library(rstan)

library(loo)

setwd("C:/R files BHMRA")

D = read.table("DS\_11\_1.txt",header=T)

attach(D)

**#**

**# Weibull PH**

**#**

weib.stan ="

data { int<lower=1> n; // number of cases

vector[n] time; // response

int<lower=0,upper=1> censored[n]; // indicates censoring (right censored=1)

int<lower=0> p; // number of regression parameters, including intercept

real<lower=0> age[n];

int<lower=0> trt[n];

int<lower=0> gender[n];

int<lower=0> marstat[n];

int<lower=0> hltst3[n];

int<lower=0> hltst4[n];

int<lower=0> hltst5[n];

}

parameters {vector[p] beta;

real<lower=0> kappa; // shape parameter

}

transformed parameters {

real eta[n];

real nu[n];

real sigma;

sigma = 1/kappa;

for (i in 1:n) {eta[i]=beta[1]+beta[2]\*age[i]/100+beta[3]\*trt[i]+beta[4]\*gender[i]

+beta[5]\*marstat[i]+beta[6]\*hltst3[i]+beta[7]\*hltst4[i]+beta[8]\*hltst5[i];

nu[i] = exp(-eta[i]/kappa); }

}

model { target += gamma\_lpdf(kappa | 0.01, 0.01);

for (i in 1:n) {

if (censored[i] == 0) { target += weibull\_lpdf(time[i] | kappa, nu[i]); }

else if (censored[i] == 1) { target += weibull\_lccdf(time[i] | kappa, nu[i]); } }

}

generated quantities{real log\_lik[n];

real S[n]; //survival function

real MR[n]; //martingale residual

real CH[n]; // cumulative hazard

for (i in 1:n) { S[i] = exp(weibull\_lccdf(time[i] | kappa, nu[i]));

CH[i] = -log(S[i]);

MR[i] = 1-censored[i]-CH[i];

if (censored[i] == 0) { log\_lik[i]= weibull\_lpdf(time[i] | kappa, nu[i]); }

else if (censored[i] == 1) { log\_lik[i]= weibull\_lccdf(time[i] | kappa, nu[i]); } }}

"

# Compilation

sm <- stan\_model(model\_code=weib.stan)

# Data

D=list(n=1601,p=8,time=time,censored=censored,age=age,trt=trt,gender=gender,marstat=marstat,

hltst3= hltst3, hltst4= hltst4, hltst5= hltst5)

# Estimation

fitwei <- sampling(sm,data =D,iter = 1500,warmup=250,chains = 2,seed= 12345)

summary(fitwei, pars = c("beta", "kappa", "sigma"), probs = c(0.025, 0.975))$summary

**# Estimated survival functions**

summary.S= summary(fitwei, pars = c("S"), probs = c(0.025,0.975))$summary

pm.S=summary.S[,1]

**# Residual measures**

**# Posterior mean Martingale residuals**

summary.MR= summary(fitwei,pars = c("MR"),probs= c(0.025,0.975))$summary

pm.MR=summary.MR[,1]

**# Normal deviate residuals**

qn.pm.S=numeric(D$n)

for (i in 1:D$n) {qn.pm.S[i]=ifelse(D$censored[i]==0,qnorm(pm.S[i]),

sum(qnorm(runif(10,0,pm.S[i])))/10)}

**# Fit**

LOO1= loo(as.matrix(fitwei,pars="log\_lik"))

loo.pw= LOO1$pointwise[,3]

**# individual fit**

subj=seq(1:D$n)

indfit=data.frame(subj,loo.pw,pm.MR,qn.pm.S)

indfit=indfit[order(-abs(qn.pm.S)),]

head(indfit)

**#**

**# Shrinkage priors on coefficients, Bayesmixsurv**

**#**

endstay=1-censored

D$agec=D$age/100

C1=bayesmixsurv(Surv(time,endstay)~trt+agec+marstat+gender+hltst3+hltst4+hltst5, D, control=bayesmixsurv.control(iter=1000,single=T))

#

**# AFT regression, generalized gamma**

#

gengamma.stan ="

functions{

real ggamma\_lpdf(real t, real alpha, real lambda, real gam) {

return( log(gam)+gam\*alpha\*(log(t)+log(lambda)) - (t\*lambda)^gam -log(t\*tgamma(alpha)));}

real ggamma\_S\_lpdf(real t, real alpha, real lambda, real gam) {

return( log(1- gamma\_p(alpha,(t\*lambda)^gam)));}

}

data { int<lower=1> n; // number of cases

vector[n] time; // response

int<lower=0,upper=1> censored[n]; // censoring indicator(0=uncensored, right censored=1)

int<lower=0> p; // number of regression parameters, including intercept

real age[n];

real trt[n];

real gender[n];

real marstat[n];

real hltst3[n];

real hltst4[n];

real hltst5[n];

}

parameters {vector[p] beta;

real<lower=0> alpha; // first shape parameter

real<lower=0> gam; // second shape parameter

}

transformed parameters {

real eta[n];

real lambda[n];

for (i in 1:n) { eta[i]=beta[1]+beta[2]\*age[i]/100+beta[3]\*trt[i]+beta[4]\*gender[i]

+beta[5]\*marstat[i]+beta[6]\*hltst3[i]+beta[7]\*hltst4[i]+beta[8]\*hltst5[i];

lambda[i] =exp(-eta[i]);}

}

model { target += gamma\_lpdf(alpha | 0.01, 0.01);

target += exponential\_lpdf(gam| 1);

for (i in 1:n) {

if (censored[i] == 0) { target += ggamma\_lpdf(time[i]|alpha, lambda[i], gam); }

else if (censored[i] == 1) { target += ggamma\_S\_lpdf(time[i]|alpha, lambda[i], gam); } }

}

generated quantities{real log\_lik[n];

for (i in 1:n) {

if (censored[i] == 0) { log\_lik[i]= ggamma\_lpdf(time[i] | alpha, lambda[i], gam); }

else if (censored[i] == 1) { log\_lik[i]= ggamma\_S\_lpdf (time[i] | alpha, lambda[i], gam); } }}

"

**# Compilation**

smgg <- stan\_model(model\_code=gengamma.stan)

**# Data, centred predictors**

D=list(n=1601,p=8,time=time,censored=censored,age=age-mean(age),trt=trt-mean(trt),gender=gender-mean(gender),marstat=marstat-mean(marstat),

hltst3= hltst3-mean(hltst3), hltst4= hltst4-mean(hltst4), hltst5= hltst5-mean(hltst5))

**# Estimation**

fitggam <- sampling(smgg,data =D,iter = 5000,warmup=250,chains = 1,seed= 12345)

summary(fitggam, pars = c("beta", "alpha", "gam"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

LOO2=loo(as.matrix(fitggam,pars="log\_lik"))

**#**

**# Weibull AFT**

**#**

weibAFT.stan ="

data { int<lower=1> n; // number of cases

vector[n] time; // response

int<lower=0,upper=1> censored[n]; // indicates censoring (right censored=1)

int<lower=0> p; // number of regression parameters, including intercept

real<lower=0> age[n];

int<lower=0> trt[n];

int<lower=0> gender[n];

int<lower=0> marstat[n];

int<lower=0> hltst3[n];

int<lower=0> hltst4[n];

int<lower=0> hltst5[n];

}

parameters {vector[p] beta;

real<lower=0> kappa; // shape parameter

}

transformed parameters {

real eta[n];

real nu[n];

real sigma;

sigma = 1/kappa;

for (i in 1:n) {eta[i]=beta[1]+beta[2]\*age[i]/100+beta[3]\*trt[i]+beta[4]\*gender[i]

+beta[5]\*marstat[i]+beta[6]\*hltst3[i]+beta[7]\*hltst4[i]+beta[8]\*hltst5[i];

nu[i] = exp(eta[i]); }

}

model { target += gamma\_lpdf(kappa | 0.01, 0.01);

for (i in 1:n) {

if (censored[i] == 0) { target += weibull\_lpdf(time[i] | kappa, nu[i]); }

else if (censored[i] == 1) { target += weibull\_lccdf(time[i] | kappa, nu[i]); } }

}

generated quantities{real log\_lik[n];

for (i in 1:n) {

if (censored[i] == 0) { log\_lik[i]= weibull\_lpdf(time[i] | kappa, nu[i]); }

else if (censored[i] == 1) { log\_lik[i]= weibull\_lccdf(time[i] | kappa, nu[i]); } }}

"

**# Compilation**

smweiAFT <- stan\_model(model\_code=weibAFT.stan)

**# Data**

D=list(n=1601,p=8,time=time,censored=censored,age=age,trt=trt,

gender=gender,marstat=marstat,hltst3= hltst3, hltst4= hltst4, hltst5= hltst5)

**# Estimation**

fitweiAFT <- sampling(smweiAFT,data =D,iter = 1500,warmup=250,chains = 2,seed= 12345)

summary(fitweiAFT, pars = c("beta", "kappa", "sigma"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

LOO3=loo(as.matrix(fitweiAFT,pars="log\_lik"))