 1, 2, \cdots, T;\ i = 1, 2, \cdots, n,$$

where $\boldsymbol{X}_{ik}^T\boldsymbol{\alpha}$ and $\eta(t_{ik})$ are the parametric and nonparametric fixed effects components, respectively, and $\boldsymbol{h}_{ik}^T\boldsymbol{a}_i$ and $\boldsymbol{v}_i(t_{ik})$ are their corresponding random components that incorporate the within-subjects correlation. Vector $\boldsymbol{\alpha}$ contains the coefficients of the covariates, and ϵ_{ij} is the error at time t_{ij} that is not explained by the rest of the model. It is assumed that

$$\boldsymbol{a}_i \sim N(0, \boldsymbol{D}_a),\ \boldsymbol{v}_i(t) \sim GP(\mu, \gamma),\ E\left[\boldsymbol{a}_i\boldsymbol{v}_i(t)\right] = \gamma_a(t),\ \boldsymbol{\epsilon}_i = [\epsilon_{i1}, \cdots, \epsilon_{iT_i}]^T \sim N(0, R_i),$$

Where $GP(\mu, \gamma)$ is a Gaussian process with mean function $\mu(t)$ and covariance function $\gamma(s, t)$.

Local polynomial smoothing with analytical weights will be used in the nonparametric components ($\eta(t_{ik})$ and $v_i(t_{ik})$) of the model (4.2.2). Because pseudo data are inserted for the response, each data point does not have the same accuracy. The most accurate data will be the real measurements, so the highest analytical weights are given to those data. The pseudo data that are close to the real measurement are given higher analytical weights than the pseudo data that are far from the real measurement. The measure of the time distance between pseudo data at time t_k and the adjacent real measurement will be

$$D_k = min\left(|t_k - t_{last\ real\ measure}| + 1, |t_k - t_{\text{next real measure}}| + 1\right).$$

We will use analytical weights type 1 and 2 defined in 3.2 to reflect the importance of the response variable data.

6.2 APPLICATION, CHALLENGES AND POSSIBLE SOLUTION

We applied the proposed semiparametric mixed-effects model to the SMART data to estimate the effect of adherence to self-monitoring on subjects' weight. First we imputed pseudo data for subject's weight. After imputation each subject had 43 measurements of weight and adherence to self-monitoring. Because adherence to self-monitoring is a fixed effect the semiparametric mixed-effects model used for SMART data was in the form of

$$y_{ik} = \boldsymbol{X}_{ik}^{T}\boldsymbol{\alpha} + \eta(t_{ik}) + \boldsymbol{v}_i(t_{ik}) + \epsilon_{ik}, \quad k = 1, 2, \cdots, T;\ i = 1, 2, \cdots, n,$$

The covariate has no random effect while the smoothed part has both random and fixed effects to handle the within-subject correlation.

There are challenges to using local polynomial smoothing with random effects. First the local polynomial smoothing works in local neighborhoods. In order to smooth with the random effect, the local neighborhoods have to be big enough so that the model can converge. When the Taylor series expansion used has higher order of degree p there are more parameters to be estimated. As a result, bigger local neighborhoods are needed. Generally, large numbers of subjects require bigger local neighborhoods. The local neighborhood for the SMART data was defined as one third of the data that are close to the point to be smoothed. There were 210 subjects in SMART study, so the order of the smoothing had to be 1 to make the model converge. Using order of 1 in the local polynomial smoothing is not ideal because it is a local linear smoother.

The other challenge was that because the semiparametric mixed-effects model estimates both the mean function and the individual function during smoothing, the individual functions are pulled towards the mean function as showed in Figure 4. The graphs plotted weight data after

imputation and the smoothed line using the proposed model for 4 subjects from the SMART study. The individual smoothed line had a similar shape because the majority of the subjects tend to lose weight at the beginning and then regain. The individual curve is driven by the population curve rather than the analytical weights during smoothing which is not good for solving the problem of inconsistent measurements.

If the longitudinal data for all subjects had similar shapes the proposed method may work. However for data like those from SMART, not all subjects have the same pattern of losing and gaining weight so the proposed method needs further modification to better solve the problem. Future work should include a method to group subjects, so subjects in the subgroup have a similar shape based on the data. In this way, the number of subjects decreases which is good for the model convergence. Also in this case the population curve will not be too different from the individual curve to create unreasonable individual functions.

APPENDIX A: TABLES AND FIGURES

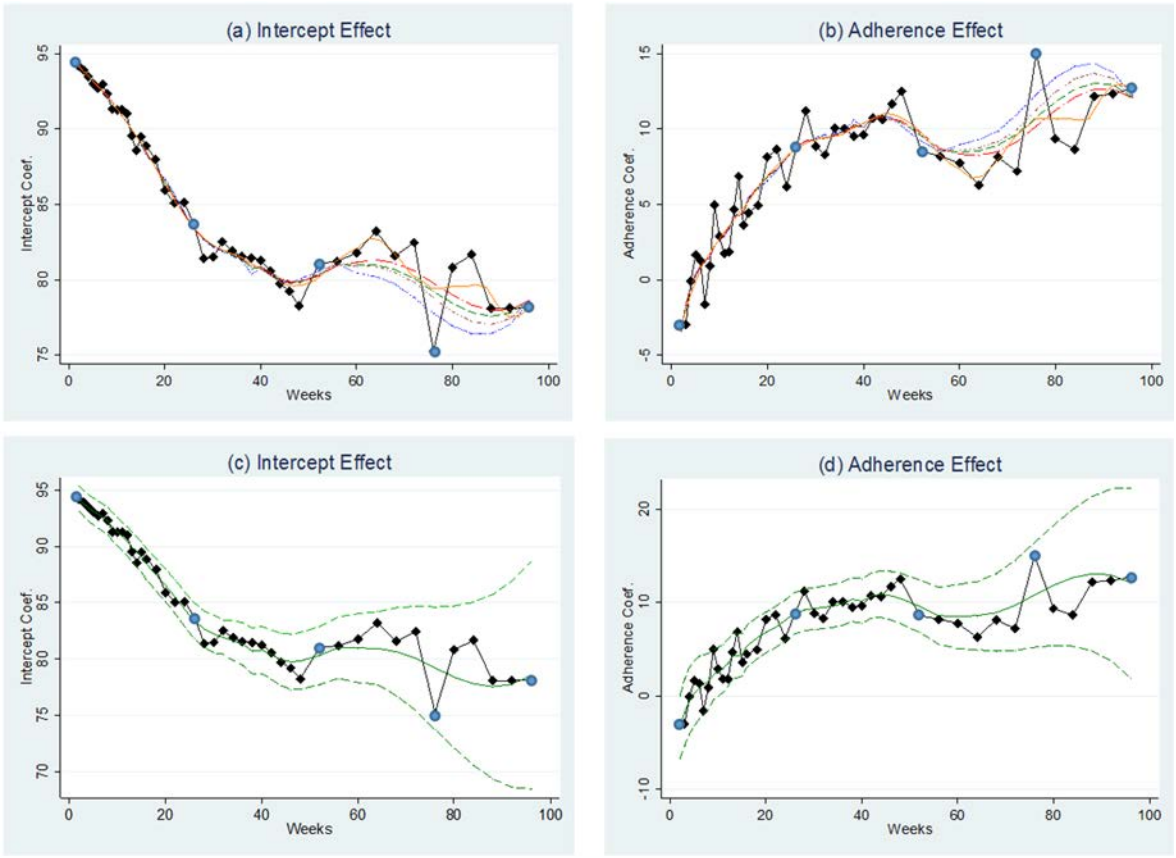

Figure 1. Superimposed smoothed coefficients using three-step estimation procedure

In (a) and (b) black diamonds, raw coefficients using pseudo data. Blue oval, raw coefficients using real measurements. Dashed line, estimates using analytical weight type 1 ($\frac{1}{\sqrt{D_k}}$). Tight dotted line, estimates using analytical weight type 2 ($\frac{1}{D_k}$). Long dash dotted line, estimates using

analytical weight type 3 ($\frac{1}{\sqrt{D_k}} + \frac{1}{\sqrt{se_k}}$). Short dash dotted line, estimates using analytical weight type 4 ($\frac{1}{D_k} + \frac{1}{se_k}$). Solid line, estimates using Epanechnikov kernel function. In (c) and (d) solid line, estimates using analytical weight type 1. Dashed line, estimates plus or minus 1 smoothed standard errors using analytical weight type 1.

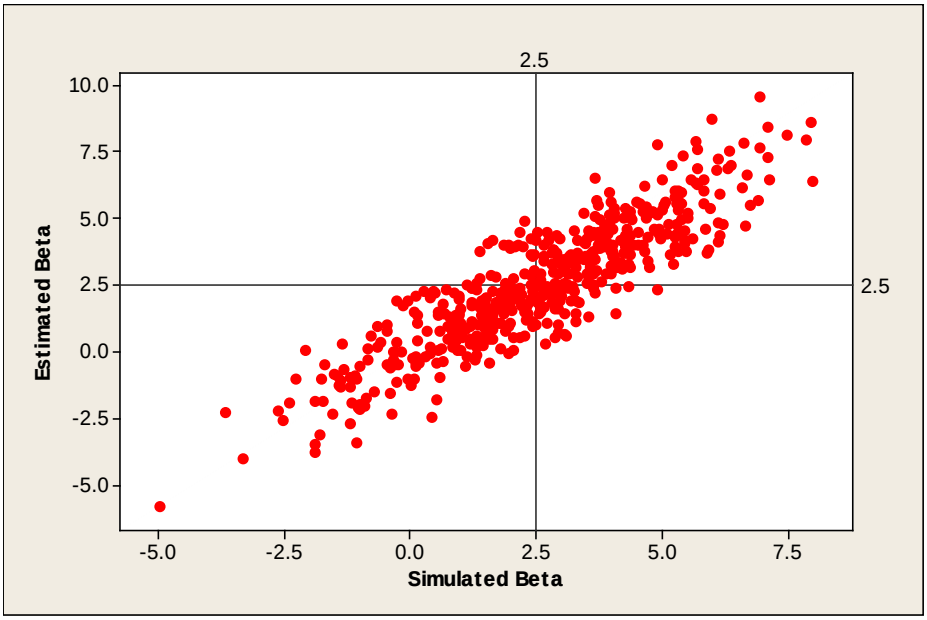

Figure 2. Simulation results using estimated variance-covariance matrix and analytical weight type 1

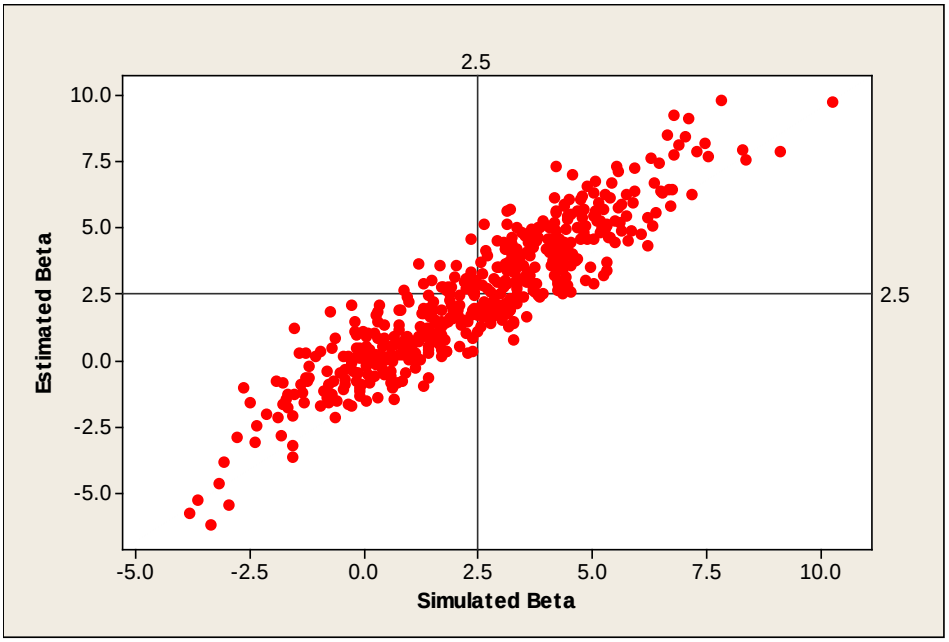

Figure 3. Simulation results using estimated variance-covariance matrix and analytical weight type 2

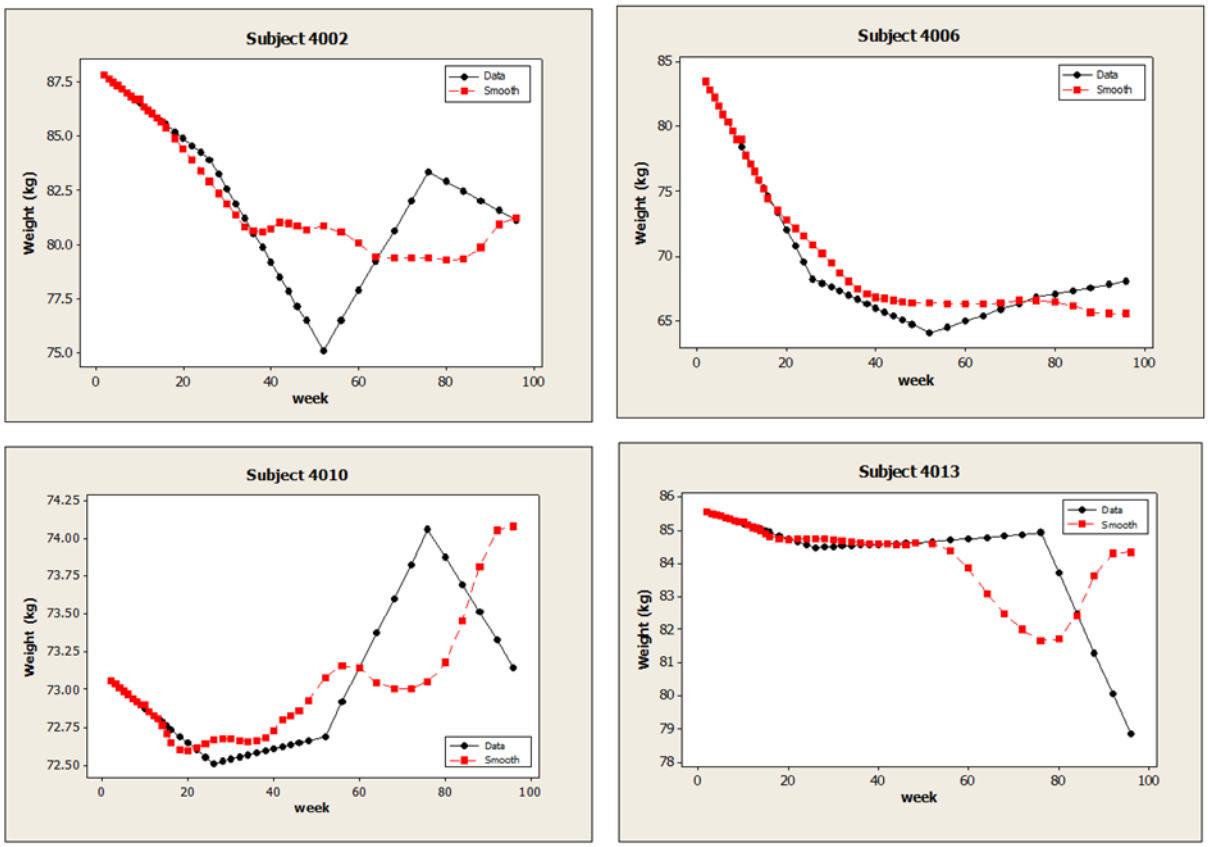

Figure 4. Data after imputation and smoothed line using analytical weights type 1

Table 1. Simulation results (Averaged Deviation) using analytical weight type 1, 2, 3 and 4

Averaged Deviation 1* Averaged Deviation 2** Averaged Deviation 3*** Averaged Deviation 4****	$\boldsymbol{\rho = 0.3}$	$\boldsymbol{\rho = 0.5}$	$\boldsymbol{\rho = 0.7}$
$\sigma^2 = \mathbf{5^2}$	0.18	-1.21	-2.27
	0.14	-0.65	-1.92
	-0.48	-1.27	-2.40
	-0.14	-0.88	-2.41
$\sigma^2 = \mathbf{10^2}$	11.13	7.08	3.42
	11.06	7.65	3.78
	11.57	7.67	2.70
	10.79	6.77	3.00
$\sigma^2 = \mathbf{15^2}$	30.93	22.31	10.63
	30.14	16.96	9.27
	32.91	20.09	10.91
	31.04	20.95	11.70
$\sigma^2 = \mathbf{20^2}$	55.05	42.35	20.16
	52.68	38.39	23.90
	61.17	44.37	22.77
	59.00	42.74	25.13
$\sigma^2 = \mathbf{25^2}$	91.87	63.29	38.57
	87.02	63.85	37.67
	91.44	61.67	36.47
	88.55	56.69	34.56
$\sigma^2 = \mathbf{30^2}$	132.10	98.36	53.40
	137.04	84.93	46.68
	140.47	98.70	62.38
	145.45	93.47	52.99
$\sigma^2 = \mathbf{35^2}$	191.54	123.05	67.91
	188.49	126.86	64.59
	173.83	123.67	77.23
	189.14	124.97	67.97
$\sigma^2 = \mathbf{40^2}$	237.17	171.99	106.66
	231.87	150.87	109.00
	238.47	180.84	100.38
	241.36	158.41	91.90

* Type 1: $\frac{1}{\sqrt{D_k}}$ ** Type 2: $\frac{1}{D_k}$ *** Type 3: $\frac{1}{\sqrt{D_k}} + \frac{1}{\sqrt{se_k}}$ **** Type 4: $\frac{1}{D_k} + \frac{1}{se_k}$

Table 2. Simulation results (Averaged Deviation) using analytical weight type 1 and 2

Averaged Deviation 1* / Averaged Deviation 2**	$\boldsymbol{\rho = 0.3}$	$\boldsymbol{\rho = 0.5}$	$\boldsymbol{\rho = 0.7}$
$\sigma^2 = \mathbf{5^2}$	0.09	0.07	0.04
	0.07	0.03	0.04
$\sigma^2 = \mathbf{10^2}$	0.35	0.20	0.14
	0.25	0.26	0.16
$\sigma^2 = \mathbf{15^2}$	0.80	0.55	0.28
	0.67	0.60	0.44
$\sigma^2 = \mathbf{20^2}$	1.18	0.91	0.27
	0.84	0.51	0.12
$\sigma^2 = \mathbf{25^2}$	1.84	1.71	0.66
	1.31	1.31	0.72
$\sigma^2 = \mathbf{30^2}$	2.12	1.83	1.29
	2.44	1.13	0.83
$\sigma^2 = \mathbf{35^2}$	3.11	2.87	1.60
	3.83	2.86	0.88
$\sigma^2 = \mathbf{40^2}$	5.42	4.48	3.00
	4.58	4.55	2.06

* Type 1: $\frac{1}{\sqrt{D_k}}$ ** Type 2: $\frac{1}{D_k}$

Table 3. Comparison between proposed method 1 and 2

Features	Method 1	Method 2
Outcome is measured less frequently than covariates	Yes	Yes
Outcome is measured more frequently than covariates	Yes	Yes
Some covariate is measured less frequently than outcome and other covariates	Yes	Yes
Repeated measures have large variances and small within-subjects correlations	No	No
Interaction among covariates	No	Yes
Less frequently measured variable has binomial, multinomial or Poisson distribution	No	No
Estimate individual profile	No	Yes
Estimate population function	Yes	Yes
Estimate time varying coefficients	Yes	No
Deal with missing at random data	No	Yes
Different analytical weights produce similar results	Yes	Yes
Results have straightforward interpretation	Yes	Yes
Short computation time	Yes	Yes

APPENDIX B: CODE

```
/*****Impute data *****/
data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 1\\example data before imputation.txt", header=T)
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
sampleY<-cbind(data[,2],data[,3],data[,4],data[,5],data[,6])
imputeY<-matrix(rep(1,210*43), 210, 43)
for(i in 1:210){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
```

```
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4), sampleY[i,5])
imputeY[i,]<-t(y)
}
/***** Smoothing for SMART example*****/
data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 1\\type2 impute raw betas.txt", header=T)
X<- data$x
Y<- data$y
x<- data[1:13,]$x
y<- data[1:13,]$y
w<- data[1:13,]$w
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- data[1:7,]$x
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
```

```
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- data[8:37,]$x
for(i in 1: 30){
x<- data[(i+1):(i+13),]$x
y<- data[(i+1):(i+13),]$y
w<- data[(i+1):(i+13),]$w
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat2[i]<-beta[1]
}
x<- data[31:43,]$x
y<- data[31:43,]$y
w<- data[31:43,]$w
Wmatrix<-diag(w)
one<-rep(1,length(x))
```

```
x0<- data[38:43,]$x
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
/*****Simulation1*****/
VARIANCE<-c(rep(5**2,3), rep(10**2,3), rep(15**2,3), rep(20**2,3), rep(25**2,3),
rep(30**2,3), rep(35**2,3), rep(40**2,3))
COVARIANCE<-
c(7.5,12.5,17.5,30,50,70,67.5,112.5,157.5,120,200,280,187.5,312.5,437.5,270,450,630,367.5,61
2.5,875.5,480,800,1120)
DIFF1<-c(rep(-1,24))
for(k in 1:24){
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
```

```
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
v<-matrix(rep(COVARIANCE[k],43*43), 43, 43)
for(i in 1:43){
v[i,i]<-VARIANCE[k]
}
E<-mvrnorm(200, rep(0,43), v)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
        }
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
```

```
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])

b1<- sampleY[i,1]-a1*t[1]

a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])

b2<- sampleY[i,2]-a2*t[20]

a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])

b3<- sampleY[i,3]-a3*t[32]

a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])

b4<- sampleY[i,4]-a4*t[38]

y1<-a1*t[2:19]+b1

y2<-a2*t[21:31]+b2

y3<-a3*t[33:37]+b3

y4<-a4*t[39:42]+b4

y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])

imputeY[i,]<-t(y)

}

rawbeta0<-rep(0,43)

rawbeta1<-rep(0,43)

for(i in 1:43){

one<-rep(1,200)

X<-cbind(one, X1[,i])

beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])

rawbeta0[i]<-beta[1]

rawbeta1[i]<-beta[2]
```

```
}
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
W<-1/sqrt(distance)
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
```

```
tempy<- rawbeta1[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
```

```
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
        yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
```

```
}
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff<-simuASEfinal- smoothASEfinal
DIFF1[k]<-diff
}
VARIANCE<-c(rep(5**2,3), rep(10**2,3), rep(15**2,3), rep(20**2,3), rep(25**2,3),
rep(30**2,3), rep(35**2,3), rep(40**2,3))
COVARIANCE<-
c(7.5,12.5,17.5,30,50,70,67.5,112.5,157.5,120,200,280,187.5,312.5,437.5,270,450,630,367.5,61
2.5,875.5,480,800,1120)
DIFF2<-c(rep(-1,24))
for(k in 1:24){
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
```

```
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
v<-matrix(rep(COVARIANCE[k],43*43), 43, 43)
for(i in 1:43){
v[i,i]<-VARIANCE[k]
}
E<-mvrnorm(200, rep(0,43), v)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
```

```
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])
imputeY[i,]<-t(y)
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
```

```
}
distance<-c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,
        11,13,13,11,9,7,5,1,5,9,13,9,5,1,5,9,9,5,1)
W<-1/(distance)
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
```

```
tempy<- rawbeta1[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
```

```
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
```

```
}
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff<-simuASEfinal- smoothASEfinal
DIFF2[k]<-diff
}
VARIANCE<-c(rep(5**2,3), rep(10**2,3), rep(15**2,3), rep(20**2,3), rep(25**2,3),
rep(30**2,3), rep(35**2,3), rep(40**2,3))
COVARIANCE<c(7.5,12.5,17.5,30,50,70,67.5,112.5,157.5,120,200,280,187.5,312.5,437.5,270,
450,630,367.5,612.5,875.5,480,800,1120)
DIFF3<-c(rep(-1,24))
for(k in 1:24){
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
```

```
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
v<-matrix(rep(COVARIANCE[k],43*43), 43, 43)
for(i in 1:43){
v[i,i]<-VARIANCE[k]
}
E<-mvrnorm(200, rep(0,43), v)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
```

```
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])
imputeY[i,]<-t(y)
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
rawbeta.se0<-rep(0,43)
rawbeta.se1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
```

```
rawbeta1[i]<-beta[2]
Y.hat <-X %*% beta
sigma.hat <- sqrt(sum((imputeY[i] - Y.hat)^2)/(200 - 2))
cov.beta <- sigma.hat^2 * solve(t(X) %*% X)
beta.se<-sqrt(diag(cov.beta))
rawbeta.se0[i]<-beta.se[1]
rawbeta.se1[i]<-beta.se[2]
}
distance<-c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,
11,13,13,11,9,7,5,1,5,9,13,9,5,1,5,9,9,5,1)
W1<-1/sqrt(distance)
W2<-t(1/sqrt(rawbeta.se1))
W<-W1+W2
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
```

```
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
```

```
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
```

```
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff<-simuASEfinal- smoothASEfinal
DIFF3[k]<-diff
        }
VARIANCE<-c(rep(5**2,3), rep(10**2,3), rep(15**2,3), rep(20**2,3), rep(25**2,3),
rep(30**2,3), rep(35**2,3), rep(40**2,3))
```

```
COVARIANCE<-
c(7.5,12.5,17.5,30,50,70,67.5,112.5,157.5,120,200,280,187.5,312.5,437.5,270,450,630,367.5,61
2.5,875.5,480,800,1120)
DIFF4<-c(rep(-1,24))
for(k in 1:24){
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
v<-matrix(rep(COVARIANCE[k],43*43), 43, 43)
for(i in 1:43){
v[i,i]<-VARIANCE[k]
}
E<-mvrnorm(200, rep(0,43), v)
```

```
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])
imputeY[i,]<-t(y)
```

```
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
rawbeta.se0<-rep(0,43)
rawbeta.se1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
Y.hat <-X %*% beta
sigma.hat <- sqrt(sum((imputeY[i] - Y.hat)^2)/(200 - 2))
cov.beta <- sigma.hat^2 * solve(t(X) %*% X)
beta.se<-sqrt(diag(cov.beta))
rawbeta.se0[i]<-beta.se[1]
rawbeta.se1[i]<-beta.se[2]
}
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
W1<-1/(distance)
W2<-t(1/(rawbeta.se1))
W<-W1+W2
```

```
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
```

```
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
```

```
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
```

```
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff<-simuASEfinal- smoothASEfinal
DIFF4[k]<-diff
}
estimatedV<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\estimated variance
matrix.txt", header=F)
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
```

```
E<-mvrnorm(200, rep(0,43), estimatedV)

Y<-matrix(rep(1,200*43), 200, 43)

        for(i in 1:43){

y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]

Y[,i]<-y

        }

sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])

imputeY<-matrix(rep(1,200*43), 200, 43)

for(i in 1:200){

a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])

b1<- sampleY[i,1]-a1*t[1]

a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])

b2<- sampleY[i,2]-a2*t[20]

a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])

b3<- sampleY[i,3]-a3*t[32]

a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])

b4<- sampleY[i,4]-a4*t[38]

y1<-a1*t[2:19]+b1

y2<-a2*t[21:31]+b2

y3<-a3*t[33:37]+b3

y4<-a4*t[39:42]+b4

y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])

imputeY[i,]<-t(y)
```

```
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
        }
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
W<-1/sqrt(distance)
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
```

```
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
```

```
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
```

```
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff1<-simuASEfinal- smoothASEfinal
estimatedV<-read.table("C:\\Users\\Lei\\Desktop\\Lei  Ye  dissertation\\data\\estimated  variance
matrix.txt", header=F)
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
```

```
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
}
E<-mvrnorm(200, rep(0,43), estimatedV)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
```

```
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])
imputeY[i,]<-t(y)
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
}
```

```
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
W<-1/(distance)
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
```

```
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
```

```
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
```

```
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
}
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff2<-simuASEfinal- smoothASEfinal
estimatedV<-read.table("C:\\Users\\Lei\\Desktop\\Lei  Ye  dissertation\\data\\estimated  variance
matrix.txt", header=F)
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
```

```
}
E<-mvrnorm(200, rep(0,43), estimatedV)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
```

```
y<cbind(sampleY[i,1],t(y1),sampleY[i,2],t(y2),sampleY[i,3],t(y3),sampleY[i,4],t(y4),sampleY[i,
5])
imputeY[i,]<-t(y)
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
rawbeta.se0<-rep(0,43)
rawbeta.se1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
Y.hat <-X %*% beta
sigma.hat <- sqrt(sum((imputeY[i] - Y.hat)^2)/(200 - 2))
cov.beta <- sigma.hat^2 * solve(t(X) %*% X)
beta.se<-sqrt(diag(cov.beta))
rawbeta.se0[i]<-beta.se[1]
rawbeta.se1[i]<-beta.se[2]
}
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
```

```
W1<-1/sqrt(distance)
W2<-t(1/sqrt(rawbeta.se1))
W<-W1+W2
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
```

```
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
```

```
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
```

```
simuASEone<- matrix(rep(0,43), 43,1)
smoothASEone<- matrix(rep(0,43), 43,1)
for(i in 1:5000){
simuASEone <- simuASEone +simuASE[,i]
smoothASEone<- smoothASEone+ smoothASE[,i]
        }
simuASEfinal<- (sum(simuASEone))/5000
smoothASEfinal<- (sum(smoothASEone))/5000
diff3<-simuASEfinal- smoothASEfinal
estimatedV<-read.table("C:\\Users\\Lei\\Desktop\\Lei  Ye  dissertation\\data\\estimated  variance
matrix.txt", header=F)
smooth<- matrix(rep(1,43*5000), 43,5000)
simulation<- matrix(rep(1,43*5000), 43,5000)
for(j in 1: 5000){
t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))
B0<-90.5-0.2*t
B1<- -1.1+0.4*t-0.003*t**2
X0<-matrix(rep(1,200*43), 200, 43)
X1<-matrix(rep(2,200*43), 200, 43)
P<-0.5
x<- rbinom(200,1, P)
for(i in 1: 43){
X1[,i]<-x
```

```
}
E<-mvrnorm(200, rep(0,43), estimatedV)
Y<-matrix(rep(1,200*43), 200, 43)
for(i in 1:43){
y<-B0[i]*X0[,i]+B1[i]*X1[,i]+E[,i]
Y[,i]<-y
}
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])
imputeY<-matrix(rep(1,200*43), 200, 43)
for(i in 1:200){
a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])
b1<- sampleY[i,1]-a1*t[1]
a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])
b2<- sampleY[i,2]-a2*t[20]
a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])
b3<- sampleY[i,3]-a3*t[32]
a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])
b4<- sampleY[i,4]-a4*t[38]
y1<-a1*t[2:19]+b1
y2<-a2*t[21:31]+b2
y3<-a3*t[33:37]+b3
y4<-a4*t[39:42]+b4
```

```
y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])
imputeY[i,]<-t(y)
}
rawbeta0<-rep(0,43)
rawbeta1<-rep(0,43)
rawbeta.se0<-rep(0,43)
rawbeta.se1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*%imputeY[,i])
rawbeta0[i]<-beta[1]
rawbeta1[i]<-beta[2]
Y.hat <-X %*% beta
sigma.hat <- sqrt(sum((imputeY[i] - Y.hat)^2)/(200 - 2))
cov.beta <- sigma.hat^2 * solve(t(X) %*% X)
beta.se<-sqrt(diag(cov.beta))
rawbeta.se0[i]<-beta.se[1]
rawbeta.se1[i]<-beta.se[2]
}
distance<c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,11,13,13,11,9,7,5,1,5,9,13,9,5,1,
5,9,9,5,1)
```

```
W1<-1/(distance)
W2<-t(1/(rawbeta.se1))
W<-W1+W2
tempx<- t[1:13]
tempy<- rawbeta1[1:13]
w<- W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- tempx[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- t[8:37]
for(i in 1: 30){
tempx<- t[(i+1):(i+13)]
tempy<- rawbeta1[(i+1):(i+13)]
```

```
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
yhat2[i]<-beta[1]
}
tempx<- t[31:43]
tempy<- rawbeta1[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(tempx))
x0<- t[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(tempx-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%tempy)
```

```
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
simubeta0<-rep(0,43)
simubeta1<-rep(0,43)
for(i in 1:43){
one<-rep(1,200)
X<-cbind(one, X1[,i])
beta<-solve(t(X)%*%X)%*%(t(X)%*% Y[,i])
simubeta0[i]<-beta[1]
simubeta1[i]<-beta[2]
}
smooth[,j]<-yhat
simulation[,j]<-simubeta1
}
simuASE<- matrix(rep(1,43*5000), 43,5000)
smoothASE<- matrix(rep(1,43*5000), 43,5000)
for(i in 1:5000){
simuASEtemp<- (simulation[,i]-B1)**2
smoothASEtemp<- (smooth[,i]-B1)**2
simuASE[,i]<-simuASEtemp
smoothASE[,i]<-smoothASEtemp
}
```

```
simuASEone<- matrix(rep(0,43), 43,1)

smoothASEone<- matrix(rep(0,43), 43,1)

for(i in 1:5000){

simuASEone <- simuASEone +simuASE[,i]

smoothASEone<- smoothASEone+ smoothASE[,i]

}

simuASEfinal<- (sum(simuASEone))/5000

smoothASEfinal<- (sum(smoothASEone))/5000

diff4<-simuASEfinal- smoothASEfinal

/*****Smooth individual data*****/

weight_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\for smooth with no missing.txt", header=T)

time_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\analytical weight and time line.txt", header=T)

smoothed<- matrix(rep(0,43*173), 173,43)

for(j in 1:173){

Y<-as.vector(t( weight_data[j,2:44]))

X<- time_data$week

W<- time_data$weight2

x<-X[1:13]

y<- Y[1:13]

w<-W[1:13]

Wmatrix<-diag(w)

one<-rep(1,length(x))

x0<- X[1:7]

yhat1<-rep(0,length(x0))

for(i in 1: length(x0)){
```

```
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- X[8:37]
for(i in 1: 30){
x<- X[(i+1):(i+13)]
y<- Y[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat2[i]<-beta[1]
}
x<- X[31:43]
y<- Y[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[38:43]
```

```
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
smoothed[j,]<-yhat
}
write.csv(smoothed, "smoothed 1to43.csv")
weight_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\for smooth with  missing1to20.txt", header=T)
time_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\analytical weight and time line.txt", header=T)
time_data<-time_data[1:20,]
smoothed<- matrix(rep(0,20*13), 13,20)
for(j in 1:13){
Y<-as.vector(t( weight_data[j,2:21]))
X<- time_data$week
W<- time_data$weight1
x<-X[1:9]
y<- Y[1:9]
w<-W[1:9]
Wmatrix<-diag(w)
one<-rep(1,length(x))
```

```
x0<- X[1:5]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,11)
x0<- X[6:16]
for(i in 1: 11){
x<- X[(i+1):(i+9)]
y<- Y[(i+1):(i+9)]
w<- W[(i+1):(i+9)]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat2[i]<-beta[1]
}
x<- X[12:20]
y<- Y[12:20]
w<- W[12:20]
```

```
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[17:20]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
smoothed[j,]<-yhat
}
write.csv(smoothed, "smoothed 1to20.csv")
weight_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\for smooth with missing1to32.txt", header=T)
time_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\analytical weight and time line.txt", header=T)
time_data<-time_data[1:32,]
smoothed<- matrix(rep(0,32*4), 4,32)
for(j in 1:4){
Y<-as.vector(t( weight_data[j,2:33]))
X<- time_data$week
W<- time_data$weight1
x<-X[1:13]
y<- Y[1:13]
```

```
w<-W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,19)
x0<- X[8:26]
for(i in 1: 19){
x<- X[(i+1):(i+13)]
y<- Y[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat2[i]<-beta[1]
}
```

```
x<- X[27:32]
y<- Y[27:32]
w<- W[27:32]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[27:32]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
smoothed[j,]<-yhat
}
write.csv(smoothed, "smoothed 1to32.csv")
weight_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\for smooth with missing1to38.txt", header=T)
time_data<-read.table("C:\\Users\\Lei\\Desktop\\Lei Ye dissertation\\data\\paper 3\\analytical weight and time line.txt", header=T)
time_data<-time_data[1:38,]
smoothed<- matrix(rep(0,38*2), 2,38)
for(j in 1:2){
Y<-as.vector(t( weight_data[j,2:39]))
X<- time_data$week
```

```
W<- time_data$weight1
x<-X[1:13]
y<- Y[1:13]
w<-W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,25)
x0<- X[8:32]
for(i in 1: 25){
x<- X[(i+1):(i+13)]
y<- Y[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
```

```
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat2[i]<-beta[1]
}
x<- X[33:38]
y<- Y[33:38]
w<- W[33:38]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- X[27:32]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
smoothed[j,]<-yhat
}
write.csv(smoothed, "smoothed 1to38.csv")

/*****Simulation2*****/

VARIANCE<-c(rep(5**2,3), rep(10**2,3), rep(15**2,3), rep(20**2,3), rep(25**2,3),
rep(30**2,3), rep(35**2,3), rep(40**2,3))
```

```
COVARIANCE<-
c(7.5,12.5,17.5,30,50,70,67.5,112.5,157.5,120,200,280,187.5,312.5,437.5,270,450,630,367.5,61
2.5,875.5,480,800,1120)

fitbeta<-rep(1,500)

simubeta<-rep(1,500)

for(k in 1: 500){

t<-c(seq(2,16),seq(18,48,2),seq(52,96,4))

B0<-90.5-0.2*t

B1<- 2.5

X0<-matrix(rep(1,200*43), 200, 43)

X1<-matrix(rep(2,200*43), 200, 43)

P<-0.5

x<- rbinom(200,1, P)

for(i in 1: 43){

X1[,i]<-x

}

v<-matrix(rep(COVARIANCE[20],43*43), 43, 43)

for(i in 1:43){

v[i,i]<-VARIANCE[20]

}

E<-mvrnorm(200, rep(0,43), v)

Y<-matrix(rep(1,200*43), 200, 43)

for(i in 1:43){

y<-B0[i]*X0[,i]+B1*X1[,i]+E[,i]

Y[,i]<-y

}

SimuY<-Y
```

```
sampleY<-cbind(Y[,1],Y[,20],Y[,32],Y[,38],Y[,43])

imputeY<-matrix(rep(1,200*43), 200, 43)

for(i in 1:200){

a1<-(sampleY[i,1]- sampleY[i,2])/(t[1]-t[20])

b1<- sampleY[i,1]-a1*t[1]

a2<-(sampleY[i,2]- sampleY[i,3])/(t[20]-t[32])

b2<- sampleY[i,2]-a2*t[20]

a3<-(sampleY[i,3]- sampleY[i,4])/(t[32]-t[38])

b3<- sampleY[i,3]-a3*t[32]

a4<-(sampleY[i,4]- sampleY[i,5])/(t[38]-t[43])

b4<- sampleY[i,4]-a4*t[38]

y1<-a1*t[2:19]+b1

y2<-a2*t[21:31]+b2

y3<-a3*t[33:37]+b3

y4<-a4*t[39:42]+b4

y<-cbind(sampleY[i,1],t(y1), sampleY[i,2],t(y2), sampleY[i,3],t(y3), sampleY[i,4],t(y4),
sampleY[i,5])

imputeY[i,]<-t(y)

}

distance<-c(1,2,3,4,5,6,7,8,9,10,15,14,13,12,11,9,7,5,3,1,3,5,7,9,

11,13,13,11,9,7,5,1,5,9,13,9,5,1,5,9,9,5,1)

W<-1/sqrt(distance)

smoothedY<- matrix(rep(0,43*200), 200,43)

for(j in 1:200){

tempY<-imputeY[j,]

tempX<- t
```

```
x<-tempX[1:13]
y<- tempY[1:13]
w<-W[1:13]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- tempX[1:7]
yhat1<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat1[i]<-beta[1]
}
yhat2<-rep(0,30)
x0<- tempX[8:37]
for(i in 1: 30){
x<- tempX[(i+1):(i+13)]
y<- tempY[(i+1):(i+13)]
w<- W[(i+1):(i+13)]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
```

```
yhat2[i]<-beta[1]
}
x<- tempX[31:43]
y<- tempY[31:43]
w<- W[31:43]
Wmatrix<-diag(w)
one<-rep(1,length(x))
x0<- tempX[38:43]
yhat3<-rep(0,length(x0))
for(i in 1: length(x0)){
x1<-(x-x0[i])
x2<-x1**2
x3<-x1**3
Xmatrix<-cbind(one, x1, x2, x3)
beta<-solve(t(Xmatrix)%*%Wmatrix%*%Xmatrix)%*%(t(Xmatrix)%*%Wmatrix%*%y)
yhat3[i]<-beta[1]
}
yhat<-c(yhat1, yhat2, yhat3)
yhat
smoothedY[j,]<-yhat
}
subj<-rep(1:200,each=43)
week<-rep(t,200)
Tweight<- rep(0,43*200)
for(n in 0:199){
for(m in 1:43){
Tweight[m+43*n]<-SimuY[n+1,m]
}}
```

```
weight<- rep(0,43*200)
for(n in 0:199){
for(m in 1:43){
weight[m+43*n]<-smoothedY[n+1,m]
}}
adhere<- rep(0,43*200)
for(n in 0:199){
for(m in 1:43){
adhere[m+43*n]<-X1[n+1,m]
}}
true<-data.frame(subj,week,Tweight,adhere)
new<-data.frame(subj,week,weight,adhere)
Tfit<-lme (Tweight ~ week+adhere, random = ~ 1|subj, data =true)
fit<-lme (weight ~ week+adhere, random = ~ 1|subj, data =new)
Tbeta<-coef(Tfit)
Tbeta<-Tbeta$adhere
beta<-coef(fit)
beta<-beta$adhere
fitbeta[k]<-beta[1]
simubeta[k]<-Tbeta[1]
}
write.csv(fitbeta, file = "S.csv")
write.csv(simubeta, file = "T.csv")
```

/*****Future direction*****/

Proc import datafile = "C:\Users\Lei\Desktop\Lei Ye issertation\data\paper 2\imputed y with missing.sav" out= work.temp;

run;
%include "C:\Users\Lei\Desktop\Lei Ye dissertation\code\paper 2\wide to long.txt";

%***tolong***(temp,long,ID,TIME,**1**,**43**,sm wt);

data long;
set long;

if time=**1** then week=**2**; if time=**1** then distance=**1**;
if time=**2** then week=**3**; if time=**2** then distance=**2**;
if time=**3** then week=**4**; if time=**3** then distance=**3**;
if time=**4** then week=**5**; if time=**4** then distance=**4**;
if time=**5** then week=**6**; if time=**5** then distance=**5**;

if time=**6** then week=**7**; if time=**6** then distance=**6**;
if time=**7** then week=**8**; if time=**7** then distance=**7**;
if time=**8** then week=**9**; if time=**8** then distance=**8**;
if time=**9** then week=**10**; if time=**9** then distance=**9**;
if time=**10** then week=**11**; if time=**10** then distance=**10**;

if time=**11** then week=**12**; if time=**11** then distance=**15**;
if time=**12** then week=**13**; if time=**12** then distance=**14**;
if time=**13** then week=**14**; if time=**13** then distance=**13**;
if time=**14** then week=**15**; if time=**14** then distance=**12**;
if time=**15** then week=**16**; if time=**15** then distance=**11**;

if time=**16** then week=**18**; if time=**16** then distance=**9**;
if time=**17** then week=**20**; if time=**17** then distance=**7**;
if time=**18** then week=**22**; if time=**18** then distance=**5**;
if time=**19** then week=**24**; if time=**19** then distance=**3**;
if time=**20** then week=**26**; if time=**20** then distance=**1**;

if time=**21** then week=**28**; if time=**21** then distance=**3**;
if time=**22** then week=**30**; if time=**22** then distance=**5**;
if time=**23** then week=**32**; if time=**23** then distance=**7**;
if time=**24** then week=**34**; if time=**24** then distance=**9**;
if time=**25** then week=**36**; if time=**25** then distance=**11**;

if time=**26** then week=**38**; if time=**26** then distance=**13**;
if time=**27** then week=**40**; if time=**27** then distance=**13**;
if time=**28** then week=**42**; if time=**28** then distance=**11**;
if time=**29** then week=**44**; if time=**29** then distance=**9**;

```
if time=30 then week=46; if time=30 then distance=7;

if time=31 then week=48; if time=31 then distance=5;
if time=32 then week=52; if time=32 then distance=1;
if time=33 then week=56; if time=33 then distance=5;
if time=34 then week=60; if time=34 then distance=9;
if time=35 then week=64; if time=35 then distance=13;

if time=36 then week=68; if time=36 then distance=9;
if time=37 then week=72; if time=37 then distance=5;
if time=38 then week=76; if time=38 then distance=1;
if time=39 then week=80; if time=39 then distance=5;
if time=40 then week=84; if time=40 then distance=9;

if time=41 then week=88; if time=41 then distance=9;
if time=42 then week=92; if time=42 then distance=5;
if time=43 then week=96; if time=43 then distance=1;
run;

data long;
set long;

analyze_w1=1/(sqrt(distance));
analyze_w2=1/distance;

run;

/************ 1  ************/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-2)*wt;
sample_x1=(sqrt(analyze_w1))*(week-2);
sample_x2=(sqrt(analyze_w1))*(week-2)*(week-2);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
```

```
run;

proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;

data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt1;
set final;
wt1=sscoeff;
keep ID wt1;
run;
/************ 2************/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-3)*wt;
sample_x1=(sqrt(analyze_w1))*(week-3);
sample_x2=(sqrt(analyze_w1))*(week-3)*(week-3);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
```

```
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt2;
set final;
wt2=sscoeff;
keep ID wt2;
run;
/************ 3 ***********/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-4)*wt;
sample_x1=(sqrt(analyze_w1))*(week-4);
sample_x2=(sqrt(analyze_w1))*(week-4)*(week-4);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
```

```
proc sort data=final;
by effect;
run;
data wt3;
set final;
wt3=sscoeff;
keep ID wt3;
run;
/************ 4************/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-5)*wt;
sample_x1=(sqrt(analyze_w1))*(week-5);
sample_x2=(sqrt(analyze_w1))*(week-5)*(week-5);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt4;
set final;
wt4=sscoeff;
keep ID wt4;
```

```
run;
/************ 5 ***********/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-6)*wt;
sample_x1=(sqrt(analyze_w1))*(week-6);
sample_x2=(sqrt(analyze_w1))*(week-6)*(week-6);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt5;
set final;
wt5=sscoeff;
keep ID wt5;
run;
/************ 6 ***********/
data sample;
set long;
if time>13 then delete;
run;
data sample;
```

```
set sample;
sample_y=(sqrt(analyze_w1))*(week-7)*wt;
sample_x1=(sqrt(analyze_w1))*(week-7);
sample_x2=(sqrt(analyze_w1))*(week-7)*(week-7);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt6;
set final;
wt6=sscoeff;
keep ID wt6;
run;
/************ 7 ***********/
data sample;
set long;
if time>13 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-8)*wt;
sample_x1=(sqrt(analyze_w1))*(week-8);
sample_x2=(sqrt(analyze_w1))*(week-8)*(week-8);
run;
proc mixed data=sample ;
class ID;
```

```
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt7;
set final;
wt7=sscoeff;
keep ID wt7;
run;
/********** 8 **********/
data sample;
set long;
if time<2 then delete;
if time>14 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-9)*wt;
sample_x1=(sqrt(analyze_w1))*(week-9);
sample_x2=(sqrt(analyze_w1))*(week-9)*(week-9);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
```

```
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt8;
set final;
wt8=sscoeff;
keep ID wt8;
run;
/********** 9**********/
data sample;
set long;
if time<3 then delete;
if time>15 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-10)*wt;
sample_x1=(sqrt(analyze_w1))*(week-10);
sample_x2=(sqrt(analyze_w1))*(week-10)*(week-10);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
```

```
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt9;
set final;
wt9=sscoeff;
keep ID wt9;
run;
/************ 10***********/
data sample;
set long;
if time<4 then delete;
if time>16 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-11)*wt;
sample_x1=(sqrt(analyze_w1))*(week-11);
sample_x2=(sqrt(analyze_w1))*(week-11)*(week-11);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
```

```
proc sort data=final;
by effect;
run;
data wt10;
set final;
wt10=sscoeff;
keep ID wt10;
run;
/************ 11***********/
data sample;
set long;
if time<5 then delete;
if time>17 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-12)*wt;
sample_x1=(sqrt(analyze_w1))*(week-12);
sample_x2=(sqrt(analyze_w1))*(week-12)*(week-12);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt11;
set final;
wt11=sscoeff;
```

```
keep ID wt11;
run;
/************ 12***********/
data sample;
set long;
if time<6 then delete;
if time>18 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-13)*wt;
sample_x1=(sqrt(analyze_w1))*(week-13);
sample_x2=(sqrt(analyze_w1))*(week-13)*(week-13);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt12;
set final;
wt12=sscoeff;
keep ID wt12;
run;
/************ 13***********/
data sample;
set long;
if time<7 then delete;
```

```
if time>19 then delete;
run;
data sample;
set sample
sample_y=(sqrt(analyze_w1))*(week-14)*wt;
sample_x1=(sqrt(analyze_w1))*(week-14);
sample_x2=(sqrt(analyze_w1))*(week-14)*(week-14);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt13;
set final;
wt13=sscoeff;
keep ID wt13;
run;

/************ 14************/
data sample;
set long;
if time<8 then delete;
if time>20 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-15)*wt;
```

```
sample_x1=(sqrt(analyze_w1))*(week-15);
sample_x2=(sqrt(analyze_w1))*(week-15)*(week-15);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt14;
set final;
wt14=sscoeff;
keep ID wt14;
run;
/************ 15***********/
data sample;
set long;
if time<9 then delete;
if time>21 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-16)*wt;
sample_x1=(sqrt(analyze_w1))*(week-16);
sample_x2=(sqrt(analyze_w1))*(week-16)*(week-16);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
```

```
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt15;
set final;
wt15=sscoeff;
keep ID wt15;
run;
/************ 16***********/
data sample;
set long;
if time<10 then delete;
if time>22 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-18)*wt;
sample_x1=(sqrt(analyze_w1))*(week-18);
sample_x2=(sqrt(analyze_w1))*(week-18)*(week-18);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
```

```
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt16;
set final;
wt16=sscoeff;
keep ID wt16;
run;
/************ 17************/
data sample;
set long;
if time<11 then delete;
if time>23 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-20)*wt;
sample_x1=(sqrt(analyze_w1))*(week-20);
sample_x2=(sqrt(analyze_w1))*(week-20)*(week-20);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
```

```
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt17;
set final;
wt17=sscoeff;
keep ID wt17;
run;
/************ 18***********/
data sample;
set long;
if time<12 then delete;
if time>24 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-22)*wt;
sample_x1=(sqrt(analyze_w1))*(week-22);
sample_x2=(sqrt(analyze_w1))*(week-22)*(week-22);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
```

```
by effect;
run;
data wt18;
set final;
wt18=sscoeff;
keep ID wt18;
run;
/************ 19***********/
data sample;
set long;
if time<13 then delete;
if time>25 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-24)*wt;
sample_x1=(sqrt(analyze_w1))*(week-24);
sample_x2=(sqrt(analyze_w1))*(week-24)*(week-24);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt19;
set final;
wt19=sscoeff;
keep ID wt19;
```

```
run;
/************ 20***********/
data sample;
set long;
if time<14 then delete;
if time>26 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-26)*wt;
sample_x1=(sqrt(analyze_w1))*(week-26);
sample_x2=(sqrt(analyze_w1))*(week-26)*(week-26);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt20;
set final;
wt20=sscoeff;
keep ID wt20;
run;
/************ 21***********/
data sample;
set long;
if time<15 then delete;
if time>27 then delete;
```

```
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-28)*wt;
sample_x1=(sqrt(analyze_w1))*(week-28);
sample_x2=(sqrt(analyze_w1))*(week-28)*(week-28);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt21;
set final;
wt21=sscoeff;
keep ID wt21;
run;
/************ 22************/
data sample;
set long;
if time<16 then delete;
if time>28 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-30)*wt;
sample_x1=(sqrt(analyze_w1))*(week-30);
sample_x2=(sqrt(analyze_w1))*(week-30)*(week-30);
```

```
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt22;
set final;
wt22=sscoeff;
keep ID wt22;
run;
/************ 23************/
data sample;
set long;
if time<17 then delete;
if time>29 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-32)*wt;
sample_x1=(sqrt(analyze_w1))*(week-32);
sample_x2=(sqrt(analyze_w1))*(week-32)*(week-32);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
```

```
                        rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt23;
set final;
wt23=sscoeff;
keep ID wt23;
run;
/************ 24************/
data sample;
set long;
if time<18 then delete;
if time>30 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-34)*wt;
sample_x1=(sqrt(analyze_w1))*(week-34);
sample_x2=(sqrt(analyze_w1))*(week-34)*(week-34);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                        rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
```

```
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt24;
set final;
wt24=sscoeff;
keep ID wt24;
run;
/************ 25***********/
data sample;
set long;
if time<19 then delete;
if time>31 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-36)*wt;
sample_x1=(sqrt(analyze_w1))*(week-36);
sample_x2=(sqrt(analyze_w1))*(week-36)*(week-36);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
```

```
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt25;
set final;
wt25=sscoeff;
keep ID wt25;
run;
/************ 26***********/
data sample;
set long;
if time<20 then delete;
if time>32 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-38)*wt;
sample_x1=(sqrt(analyze_w1))*(week-38);
sample_x2=(sqrt(analyze_w1))*(week-38)*(week-38);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
```

```
data wt26;
set final;
wt26=sscoeff;
keep ID wt26;
run;
/************ 27***********/
data sample;
set long;
if time<21 then delete;
if time>33 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-40)*wt;
sample_x1=(sqrt(analyze_w1))*(week-40);
sample_x2=(sqrt(analyze_w1))*(week-40)*(week-40);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt27;
set final;
wt27=sscoeff;
keep ID wt27;
run;
/************ 28***********/
```

```
data sample;
set long;
if time<22 then delete;
if time>34 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-42)*wt;
sample_x1=(sqrt(analyze_w1))*(week-42);
sample_x2=(sqrt(analyze_w1))*(week-42)*(week-42);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt28;
set final;
wt28=sscoeff;
keep ID wt28;
run;
/********** 29**********/
data sample;
set long;
if time<23 then delete;
if time>35 then delete;
run;
data sample;
```

```
set sample;
sample_y=(sqrt(analyze_w1))*(week-44)*wt;
sample_x1=(sqrt(analyze_w1))*(week-44);
sample_x2=(sqrt(analyze_w1))*(week-44)*(week-44);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt29;
set final;
wt29=sscoeff;
keep ID wt29;
run;
/************ 30***********/
data sample;
set long;
if time<24 then delete;
if time>36 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-46)*wt;
sample_x1=(sqrt(analyze_w1))*(week-46);
sample_x2=(sqrt(analyze_w1))*(week-46)*(week-46);
run;
proc mixed data=sample ;
```

```
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt30;
set final;
wt30=sscoeff;
keep ID wt30;
run;
/************ 31***********/
data sample;
set long;
if time<25 then delete;
if time>37 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-48)*wt;
sample_x1=(sqrt(analyze_w1))*(week-48);
sample_x2=(sqrt(analyze_w1))*(week-48)*(week-48);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
```

```
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt31;
set final;
wt31=sscoeff;
keep ID wt31;
run;
/************ 32************/
data sample;
set long;
if time<26 then delete;
if time>38 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-52)*wt;
sample_x1=(sqrt(analyze_w1))*(week-52);
sample_x2=(sqrt(analyze_w1))*(week-52)*(week-52);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                        rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
```

```
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt32;
set final;
wt32=sscoeff;
keep ID wt32;
run;
/************ 33***********/
data sample;
set long;
if time<27 then delete;
if time>39 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-56)*wt;
sample_x1=(sqrt(analyze_w1))*(week-56);
sample_x2=(sqrt(analyze_w1))*(week-56)*(week-56);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
```

```
run;
proc sort data=final;
by effect;
run;
data wt33;
set final;
wt33=sscoeff;
keep ID wt33;
run;
/************ 34***********/
data sample;
set long;
if time<28 then delete;
if time>40 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-60)*wt;
sample_x1=(sqrt(analyze_w1))*(week-60);
sample_x2=(sqrt(analyze_w1))*(week-60)*(week-60);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt34;
set final;
```

```
wt34=sscoeff;
keep ID wt34;
run;

/************ 35***********/
data sample;
set long;
if time<29 then delete;
if time>41 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-64)*wt;
sample_x1=(sqrt(analyze_w1))*(week-64);
sample_x2=(sqrt(analyze_w1))*(week-64)*(week-64);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt35;
set final;
wt35=sscoeff;
keep ID wt35;
run;

/************ 36***********/
```

```
data sample;
set long;
if time<30 then delete;
if time>42 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-68)*wt;
sample_x1=(sqrt(analyze_w1))*(week-68);
sample_x2=(sqrt(analyze_w1))*(week-68)*(week-68);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt36;
set final;
wt36=sscoeff;
keep ID wt36;
run;
/********** 37**********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
```

```
set sample;
sample_y=(sqrt(analyze_w1))*(week-72)*wt;
sample_x1=(sqrt(analyze_w1))*(week-72);
sample_x2=(sqrt(analyze_w1))*(week-72)*(week-72);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt37;
set final;
wt37=sscoeff;
keep ID wt37;
run;
/************ 38***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-76)*wt;
sample_x1=(sqrt(analyze_w1))*(week-76);
sample_x2=(sqrt(analyze_w1))*(week-76)*(week-76);
run;
proc mixed data=sample ;
```

```
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt38;
set final;
wt38=sscoeff;
keep ID wt38;
run;
/************ 39***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-80)*wt;
sample_x1=(sqrt(analyze_w1))*(week-80);
sample_x2=(sqrt(analyze_w1))*(week-80)*(week-80);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
```

```
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt39;
set final;
wt39=sscoeff;
keep ID wt39;
run;
/************ 40***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-84)*wt;
sample_x1=(sqrt(analyze_w1))*(week-84);
sample_x2=(sqrt(analyze_w1))*(week-84)*(week-84);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                        rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                        rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
```

```
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt40;
set final;
wt40=sscoeff;
keep ID wt40;
run;
/************ 41***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-88)*wt;
sample_x1=(sqrt(analyze_w1))*(week-88);
sample_x2=(sqrt(analyze_w1))*(week-88)*(week-88);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
```

```
run;
proc sort data=final;
by effect;
run;
data wt41;
set final;
wt41=sscoeff;
keep ID wt41;
run;
/************ 42***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-92)*wt;
sample_x1=(sqrt(analyze_w1))*(week-92);
sample_x2=(sqrt(analyze_w1))*(week-92)*(week-92);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt42;
set final;
```

```
wt42=sscoeff;
keep ID wt42;
run;
/************ 43***********/
data sample;
set long;
if time<31 then delete;
if time>43 then delete;
run;
data sample;
set sample;
sample_y=(sqrt(analyze_w1))*(week-93)*wt;
sample_x1=(sqrt(analyze_w1))*(week-93);
sample_x2=(sqrt(analyze_w1))*(week-93)*(week-93);
run;
proc mixed data=sample ;
class ID;
model sample_y= sample_x1 sample_x2/ddfm=kr s;
random sample_x1/type=un subject=ID  s;
ods output solutionf=sf1(keep=id effect estimate
                         rename=(estimate=overall));
ods output solutionr=sr1(keep=id effect variety estimate
                         rename=(estimate=ssdev));
run;
proc sort data=sf1;
by effect;
run;
proc sort data=sr1;
by effect;
run;
data final;
merge sf1 sr1;
by effect;
sscoeff = overall + ssdev;
run;
proc sort data=final;
by effect;
run;
data wt43;
set final;
wt43=sscoeff;
keep ID wt43;
run;
```

BIBLIOGRAPHY

Burke, L. E., Styn, M. A., Glanz, K., Ewing, L. J., Elci, O. U., Conroy, M. B., . . . Sevick, M. A. (2009). SMART Trial: A randomized clinical trial testing three methods of self-monitoring in behavioral weight management – design, baseline characteristics and self-monitoring intervention. *Contemporary Clinical Trials, 30*(6), 540-551.

Burke, L. E., Styn, M. A., Sereika, S. M., Conroy, M. B., Ye, L., Glanz, K., . . . Ewing, L. J. (2012). Using mHealth technology to enhance self-monitoring for weight loss: a randomized trial. *Am J Prev Med, 43*(1), 20-26. doi: 10.1016/j.amepre.2012.03.016

Cheng, M. Y., Fan, J., & Marron, J. S. (1993). *Minimax efficiency of local polynomial fit estimators at boundaries*. Institute of Statistics mimeo Series #2098, University of North Carolina at Chapel Hill.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of American Statistical Association, 74*, 829-836.

Cleveland, W. S., & Devlin, S. J. (1988). Locally-weighted regression: an approach to regression analysis by local fitting. *Journal of American Statistical Association, 83*, 597-610.

Davidian, M., & Giltinan, D. M. (1995). *Nonlinear Models for Repeated Measurement Data*. Chapman and Hall, London.

Demidenko, E. (2004). *Mixed Models: Theory and Applications*. Wiley, New York.

Dempster, A. P., Rubin, D. B., & Tsutakawa, R. K. (1981). Estimation in covariance components models. *Journal of American Statistical Association, 76*, 341-353.

Diggle, P. J. (1988). An approach to the analysis of repeated measurements. *Biometrics, 44*, 959-971.

Fan, J. (1992). Design-adaptive nonparametric regression. *Journal of American Statistical Association, 87*, 998-1004.

Fan, J., & Gijbels, I. (1992). Variable bandwidth and local linear regression smoother. *The Annals of Statistics, 20*, 2008-2036.

Fan, J., Gijbels, I., Hu, T., & Huang, L. (1996). A Study of Variable Bandwidth Selection for Local Polynomial Regression *Statistica Sinica, 6*, 113-127.

Fan, J., & Zhang, J. T. (2000). Two-step estimation of functional linear models with application to longitudinal data. *Journal of Royal Statistical Society, Series B, 62*, 303-322.

Harville, D. A. (1976). Extension of the Gauss-Markov theorem to include the estimation of random effects. *Annals of Statistics, 4*, 384-395.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of American Statistical Association, 72*(320-340).

Hastie, T. J., & Loader, C. (1993). Local regression: automatic kernel carpentry (with discussion). *statistical science, 8*, 120-143.

Hastie, T. J., & Tibshirani, R. (1990). *Generalized Additive Models*. Chapman and Hall, London.

Laird, N. M., Lange, N., & Stram, D. (1987). Maximum likelihood computations with repeated measures. *Journal of American Statistical Association, 82*, 97-105.

Laird, N. M., & Ware, J. H. (1982). Random effects models for longitudinal data. *Biometrics, 38*, 963-974.

Muller, H. G. (1987). Weighted local regression and kernel methods for nonparametric curve fitting. *Journal of American Statistical Association, 82*, 231-238.

Ngo, L., & Brand, R. (2002). Model selection in linear mixed effects models using SAS Proc Mixed. *SUGI*, 22.

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-Effects models in S and S-PLUS*. Springer, New York.

Ruppert, D., Sheather, S. J., & Wand, M. P. (1995). An effective bandwidth selector for local least square regression. *Journal of American Statistical Association, 90*, 1257-1270.

Ruppert, D., & Wand, M. P. (1994). Multivariate weighted least squares regression. *The Annals of Statistics, 22*, 1346-1370.

Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press.

Vonesh, E. F., & Chinchilli, V. M. (1996). *Linear and Nonlinear Models for the Analysis of Repeated Measurements*: Marcel Dekker, New York.

Wand, M. P., & Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

Wu, H., & Zhang, J. (2006). *Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed-Effects Modeling Approaches* John Wiley & Sons, New Jersey.

Zeger, S. L., & Diggle, P. J. (1994). Semiparametric models for longitudinal data with application to CD4 cell numbers in HIV seroconverters. *Biometrics, 50*, 689-699.

Zhang, D., Lin, X., Raz, J., & Sowers, M. (1998). Semiparametric stochastic mixed models for longitudinal data. *Journal of American Statistical Association, 93*, 710-719.