Web-based Supporting Materials for "Bayesian regression model for recurrent event data with event-varying covariate effects and event effect" by Li-An Lin, Sheng Luo, and Barry R. Davis

Table 1: Simulation results of the simulation study III.

	Frailty Model				PWP-GT Model			
	Bias	SEM	SE	CP(%)	Bias	SEM	SE	CP(%)
$\beta = 0.5$	−0.011	0.091	0.942	92.8	−0.011	0.098	0.095	94.9
$\eta = -0.5$	−0.012	0.090	0.825	95.7	0.005	0.096	0.089	95.5
	Conditional Frailty Model				Proposed Model			
	Bias	SEM	SE	CP(%)	Bias	SEM	SE	CP(%)
$\beta = 0.5$	0.019	0.110	0.119	91.9	0.007	0.099	0.100	96.3
$\eta = -0.5$	0.058	0.110	0.102	91.9	−0.013	0.099	0.095	94.4

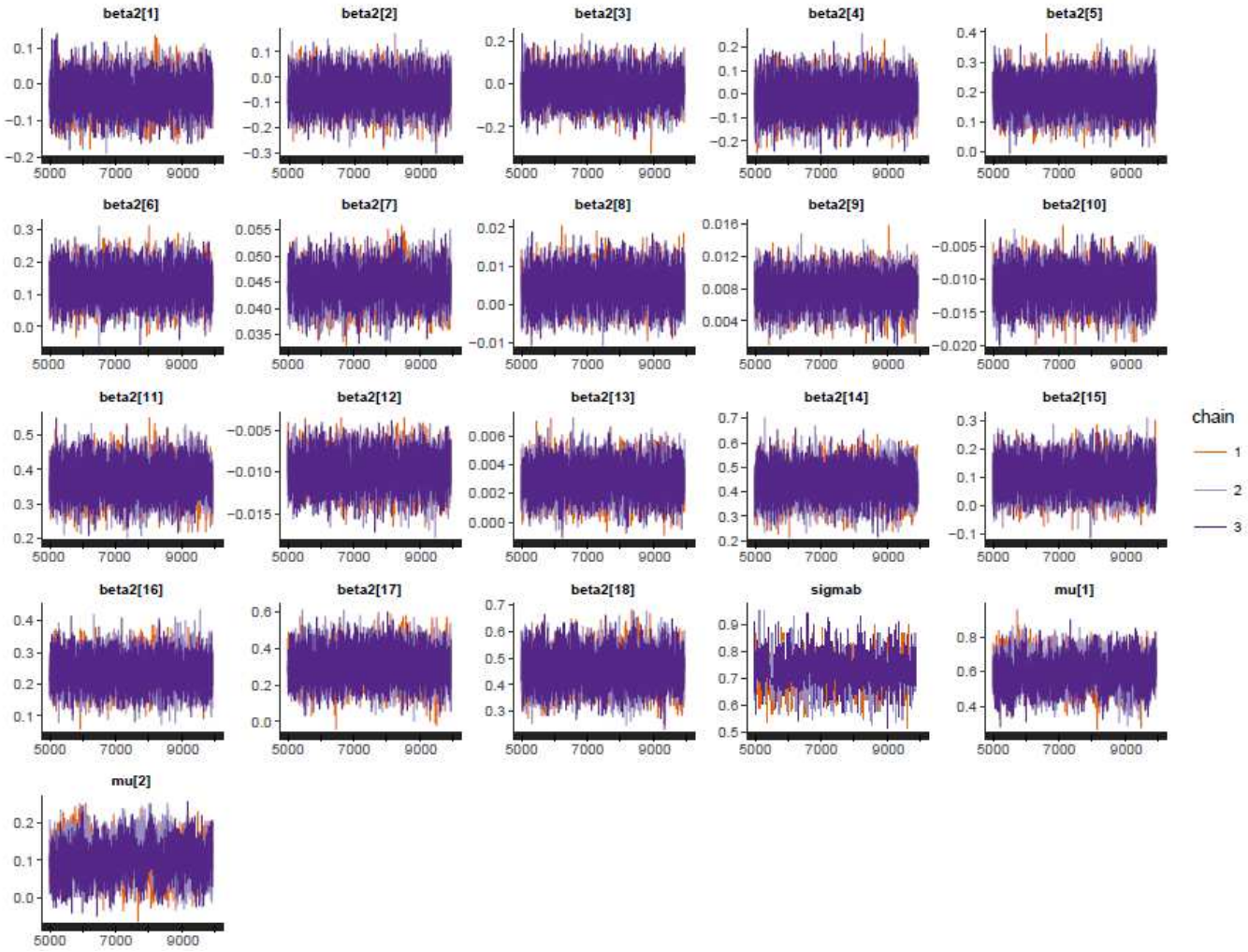

Figure 1: Trace plots to assess convergence of the proposed model.

Stan program to fit the proposed model

```
data {
  int I; // number of subject
  int TI; //number of total events
  int TIi; //number of pieces for quadratic integration
  int mz; //maximum number of events per subject
  int ncov; //number of covariates
  int nkot; //number of knots for spline regression
  int<lower=1,upper=I> iid[TI]; //id number for all recorded event
  int<lower=1,upper=I> iidi[TIi]; //id number for quadratic integration
  matrix[TI,ncov] xr; //covariate matrix for all recorded event
  vector[TI] trt; //treatment for all recorded event
  matrix[TIi,ncov] xri; //covariate matrix for quadratic integration
  vector[TIi] trti; //treatment for quadratic integration
  int<lower=1,upper=mz> znr1[TI]; //number of previous event occurred for all recorded event
```

```
  int<lower=1,upper=mz> znr1i[TIi]; //number of previous event occurred for quadratic integration
  matrix[TI,nkot] bhzr; //spline basis for all recorded event
  matrix[TIi,nkot] bhzri; //spline basis for quadratic integration
  vector[TIi] cstar1i; //weight for quadratic integration
}
parameters {
  vector[ncov] beta; //covariate effects
  vector[mz-1] ue; //event effect
  vector[mz] u1; //event-varying treatment effect
  vector[nkot] ur; //B-spline coefficients
  real<lower=0> sigmasqb; //variance parameter for frailty term
  real<lower=0> sigmasque; //variance parameter for event effect
  real<lower=0> sigmasqu1; //variance parameter for treatment effect
  real<lower=0> sigmasqur; //variance parameter for spline regression
  vector[I] bi; // frailty term
}
transformed parameters {
  real<lower=0> sigmasqb; //variance parameter for frailty term
  real<lower=0> sigmasque; //variance parameter for event effect
  real<lower=0> sigmasqu1; //variance parameter for treatment effect
  real<lower=0> sigmasqur; //variance parameter for spline regression
  vector[mz] ue2; //cumulative event effect
  sigmab <- sqrt(sigmasqb);
  sigmaue <- sqrt(sigmasque);
  sigmau1 <- sqrt(sigmasqu1);
  sigmaur <- sqrt(sigmasqur);
  ue2[1] <- 0;
  ue2[2:mz] <- cumulative_sum(ue);
}
model {
  vector[TI] b1;
  vector[TIi] b1i;
  vector[TI] covr1;
  vector[TIi] covr1i;
  #prior
  beta ~ normal(0,100);
  bi ~ normal(0,sigmab);
  ue[1] ~ normal(0,100);
  tail(ue,(mz-2)) ~ normal(head(ue,(mz-2)),sigmaue);
  u1[1] ~ normal(0,100);
  tail(u1,(mz-1)) ~ normal(head(u1,(mz-1)),sigmau1);
  ur[1] ~ normal(0,100);
  tail(ur,(nkot-1)) ~ normal(head(ur,(nkot-1)),sigmaur);
  sigmasqb ~ inv_gamma(0.001,0.001);
  sigmasque ~ inv_gamma(0.001,0.001);
  sigmasqu1 ~ inv_gamma(0.001,0.001);
  sigmasqur ~ inv_gamma(0.001,0.001);
  for (i in 1:TI){
    b1[i] <- bi[iid[i]]+ue2[znr1[i]]+trt[i]*u1[znr1[i]];
  }
  for (i in 1:TIi){
    b1i[i] <- bi[iidi[i]]+ue2[znr1i[i]]+trti[i]*u1[znr1i[i]];
  }
  #log-likelihood function
  covr1 <- bhzr*ur+xr*beta2+b1;
  covr1i <- bhzri*ur+xri*beta2+b1i+log(cstar1i);
  increment_log_prob(sum(covr1) - sum(exp(covr1i)));
}
}
```