# Web-based Supporting Materials for Regression modeling of longitudinal data with outcome-dependent observation times: Extensions and comparative evaluation

by Kay See Tan, Benjamin French, Andrea B. Troxel

In this supplement, we provide theoretical results for the Weighted-Liang and Weighted-Sun methods. We include R code to reproduce estimates and standard errors from Case Study Table 4 using publicly available data. The standard errors are calculated from 1000 cluster-bootstrap samples in which the subjects are the sampling units.

## Appendix A Theoretical results for extensions to the Liang method: Weighted-Liang

In this appendix, we provide the theoretical results for our extention to the Liang method to allow for time-dependent covariates in the observation-time model as well as to accommodate M2. We assume that the frailty $\eta_{i2}$ follows a Gamma distribution with mean 1, variance $\sigma^2$ and $\mathrm{E}(\eta_{i1} \mid \eta_{i2}) = \theta(\eta_{i2} - 1)$. Following the notation from Liang et al. [1], we first define:

$$
\begin{aligned}
S_Z^{(k)}(t,\gamma) &= n^{-1}\sum_{i=1}^{n}\xi_i(t)Z_i^k(t)\exp\{\gamma' Z_i(t)\}\\
P_X^{(k)}(t,\delta,\Lambda) &= n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}X_i^k\frac{m_i}{\pi(C_i;Z_i)}\\
P_B^{(1)}(t,\delta,\Lambda) &= n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}B_i\frac{m_i}{\pi(C_i;Z_i)},
\end{aligned}
$$

in which $k = 0, 1$, and $k = 1$ indexes the presence of the covariate $X_i(t)$ or $Z_i(t)$. We let $s_Z^{(k)}(t)$, $p_X^{(k)}(t)$, $p_B^{(1)}(t)$, $\mu_Z(t)$, $\tilde{\mu}_X(t)$, and $\tilde{\mu}_B(t)$ be the asymptotic limit of $S_Z^{(k)}(t,\gamma)$, $P_X^{(k)}(t,\Lambda_0)$, $P_B^{(1)}(t,\Lambda)$, $\frac{S_Z^{(1)}(t,\gamma)}{S_Z^{(0)}(t,\gamma)}$, $\frac{P_X^{(1)}(t,\Lambda)}{P_X^{(0)}(t,\Lambda)}$, and $\frac{P_B^{(1)}(t,\Lambda)}{P_X^{(0)}(t,\Lambda)}$. Let $\phi = (\beta', \alpha')'$.
Furthermore, define the mean-zero processes as:

$$
M_i(t) = M_i(t,\beta,\theta,\mathcal{A},\Lambda,B_i) = \int_0^t \frac{1}{\rho_i(s;\gamma,\delta)}\left[\{Y_i(s) - \beta' X_i(s) - \theta' B_i(s)\}dN_i(s) - \xi_i(s)m_i\frac{d\mathcal{A}(s)}{\Lambda(C_i)}\right],
$$

and:

$$
M_i^*(t) = N_i(t) - \int_0^t \xi_i(u)\exp\{\gamma' X_i(t)\}d\Lambda(u)
$$

and the positive definite matrix:

$$
A = E\left[\int_0^\tau \{Z_i - \mu_Z(t)\}^{\otimes 2}\xi_i(t)\exp\{\gamma' Z_i(t)\}d\Lambda(t)\right]
$$

## A.1 Asymptotic results for $\hat{\gamma}$

Lin et al. [2] showed the consistency of $\hat{\gamma}$, which can be written as:

$$\sqrt{n}(\hat{\gamma}-\gamma) = A^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\int_0^{\tau}\{Z_i(t)-\mu_Z(t)\}dM_i^*(t)+o_p(1). \tag{A.1}$$

## A.2 Asymptotic results for $\hat{\Lambda}(t)$

Additionally, $\sqrt{n}(\hat{\Lambda}(t)-\Lambda(t))$ is asymptotically equivalent to

$$\sqrt{n}\{\hat{\Lambda}(t)-\Lambda(t)\} = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\int_0^{t}\frac{dM_i^*(u)}{S_Z^{(0)}(u)} - \int_0^{t}\mu_Z'(u)d\Lambda(u)A^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\int_0^{\tau}\{Z_i(t)-\mu_Z(t)\}dM_i^*(t)+o_p(1). \tag{A.2}$$

## A.3 Asymptotic results for $\hat{\sigma}^2$

According to Liang et al. [1],

$$\sqrt{n}(\hat{\sigma}^2-\sigma^2) = \frac{1}{\sqrt{n}}\frac{1}{T}\sum_{i=1}^{n}\{m_i^2-m_i-T(\sigma^2+1)\}+o_p(1), \tag{A.3}$$

in which $T=\lim_{n\to\infty}(1/n)\sum_{i=1}^{n}\exp\{\gamma' Z_i(t)\}\Lambda(C_i)$.

## A.4 Asymptotic properties of $(1/\sqrt{n})U_1(\phi,\hat{\Lambda},\hat{B})$

Let

$$U_{11}(\phi,\hat{\Lambda},\hat{B}) = \sum_{i=1}^{n}\int_0^{\tau}\frac{1}{\rho_i(t;\hat{\gamma},\delta)}\{X_i(t)-\bar{X}(t)\}\{Y_i(t)-\beta' X_i(t)-\theta'\hat{B}_i(t)\}dN_i(t),$$

$$U_{12}(\phi,\hat{\Lambda},\hat{B}) = \sum_{i=1}^{n}\int_0^{\tau}\frac{1}{\rho_i(t;\hat{\gamma},\delta)}\{\hat{B}_i(t)-\bar{\hat{B}}(t)\}\{Y_i(t)-\beta' X_i(t)-\theta'\hat{B}_i(t)\}dN_i(t),$$

in which $\bar{X}(t)$ and $\bar{\hat{B}}(t)$ are as previously defined. Expressions to derive the asymptotic properties of $(1/\sqrt{n})U_1(\phi,\hat{\Lambda},\hat{B})$ largely follows that outlined in Liang et al. [1], with the inclusion of the observation-level inverse weights $\frac{1}{\rho_i(t;\gamma,\delta)}$, changing $V_i$ to $Z_i(t)$, and replacing the sub-functions (i.e., $S_Z^{(k)}(t,\gamma)$, $M_i(t,\beta,\theta,\mathcal{A},\Lambda,B_i)$) as we have described above.

# Appendix B Theoretical results for extensions to the Sun method: Weighted-Sun

In this appendix, we provide the theoretical results for our extension to the Sun method to accommodate M2 under the assumption that the observation-time process is conditionally independent of the censoring times, following the notation from Sun et al. [3].

We first define:

$$S_z^{(k)}(t;\gamma) = n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\gamma' Z_i(t)\}Z_i^k(t)$$
$$S^{(0)}(t;\delta) = n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}\ m_i/\pi(C_i;Z_i)$$
$$S_x^{(k)}(t;\delta) = n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}X_i^k(t)\ m_i/\pi(C_i;Z_i)$$
$$S_\eta^{(1)}(t;\delta) = n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}\hat{\eta}_i(t)\ m_i/\pi(C_i;Z_i)$$
$$S_{\eta x}^{(2)}(t;\delta) = n^{-1}\sum_{i=1}^{n}\xi_i(t)\exp\{\delta' X_i(t)\}\hat{\eta}_i X_i(t)\ m_i/\pi(C_i;Z_i)$$

Let $s_z^{(0)}(t)$, $s_z^{(1)}(t)$, $s^{(0)}(t)$, $s_x^{(1)}(t)$, $s_x^{(2)}(t)$, $s_\eta^{(1)}(t)$, $s_{\eta x}^{(2)}(t)$ and $\mu_Z(t)$ denote the limiting values of $S_z^{(0)}(t;\gamma_0)$, $S_z^{(1)}(t;\gamma_0)$, $S^{(0)}(t;\delta)$, $S_x^{(1)}(t;\delta)$, $S_x^{(2)}(t;\delta)$, $S_\eta^{(1)}(t;\delta)$, $S_{\eta x}^{(2)}(t;\delta)$ and $s_z^{(1)}(t)/s_z^{(0)}(t)$. Furthermore, let $\bar{x}(t) = s_x^{(1)}(t)/s^{(0)}(t)$ and $\bar{\eta}(t) = s_\eta^{(1)}(t)/s^{(0)}(t)$. Let $A$ be as defined in the previous section.

If the observation-time process is conditionally independent of the censoring times, the asymptotic results of $\gamma$ and $\Lambda$ follows A.1 and A.2.

Let

$$U_1(\beta,\alpha;\hat{\gamma}) = \sum_{i=1}^{n}\int_0^\tau \frac{W(t)}{\rho_i(t;\hat{\gamma},\delta)}[\{X_i(t)-\bar{X}(t)\}\{Y_i(t)-\beta' X_i(t)-\alpha\hat{\eta}_i\}]dN_i(t)$$
$$U_2(\beta,\alpha;\hat{\gamma}) = \sum_{i=1}^{n}\int_0^\tau \frac{W(t)}{\rho_i(t;\hat{\gamma},\delta)}[\{\hat{\eta}_i-\bar{\eta}(t)\}\{Y_i(t)-\beta' X_i(t)\}-\alpha\{\hat{\Omega}_i-\hat{\eta}_i\bar{\eta}(t)\}]dN_i(t),$$

Define $U(\beta,\alpha;\hat{\gamma}) = (U_1(\beta,\alpha;\hat{\gamma})', U_1(\beta,\alpha;\hat{\gamma})')$. The asymptotic properties of $(1/\sqrt{n})U(\beta,\alpha;\hat{\gamma})$ follows from the expressions of Sun et al. [3], except with the assumption of non-informative censoring, the inclusion of the observation-level inverse weights $\frac{1}{\rho_i(t;\gamma,\delta)}$, and replacing the sub-functions (i.e., $S_z^{(k)}(t;\gamma)$) as we have described above.

We develop the asymptotic results for Appendix A and B under the assumption that $\delta$ is a fixed value. In practice, $\delta$ is estimated by $\hat{\delta}$ using the data. Following Liang and Zeger [4], the variability from estimating $\delta$ does not affect the asymptotic behavior of $\hat{\beta}$ using stabilized weights.

# Appendix C Relative efficiency

We examine the relative efficiency of weighted versus unweighted methods under Simulation Setting 1 to determine the potential loss of efficiency when methods include an additional covariate when it is not necessary.

Table 1: Simulation results for $\beta_1$ under (M2): Bias, $\hat{\beta}_1 - \beta_1$, $\beta_1 = 1$; ESE, empirical sample error; ERE, estimated relative efficiency

| $n$ | $\beta_2$ | $\gamma_2$ | **Lin** | | **Bůžková** | | | **Liang** (extension) | | **Weighted-Liang** (extension) | | | **Sun** | | **Weighted-Sun** (extension) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Bias | ESE | Bias | ESE | ERE[a] | Bias | ESE | Bias | ESE | ERE[b] | Bias | ESE | Bias | ESE | ERE[c] |
| 100 | 0 | 0 | 0.003 | 0.284 | 0.003 | 0.284 | 1.000 | 0.004 | 0.280 | −0.006 | 0.393 | 1.970 | 0.004 | 0.282 | −0.006 | 0.393 | 1.942 |
| | | −0.2 | −0.001 | 0.292 | −0.002 | 0.289 | 0.980 | 0.003 | 0.287 | −0.012 | 0.428 | 2.224 | 0.003 | 0.288 | −0.013 | 0.428 | 2.209 |
| | | 0.5 | −0.020 | 0.342 | −0.009 | 0.289 | 0.714 | −0.018 | 0.324 | −0.007 | 0.379 | 1.368 | −0.018 | 0.329 | −0.008 | 0.379 | 1.327 |
| | 0.3 | 0 | 0.003 | 0.318 | 0.002 | 0.313 | 0.969 | 0.004 | 0.313 | −0.008 | 0.411 | 1.724 | 0.004 | 0.315 | −0.007 | 0.411 | 1.702 |
| | 1 | 0 | 0.003 | 0.543 | −0.001 | 0.521 | 0.921 | 0.004 | 0.536 | −0.010 | 0.577 | 1.159 | 0.004 | 0.539 | −0.010 | 0.577 | 1.146 |
| 200 | 0 | 0 | −0.004 | 0.203 | −0.003 | 0.203 | 1.000 | −0.002 | 0.202 | −0.010 | 0.284 | 1.977 | −0.003 | 0.202 | −0.010 | 0.285 | 1.991 |
| | | −0.2 | −0.004 | 0.213 | −0.004 | 0.209 | 0.963 | −0.001 | 0.210 | −0.009 | 0.304 | 2.096 | −0.001 | 0.211 | −0.009 | 0.304 | 2.076 |
| | | 0.5 | 0.001 | 0.258 | 0.004 | 0.202 | 0.613 | −0.003 | 0.237 | 0.002 | 0.269 | 1.288 | −0.001 | 0.243 | 0.001 | 0.269 | 1.225 |
| | 0.3 | 0 | −0.007 | 0.227 | −0.007 | 0.224 | 0.974 | −0.006 | 0.225 | −0.014 | 0.299 | 1.766 | −0.006 | 0.226 | −0.014 | 0.299 | 1.750 |
| | 1 | 0 | −0.014 | 0.389 | −0.015 | 0.374 | 0.924 | −0.015 | 0.386 | −0.023 | 0.421 | 1.190 | −0.014 | 0.388 | −0.023 | 0.421 | 1.177 |

Estimated relative efficiency was calculated for unbiased estimators with:
[a] the variance of the Lin parameter estimate in the denominator;
[b] the variance of the Liang parameter estimate in the denominator;
[c] the variance of the Sun parameter estimate in the denominator.

# Appendix D  Density Curves

We use density curves of the latent variables as an informal graphical assessment of whether the latent variables are covariate-dependent. We plot the estimated values of the latent variables. Under the Weighted-Liang method, the estimation of $\eta_{i2}$ is based on Gamma distribution assumption. Under the Weighted-Sun method, the estimation of $\eta_i$ does not assume any parametric assumptions on the latent variables.

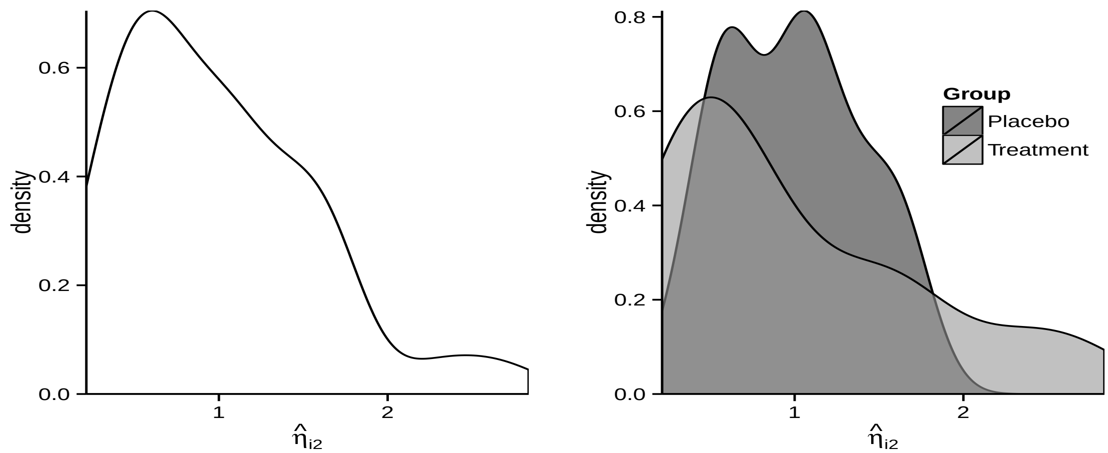


Figure 1: Density plot of estimated $\eta_{i2}$ under Weighted-Liang method

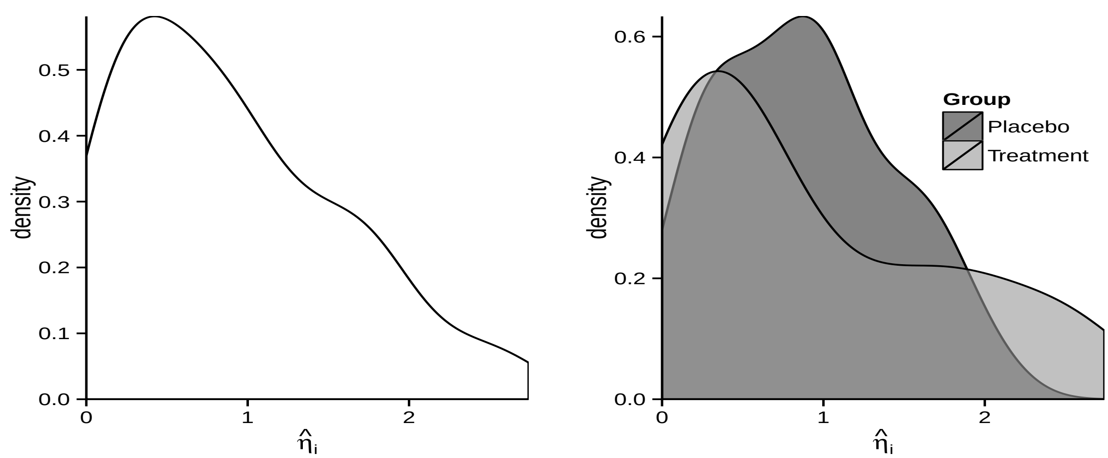


Figure 2: Density plot of estimated $\eta_i$ under Weighted-Sun method

# Appendix E  Residual plots

We show the residual plots from each of the models that utilized observation-level weights to assess adequacy of model-fit. R code to recreate these plots can be found in Appendix F. There is some evidence of lack of fit for large outcome values but it is not systematic with respect to time and are similar across all models.

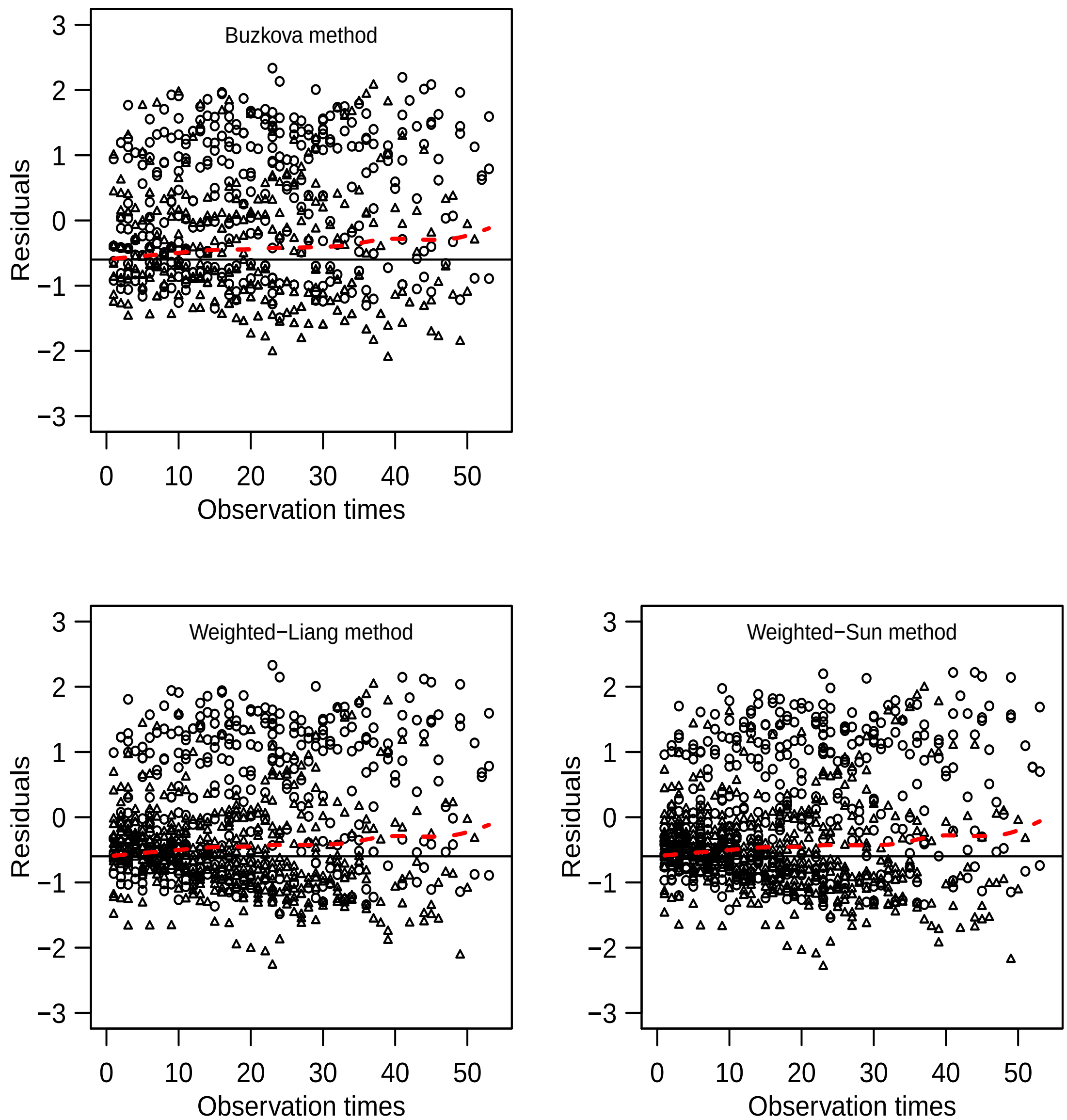


Figure 3: Residual plots by observation times

# Appendix F Residual plots

We provide R code to reproduce Table 4 from the case study and implement model-checking procedures.

```
##############################################
## Case Study Table 4: Bladder Data
## Additional Covariate:
## cumulative # of tumors since baseline
##############################################

## Load extension packages
library(Hmisc)
library(plyr)
library(zoo)
library(nleqslv)

## Load dataset
## The data is available from the delisted r package spef,
## available at http://cran.r-project.org/src/contrib/Archive/spef/
## The dataset: blaTum
library(spef); data(blaTum);

m <- NULL    # m = number of observations per subject
for (ii in unique(blaTum$id)){
  m[blaTum$id==ii] <- nrow(blaTum[blaTum$id==ii,])
}

######################################################################
## X3 <- additional covariate in observation-time model: # of new tumors since baseline
prev_cumtumors <- ave(blaTum$count, blaTum$id, FUN = function(x) Lag(cumsum(x), shift=1))
prev_cumtumors[is.na(prev_cumtumors)]<-0

## outcome Y = cumulative number of new tumors since baseline
Ycumtumors <- ave(blaTum$count, blaTum$id, FUN = cumsum)
######################################################################

#######################################################################
# Analysis dataset
# Q=Q(t) from Liang approach: Q(t) may be an outcome model covariate
# Here: Q=blaTum$treatment
#######################################################################
sim.data <- data.frame(ID=blaTum$id, t=blaTum$time, m=m, Y = log(Ycumtumors+1), X1=blaTum$treatment,
                       X2=log(blaTum$num+1), X3=log(prev_cumtumors+1), Q=blaTum$treatment, C=53) # Q=1

#/****** Select first record of each patient **/
baseData <- ddply(sim.data, .(ID), function(x) x[1, ])
meannumvisits <- mean(baseData$m)
udt <- unique(sort(sim.data$t[sim.data$t>0]))
N <- length(baseData$ID)

## Start function
DepObsTimes <- function(sim.data, baseData, udt){
  ###########################################################################
  #### expand time-varying covariates to full set of t=53 rows            ####
  #### Analyst may consider other methods such as imputation or closest neighbor
  ###########################################################################
  testdata  <- data.frame(ID=sim.data$ID, t=sim.data$t, X3=sim.data$X3)
  testdata2 <- reshape(testdata, timevar="t", idvar="ID", direction="wide", v.names="X3")
  testdata3 <- reshape(testdata2, idvar="ID",direction="long")
  testdata3 <- testdata3[order(testdata3$ID,testdata3$t),]
  testdata4 <- NULL
  for (i in unique(testdata3$ID)){testdata4 <- rbind(testdata4, na.locf(testdata3[testdata3$ID==i,]))}
  testdata4[is.na(testdata4)] <- 0

  #################################################################
  # GAMMA.HAT (observation-time model covariates)
  #################################################################
  f <- function(gamma){
    exp_gamma <- function(tt){exp(gamma[1]*baseData$X1+gamma[2]*baseData$X2+gamma[3]*testdata4$X3[testdata4$t==tt])}
    numer1 <- sapply(sim.data$t, function(u){sum( (baseData$X1*exp_gamma(u))[u<=baseData$C], na.rm=T) } )
    numer2 <- sapply(sim.data$t, function(u){sum( (baseData$X2*exp_gamma(u))[u<=baseData$C], na.rm=T) } )
    numer3 <- sapply(sim.data$t, function(u){sum( (testdata4$X3[testdata4$t==u]*exp_gamma(u))[u<=baseData$C], na.rm=T) } )
    denom  <- sapply(sim.data$t, function(u){sum( (exp_gamma(u))[u<=baseData$C], na.rm=T) } )
    Vbar <- cbind(numer1/denom, numer2/denom, numer3/denom)

    bigV <-cbind(sim.data$X1, sim.data$X2, sim.data$X3)
    temp <- colSums((bigV-Vbar)/N, na.rm=T)
    temp
```

```
}
gamma <- c(0.5, 0.5, 0.5)
gamma.hat <- nleqslv(gamma, f)$x

##################################################################
# DELTA.HAT (outcome model covariates)
##################################################################
f <- function(gamma){
  exp_delta <- exp(gamma[1]*baseData$X1+gamma[2]*baseData$X2)
  numer1 <- sapply(sim.data$t, function(u){sum( (baseData$X1*exp_delta)[u<=baseData$C], na.rm=T) } )
  numer2 <- sapply(sim.data$t, function(u){sum( (baseData$X2*exp_delta)[u<=baseData$C], na.rm=T) } )
  denom  <- sapply(sim.data$t, function(u){sum( (exp_delta)[u<=baseData$C], na.rm=T) } )
  Vbar   <- cbind(numer1/denom, numer2/denom)
  Vbar.long  <- t(sapply(sim.data$t, function(tt) Vbar[tt,]))

  bigV <-cbind(sim.data$X1, sim.data$X2)
  temp <- colSums((bigV-Vbar.long)/N, na.rm=T)
  temp
}
gamma <- c(0, 0)
delta.hat <- nleqslv(gamma, f)$x

##################################################################
# Set-Up
##################################################################
exp_gamma <- function(u){exp(gamma.hat[1]*baseData$X1+
                             gamma.hat[2]*baseData$X2+gamma.hat[3]*testdata4$X3[testdata4$t==u])}
denom_gamma <- sapply(udt, function(u){sum( (exp_gamma(u))[u<=baseData$C], na.rm=T) } )
exp_delta <- exp(delta.hat[1]*baseData$X1+delta.hat[2]*baseData$X2)
denom_delta <- sapply(udt, function(u){sum( (exp_delta)[u<=baseData$C], na.rm=T) } )

#/**** : estimated dLam(t) under X1+X2 in observation-time model ****/
estlam.t.delta <- sapply(1:length(udt), function(u) sum( ((sim.data$t==udt[u])/denom_delta[u])) )

#/**** : estimated dLam(t) under X1+X2+X3 in observation-time model ****/
estlam.t.gamma <- sapply(1:length(udt), function(u) sum( ((sim.data$t==udt[u])/denom_gamma[u])) )

#/**** : estimated Ybar_star (closest neighbor): Only used in LY and Buzkova ****/
Y_star <- function(t){
  sapply(baseData$ID, function(n){
    tail(sim.data$Y[sim.data$ID==n][(abs(sim.data$t[sim.data$ID==n]-t)==min(abs(sim.data$t[sim.data$ID==n]-t)))],1) })
}

numer <- sapply(udt, function(u) sum((Y_star(u)*exp_delta)[ baseData$C >= u ]) )
Ybar_star <- (numer/denom_delta)
Ybar_starX <- (sapply(sim.data$t, function(tt) Ybar_star[tt]))

###########################################
#######################################
# LIN & YING METHOD
#######################################
###########################################
#/**** : estimated Xbar1 & Xbar2 ****/
denom  <- sapply(sim.data$t, function(u) sum(exp_delta[ baseData$C >= u  ]) )
numer1 <- sapply(sim.data$t, function(u) sum((baseData$X1*exp_delta)[ baseData$C >= u ]) )
numer2 <- sapply(sim.data$t, function(u) sum((baseData$X2*exp_delta)[ baseData$C >= u ]) )
Xbar <- cbind(numer1/denom, numer2/denom)
bigX <- as.matrix(cbind(sim.data$X1, sim.data$X2))

f <- function(beta){
  temp <- rep(0,length=ncol(bigX))
  temp[1] <- sum(((bigX[,1]-Xbar[,1])*(sim.data$Y-Ybar_starX-beta[1]*(bigX[,1]-Xbar[,1])
                          -beta[2]*(bigX[,2]-Xbar[,2]) )), na.rm=T)
  temp[2] <- sum(((bigX[,2]-Xbar[,2])*(sim.data$Y-Ybar_starX-beta[1]*(bigX[,1]-Xbar[,1])
                          -beta[2]*(bigX[,2]-Xbar[,2]) )), na.rm=T)
  temp
}
beta <- c(0,0)
LY.beta <- nleqslv(beta, f)$x

###########################################
#######################################
# BUZKOVA METHOD
#######################################
###########################################
#################/**** calculate weights (iirr2 = rho) ****/#############
Z <- cbind(sim.data$X1, sim.data$X2, sim.data$X3)
X <- cbind(sim.data$X1, sim.data$X2)
iirr2 <- exp(Z %*% as.matrix(gamma.hat))/exp(X %*% as.matrix(delta.hat))

bigX <- as.matrix(cbind(sim.data$X1, sim.data$X2))
```

```
f <- function(beta){
  temp <- rep(0,length=ncol(bigX))
  temp[1] <- sum( 1/iirr2*((bigX[,1]-Xbar[,1])*(sim.data$Y-Ybar_starX-beta[1]*
                                             (bigX[,1]-Xbar[,1])-beta[2]*(bigX[,2]-Xbar[,2]) )), na.rm=T)
  temp[2] <- sum( 1/iirr2*((bigX[,2]-Xbar[,2])*(sim.data$Y-Ybar_starX-beta[1]*
                                             (bigX[,1]-Xbar[,1])-beta[2]*(bigX[,2]-Xbar[,2]) )), na.rm=T)
  temp
}
beta <- c(0,0)
Buzkova.stable.beta <- nleqslv(beta, f)$x

###########################################
#######################################
# LIANG METHOD
#######################################
###########################################
gammaV_b <- sapply(baseData$ID, function(nn){sum((exp(delta.hat[1]*baseData$X1[baseData$ID==nn]+
                delta.hat[2]*baseData$X2[baseData$ID==nn])*estlam.t.delta)[udt<=baseData$C[baseData$ID==nn]])})

#/*** estsigma ***/
estsigma2 <- max(sum((baseData$m^2-baseData$m-(gammaV_b)^2)/sum((gammaV_b)^2)),0)

#/**** Bhat ****/
Bhat_i <- ( (1+baseData$m*estsigma2)/(1+gammaV_b*estsigma2)-1)
Bhat_long <- sapply(sim.data$ID, function(i) Bhat_i[baseData$ID==i])
Bhat <- sim.data$Q*Bhat_long

#/**** vector of "observed": ****/
bigX <- as.matrix(cbind(sim.data$X1, sim.data$X2, Bhat))
bigX_base <- as.matrix(cbind(baseData$X1, baseData$X2, Bhat_i*baseData$Q))

#/**** : estimated Xbar1 & Xbar2 & Bhat****/
denom  <- sapply(sim.data$t, function(u) sum((exp_delta*(baseData$m/gammaV_b))[ baseData$C >= u ]) )
numer1 <- sapply(sim.data$t, function(u) sum((bigX_base[,1]*exp_delta*(baseData$m/gammaV_b))[ baseData$C >= u ]) )
numer2 <- sapply(sim.data$t, function(u) sum((bigX_base[,2]*exp_delta*(baseData$m/gammaV_b))[ baseData$C >= u ]) )
numer3 <- sapply(sim.data$t, function(u) sum((bigX_base[,3]*exp_delta*(baseData$m/gammaV_b))[ baseData$C >= u ]) )

if (estsigma2 != 0 ){
  Xbar <- cbind(numer1/denom,numer2/denom,numer3/denom)
  f <- function(beta){
    temp <- rep(0,length=ncol(bigX))
    temp[1] <- sum(( (bigX[,1]-Xbar[,1])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp[2] <- sum(( (bigX[,2]-Xbar[,2])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp[3] <- sum(( (bigX[,3]-Xbar[,3])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp
  }
  beta <- c(0,0,0)
  Liang.beta <- nleqslv(beta, f)$x
}

if (estsigma2 == 0 ){
  Xbar <- cbind(numer1/denom,numer2/denom)
  f <- function(beta){
    temp <- rep(0,length=ncol(Xbar))
    temp[1] <- sum(( (bigX[,1]-Xbar[,1])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])), na.rm=T)
    temp[2] <- sum(( (bigX[,2]-Xbar[,2])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])), na.rm=T)
    temp
  }
  beta <- c(0,0)
  Liang.beta <- c(nleqslv(beta, f)$x, NA)
}

###########################################
#######################################
# WEIGHTED-LIANG METHOD
#######################################
###########################################
gammaV_b <- sapply(baseData$ID, function(nn){
  sum((exp(gamma.hat[1]*baseData$X1[baseData$ID==nn]+gamma.hat[2]*baseData$X2[baseData$ID==nn]+
             gamma.hat[3]*testdata4$X3[testdata4$ID==nn])*estlam.t.gamma)[udt<=baseData$C[baseData$ID==nn]])})

#/*** estsigma ***/
estsigma2 <- max(sum((baseData$m^2-baseData$m-(gammaV_b)^2)/sum((gammaV_b)^2)),0)

#/**** Bhat ****/
Bhat_i <- ( (1+baseData$m*estsigma2)/(1+gammaV_b*estsigma2)-1)
Bhat_long <- sapply(sim.data$ID, function(i) Bhat_i[baseData$ID==i])
Bhat <- sim.data$Q*Bhat_long

#/**** vector of "observed": ****/
bigX <- as.matrix(cbind(sim.data$X1, sim.data$X2, Bhat))
```

```
bigX_base <- as.matrix(cbind(baseData$X1, baseData$X2, Bhat_i*baseData$Q))

#/**** : estimated Xbar1 & Xbar2 & Bhat****/
denom  <- sapply(sim.data$t, function(u) sum((exp_delta*(baseData$m/gammaV_b))[ baseData$C >= u ]) )
numer1 <- sapply(sim.data$t, function(u) sum((bigX_base[,1]*exp_delta*(baseData$m/gammaV_b))[baseData$C>=u]))
numer2 <- sapply(sim.data$t, function(u) sum((bigX_base[,2]*exp_delta*(baseData$m/gammaV_b))[baseData$C>=u]))
numer3 <- sapply(sim.data$t, function(u) sum((bigX_base[,3]*exp_delta*(baseData$m/gammaV_b))[baseData$C>=u]))

if (estsigma2 != 0 ){
  Xbar <- cbind(numer1/denom,numer2/denom,numer3/denom)
  f <- function(beta){
    temp <- rep(0,length=ncol(bigX))
    temp[1] <- sum(1/iirr2*((bigX[,1]-Xbar[,1])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp[2] <- sum(1/iirr2*((bigX[,2]-Xbar[,2])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp[3] <- sum(1/iirr2*((bigX[,3]-Xbar[,3])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]-beta[3]*bigX[,3])), na.rm=T)
    temp
  }
  beta <- c(0,0,0)
  Weighted.Liang.beta <- nleqslv(beta, f)$x
}

if (estsigma2 == 0 ){
  Xbar <- cbind(numer1/denom,numer2/denom)
  f <- function(beta){
    temp <- rep(0,length=ncol(Xbar))
    temp[1] <- sum(1/iirr2*((bigX[,1]-Xbar[,1])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])), na.rm=T)
    temp[2] <- sum(1/iirr2*((bigX[,2]-Xbar[,2])*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])), na.rm=T)
    temp
  }
  beta <- c(0,0)
  Weighted.Liang.beta <- c(nleqslv(beta, f)$x, NA)
}

###########################################
#######################################
# SUN METHOD
#######################################
###########################################
piCi <- sapply(baseData$ID, function(n, t){sum( (exp(delta.hat[1]*baseData$X1[baseData$ID==n]+
                       delta.hat[2]*baseData$X2[baseData$ID==n])*estlam.t.delta)
                       [t<=baseData$C[baseData$ID==n]], na.rm=T) }, t=udt  )

### Zhat & Ohat ###
Zhat <- (baseData$m-1)/piCi
Ohat <- (baseData$m-1)*(baseData$m-2)/piCi^2

### Xbar ###
S0  <-  sapply(sim.data$t, function(u){sum((exp_delta*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
Sx1 <-  sapply(sim.data$t, function(u){sum((exp_delta*baseData$X1*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
Sx2 <-  sapply(sim.data$t, function(u){sum((exp_delta*baseData$X2*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
Sz  <-  sapply(sim.data$t, function(u){sum((exp_delta*Zhat*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})

Xbar1 <- Sx1/S0
Xbar2 <- Sx2/S0
Zbar <- Sz/S0

bigX <- cbind(sim.data$X1, sim.data$X2)
Zhat.long <- sapply(sim.data$ID, function(i) Zhat[baseData$ID==i])
Ohat.long <- sapply(sim.data$ID, function(i) Ohat[baseData$ID==i])

f <- function(beta){
  temp <- rep(0,length=ncol(bigX)+1)
  temp[1] <- sum(((bigX[,1]-Xbar1)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]
                                   -beta[3]*Zhat.long )), na.rm=T)
  temp[2] <- sum(((bigX[,2]-Xbar2)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]
                                   -beta[3]*Zhat.long )), na.rm=T)
  temp[3] <- sum(((Zhat.long - Zbar)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])
                 -beta[3]*(Ohat.long - Zhat.long*Zbar) ), na.rm=T)
  temp
}
beta <- c(1,-1,0)
Sun.beta <- nleqslv(beta, f)$x

###########################################
#######################################
# WEIGHTED-SUN METHOD
#######################################
###########################################
piCi <- sapply(baseData$ID, function(n, t){sum( (exp(gamma.hat[1]*baseData$X1[baseData$ID==n]+
               gamma.hat[2]*baseData$X2[baseData$ID==n]+gamma.hat[3]*testdata4$X3[testdata4$t==t & testdata4$ID==n])*
               estlam.t.gamma)[t<=baseData$C[baseData$ID==n]], na.rm=T) }, t=udt  )
```

```
  ### Zhat & Ohat ###
  Zhat <- (baseData$m-1)/piCi
  Ohat <- (baseData$m-1)*(baseData$m-2)/piCi^2

  ### Xbar ###
  S0  <-  sapply(sim.data$t, function(u){sum((exp_delta*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
  Sx1 <-  sapply(sim.data$t, function(u){sum((exp_delta*baseData$X1*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
  Sx2 <-  sapply(sim.data$t, function(u){sum((exp_delta*baseData$X2*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})
  Sz  <-  sapply(sim.data$t, function(u){sum((exp_delta*Zhat*(baseData$m/piCi))[u<=baseData$C], na.rm=T)})

  Xbar1 <- Sx1/S0
  Xbar2 <- Sx2/S0
  Zbar <- Sz/S0

  bigX <- cbind(sim.data$X1, sim.data$X2)
  Zhat.long <- sapply(sim.data$ID, function(i) Zhat[baseData$ID==i])
  Ohat.long <- sapply(sim.data$ID, function(i) Ohat[baseData$ID==i])

  f <- function(beta){
    temp <- rep(0,length=ncol(bigX)+1)
    temp[1] <- sum( ( (1/iirr2)*(bigX[,1]-Xbar1)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]
                                             -beta[3]*Zhat.long )), na.rm=T)
    temp[2] <- sum( ( (1/iirr2)*(bigX[,2]-Xbar2)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2]
                                             -beta[3]*Zhat.long )), na.rm=T)
    temp[3] <- sum( ( (1/iirr2)*(Zhat.long - Zbar)*(sim.data$Y-beta[1]*bigX[,1]-beta[2]*bigX[,2])
                    -beta[3]*(Ohat.long - Zhat.long*Zbar) ), na.rm=T)
    temp
  }
  beta <- c(1,-1,1)
  Weighted.Sun.beta <- nleqslv(beta, f)$x

  ## Obtain estimates
  est_out <- data.frame(t(gamma.hat),t(delta.hat), t(LY.beta), t(Buzkova.stable.beta),
                        t(Liang.beta),t(Weighted.Liang.beta), t(Sun.beta), t(Weighted.Sun.beta))
  est_out
}

###### Run function and save original estimates
sim.data.orig  <- sim.data
Orig_Est <- round(DepObsTimes(sim.data=sim.data, baseData=baseData, udt=udt),3)

#########################################################
###### CLUSTER-BOOTSTRAP for SD of current dataset ######
######## Subjects are sampled with replacement ##########
simout.sd <- NULL
Bcluster <- 1000
for (bbc in 1:Bcluster){
  if (bbc %% 2 ==0) cat("inner=", bbc, "\n")

  set.seed(bbc)
  #########################################################

  sim.data.sd <- NULL
  for(i in 1:N){
    select <- sample(unique(sim.data.orig$ID),1)
    m <- nrow(sim.data.orig[sim.data.orig$ID==select,])
    sim.data.sd <- rbind(sim.data.sd, data.frame(ID=rep(i, nrow=m), sim.data.orig[sim.data.orig$ID==select,]))
    i <- i+1
  }
  baseData.sd <- ddply(sim.data.sd, .(ID), function(x) x[1, ])
  udt.sd <- unique(sort(sim.data.sd$t[sim.data.sd$t>0]))

  #### Run function
  Est_SD <- DepObsTimes(sim.data=sim.data.sd, baseData=baseData.sd, udt=udt.sd)


  #### Cluster-ootstrap coefficients
  simout.sd <- rbind(simout.sd, Est_SD)
  bbc <- bbc+1
}

sd <- round(apply(simout.sd,2, sd, na.rm=T),3)

cat("\n\n ===============================================\n\n",
    "\n Bladder Cancer Case Study: Table 4\n",
    "\n (SE: Cluster-bootstrap based on", Bcluster, "repetitions)\n",
    "\n\n Estimation of gamma's (Observation-time model):",
    "\n gamma  ",paste("g1=",Orig_Est[1],"(",sd[1],"), g2=",Orig_Est[2],"(",sd[2],"),g3=",Orig_Est[3],"(",sd[3],")"),
    "\n\n Estimation of betas using different methods:",
    "\n LY     ",paste("b1=",Orig_Est[6], "(",sd[6],"), b2=",Orig_Est[7],"(", sd[7], ")"),
```

```
    "\n Buzkova",paste("b1=",Orig_Est[8], "(",sd[8],"), b2=",Orig_Est[9],"(", sd[9], ")"),
    "\n Liang  ",paste("b1=",Orig_Est[10],"(",sd[10],"), b2=",Orig_Est[11],"(",sd[11],"), theta=",Orig_Est[12],"(",sd[12],")"),
    "\n W-Liang",paste("b1=",Orig_Est[13],"(",sd[13],"), b2=",Orig_Est[14],"(",sd[14],"), theta=",Orig_Est[15],"(",sd[15],")"),
    "\n Sun    ",paste("b1=",Orig_Est[16],"(",sd[16],"), b2=",Orig_Est[17],"(",sd[17],"), alpha=",Orig_Est[18],"(",sd[18],")"),
    "\n W-Sun  ",paste("b1=",Orig_Est[19],"(",sd[19],"), b2=",Orig_Est[20],"(",sd[20],"), alpha=",Orig_Est[21],"(",sd[21],")"),
    "\n\n ================================================\n\n" )


######################################################
######################################################
# CHECK if latent variables are covariate-dependent
# Using density curves of estimated latent variables
######################################################
######################################################
#install.packages("ggplot2")
library(ggplot2)
library(grid)
require(gridExtra)


## First run original sim.data (from blaTum) within DepObsTimes() function to obtain individual estimates.

#/******** Density Curve of \eta_i under WEIGHTED-SUN METHOD: Gamma distributed eta-i2
gammaV_b <- sapply(baseData$ID, function(nn){sum((exp(gamma.hat[1]*baseData$X1[baseData$ID==nn]+
             gamma.hat[2]*baseData$X2[baseData$ID==nn]+gamma.hat[3]*testdata4$X3[testdata4$ID==nn])*
             estlam.t.gamma)[udt<=baseData$C[baseData$ID==nn]])})

#/*estimated variance and \eta_{i2}
estsigma2 <- max(sum((baseData$m^2-baseData$m-(gammaV_b)^2)/sum((gammaV_b)^2)),0)
eta_i2 <- (1+baseData$m*estsigma2)/(1+gammaV_b*estsigma2)

## combined
plot1 <- ggplot(baseData)+geom_density(alpha=.6, aes(x=eta_i2))+
  theme_bw()+theme(axis.line = element_line(color = 'black'))+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme( plot.background = element_blank()
         ,panel.grid.major = element_blank()
         ,panel.grid.minor = element_blank()
         ,panel.border = element_blank()
         ,panel.background = element_blank()) +xlab(expression(hat(eta)[i2]))
## by treatment group
plot2 <- ggplot(baseData)+geom_density(alpha=.6, aes(x=eta_i2, fill=as.factor(baseData$X1)))+
  theme_bw()+scale_fill_manual(values=c("grey20", "grey60"), name="Group", breaks=c("0", "1"),
                               labels=c("Placebo", "Treatment"))+theme(axis.line = element_line(color = 'black'))+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme( plot.background = element_blank()
         ,panel.grid.major = element_blank()
         ,panel.grid.minor = element_blank()
         ,panel.border = element_blank()
         ,panel.background = element_blank()) + theme(legend.position = c(.8, .7))+xlab(expression(hat(eta)[i2]))

grid.arrange(plot1, plot2, ncol=2)


#/******** Density Curve of \eta_i under WEIGHTED-SUN METHOD
piCi <- sapply(baseData$ID, function(n, t){sum( (exp(gamma.hat[1]*baseData$X1[baseData$ID==n]+
         gamma.hat[2]*baseData$X2[baseData$ID==n]+gamma.hat[3]*testdata4$X3[testdata4$t==t & testdata4$ID==n])*
         estlam.t.gamma)[t<=baseData$C[baseData$ID==n]], na.rm=T) }, t=udt  )

piCi <- sapply(baseData$ID, function(n, t){sum( (exp(delta.hat[1]*baseData$X1[baseData$ID==n]+
         delta.hat[2]*baseData$X2[baseData$ID==n])*
         estlam.t.delta)[t<=baseData$C[baseData$ID==n]], na.rm=T) }, t=udt  )
#/*estimated \eta_i
Zhat <- (baseData$m-1)/piCi

## combined
plot3 <- ggplot(baseData)+geom_density(alpha=.6, aes(x=Zhat))+
  theme_bw()+theme(axis.line = element_line(color = 'black'))+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme( plot.background = element_blank()
         ,panel.grid.major = element_blank()
         ,panel.grid.minor = element_blank()
         ,panel.border = element_blank()
         ,panel.background = element_blank()) +xlab(expression(hat(eta)[i]))
## by treatment group
plot4 <- ggplot(baseData)+geom_density(alpha=.6, aes(x=Zhat, fill=as.factor(baseData$X1)))+
  theme_bw()+scale_fill_manual(values=c("grey20", "grey60"), name="Group", breaks=c("0", "1"),
                               labels=c("Placebo", "Treatment"))+theme(axis.line = element_line(color = 'black'))+
  scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0))+
  theme( plot.background = element_blank()
         ,panel.grid.major = element_blank()
```

```
         ,panel.grid.minor = element_blank()
         ,panel.border = element_blank()
         ,panel.background = element_blank()) + theme(legend.position = c(.8, .7))+xlab(expression(hat(eta)[i]))

grid.arrange(plot3, plot4, ncol=2)




##############################################
##############################################
# Check overall model-fit using residuals
# Only focus on models with weights (case 2)
##############################################
##############################################
##/* setup */
exp_delta <- exp(delta.hat[1]*baseData$X1+delta.hat[2]*baseData$X2)
denom_delta <- sapply(udt, function(u){sum( (exp_delta)[u<=baseData$C], na.rm=T) } )
estlam.t.delta <- sapply(1:length(udt), function(u) sum( ((sim.data$t==udt[u])/denom_delta[u])) )

exp_gamma <- function(u){exp(gamma.hat[1]*baseData$X1+
                              gamma.hat[2]*baseData$X2+gamma.hat[3]*testdata4$X3[testdata4$t==u])}
denom_gamma <- sapply(udt, function(u){sum( (exp_gamma(u))[u<=baseData$C], na.rm=T) } )
estlam.t.gamma <- sapply(1:length(udt), function(u) sum( ((sim.data$t==udt[u])/denom_gamma[u])) )

plot_resid <- function(est_resid, addtitle){
  plot(est_resid[sim.data$X1==0]~sim.data$t[sim.data$X1==0], pch=1, ylim=c(-3, 3), xlim=c(0,54),
       xlab="Observation times", ylab="Residuals", las=1, cex=.7)
  par(new=T)
  plot(est_resid[sim.data$X1==1]~sim.data$t[sim.data$X1==1], pch=2, ylim=c(-2, 3), xlim=c(0,54),
       xlab=" ", ylab=" ", xaxt="n", yaxt='n', cex=.5)
  abline(h=0)
  lines(smooth.spline(est_resid~sim.data$t, df = 10), lty = 2, col = "red", lwd=2)
  mtext(addtitle, 3, line=-1.2, cex=.7)
}

##/************** Buzkova method **************/##
#/*** d\mathcal{A}(t) by id ***/
numer <- (1/iirr2)*(sim.data$Y-Buzkova.stable.beta[1]*sim.data$X1-Buzkova.stable.beta[2]*sim.data$X2)
mathcal_A.delta <- sapply(1:length(udt), function(uu) sum(numer[sim.data$t==udt[uu]]/denom_delta[uu], na.rm=T) )

alpha_hat_step <- mathcal_A.delta/estlam.t.gamma
alpha_hat_sim  <- sapply(sim.data$t, function(uu) alpha_hat_step[udt==uu] )

#/*** residuals ***/#
pred_Y_Buzkova <- Buzkova.stable.beta[1]*sim.data$X1+Buzkova.stable.beta[2]*sim.data$X2
est_resid_Buzkova <- sim.data$Y - alpha_hat_sim - Buzkova.stable.beta[1]*sim.data$X1-Buzkova.stable.beta[2]*sim.data$X2
plot_resid(est_resid_Buzkova, "Buzkova method")


##/************** Weighted-Liang method: Q=X1 **************/##
#/*** Bhat ****/
Bhat2_i    <- ( (1+baseData$m*estsigma2)/(1+gammaV_b*estsigma2)-1)
Bhat_long <- sapply(sim.data$ID, function(i) Bhat2_i[baseData$ID==i])*sim.data$Q

#/*** d\mathcal{A}(t) by id ***/
numer <- (1/iirr2)*(sim.data$Y-Weighted.Liang.beta[1]*sim.data$X1-Weighted.Liang.beta[2]*sim.data$X2-
                    Weighted.Liang.beta[3]*Bhat_long)
mathcal_A.delta <- sapply(1:length(udt), function(uu) sum(numer[sim.data$t==udt[uu]]/denom_delta[uu], na.rm=T) )

alpha_hat_step <- mathcal_A.delta/estlam.t.gamma
alpha_hat_sim  <- sapply(sim.data$t, function(uu) alpha_hat_step[udt==uu] )

#/*** residuals ***/#
pred_Y_WLiang_X1 <- Weighted.Liang.beta[1]*sim.data$X1+Weighted.Liang.beta[2]*sim.data$X2+Weighted.Liang.beta[3]*Bhat_long
est_resid_WLiang_X1 <- sim.data$Y - alpha_hat_sim - Weighted.Liang.beta[1]*sim.data$X1-
  Weighted.Liang.beta[2]*sim.data$X2-Weighted.Liang.beta[3]*Bhat_long
plot_resid(est_resid_WLiang_X1, "Weighted-Liang method")


##/************** Weighted-Sun method **************/##

#/**** function for observation-level weights ****/
Z <- function(uu){cbind(baseData$X1, baseData$X2, testdata4$X3[testdata4$t==udt[uu]])}
X <- function(uu){cbind(baseData$X1, baseData$X2)}
iirr2b <- function(uu){exp(Z(uu) %*% as.matrix(gamma.hat))/exp(X(uu) %*% as.matrix(delta.hat))}

#/****/
piCi <- sapply(baseData$ID, function(n, t){sum( (exp(gamma.hat[1]*baseData$X1[baseData$ID==n]+
         gamma.hat[2]*baseData$X2[baseData$ID==n]+gamma.hat[3]*testdata4$X3[testdata4$t==t & testdata4$ID==n])*
         estlam.t.gamma)[t<=baseData$C[baseData$ID==n]], na.rm=T) }, t=udt  )
```

```
### Zhat & Ohat ###
Zhat <- (baseData$m-1)/piCi
Zhat.long <- sapply(sim.data$ID, function(i) Zhat[baseData$ID==i])
numer <- (1/iirr2)*(sim.data$Y-Weighted.Sun.beta[1]*sim.data$X1-Weighted.Sun.beta[2]*sim.data$X2-Weighted.Sun.beta[3]*Zhat.long)
mathcal_A.delta <- sapply(1:length(udt), function(uu) sum(numer[sim.data$t==udt[uu]]/denom_delta[uu], na.rm=T) )

alpha_hat_step <- mathcal_A.delta/estlam.t.gamma
alpha_hat_sim  <- sapply(sim.data$t, function(uu) alpha_hat_step[udt==uu] )

#/*** residuals ***/#
pred_Y_WSun <- alpha_hat_sim + Weighted.Sun.beta[1]*sim.data$X1+Weighted.Sun.beta[2]*sim.data$X2+Weighted.Sun.beta[3]*Zhat.long
est_resid_WSun <- sim.data$Y - alpha_hat_sim - Weighted.Sun.beta[1]*sim.data$X1-
  Weighted.Sun.beta[2]*sim.data$X2-Weighted.Sun.beta[3]*Zhat.long
plot_resid(est_resid_WSun, "Weighted-Sun method")
```

# References


[1] Liang Y, Lu W, Ying Z. Joint modeling and analysis of longitudinal data with informative observation times. *Biometrics* 2009; **65**(2):377–384.

[2] Lin DY, Wei LJ, Yang I, Ying Z. Semiparametric regression for the mean and rate functions of recurrent events. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2000; **62**(4):711–730.

[3] Sun L, Song X, Zhou J. Regression analysis of longitudinal data with time-dependent covariates in the presence of informative observation and censoring times. *Journal of Statistical Planning and Inference* 2011; **141**(8):2902–2919.

[4] Liang KY, Zeger SL. Longitudinal data analysis using generalized linear models. *Biometrika* 1986; **73**(1):13–22.