setwd("C:/R files BHMRA")

data = read.table("DS\_11\_3.txt",header=T)

library(rstan)

library(loo)

attach(data)

**#**

**# Log-logistic**

**#**

D= list(n=1508, p=7,d=d,t=t,educ=educ,sibs=sibs,white=white,immig=immig,

lowinc=lowinc,city=city)

loglogist.stan ="

functions{

real loglogist\_lpdf(real t, real kappa, real lambda) {

return( log(kappa)+log(lambda)+(kappa-1)\*log(t)

-2\*log(1+lambda\*t^kappa) );}

real loglogist\_S\_lpdf(real t, real kappa, real lambda) {

return( log( 1/(1+lambda\*t^kappa) ));}

}

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

vector[n] t; // response

int<lower=0,upper=1> d[n]; // event indicator(1=occurred, 0=censored)

int<lower=0> p; // total regression parameters, incl. intercept

int<lower=0> educ[n];

int<lower=0> sibs[n];

int<lower=0> white[n];

int<lower=0> immig[n];

int<lower=0> lowinc[n];

int<lower=0> city[n];

}

parameters {vector[p] beta;

real<lower=1> kappa; // shape parameter

}

transformed parameters {

real eta[n];

real lambda[n];

real lambdaT;

real modeT; // modal age first maternity (13 years education, 3 siblings, white)

lambdaT = exp(beta[1]+beta[2]\*13+beta[3]\*3+beta[4]);

modeT = ((kappa-1)/lambdaT)^(1/kappa);

for (i in 1:n) { eta[i]= beta[1]+beta[2]\*educ[i]+beta[3]\*sibs[i]

+beta[4]\*white[i]+beta[5]\*immig[i]

+beta[6]\*lowinc[i]+beta[7]\*city[i];

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

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

for (i in 1:n) {

if (d[i] == 1) { target += loglogist\_lpdf(t[i]|kappa, lambda[i]); }

else if (d[i] == 0) { target += loglogist\_S\_lpdf(t[i]|kappa, lambda[i]); }}}

generated quantities{real log\_lik[n];

for (i in 1:n) {

if (d[i] == 1) { log\_lik[i]= loglogist\_lpdf(t[i] | kappa, lambda[i]); }

else if (d[i] == 0) { log\_lik[i]= loglogist\_S\_lpdf(t[i] | kappa, lambda[i]); }}}

"

**# Compilation**

smLL <- stan\_model(model\_code=loglogist.stan)

**# Estimation**

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

summaryLL <- summary(fitLL, pars = c("beta", "kappa", "modeT"), probs = c(0.025,0.05, 0.95, 0.975))$summary

# Fit

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

**#**

**# Model with Cure Fraction**

**#**

D= list(n=1508, p=7,d=d,t=t,educ=educ,sibs=sibs,white=white,immig=immig,

lowinc=lowinc,city=city)

loglogistCF.stan ="

functions{

real loglogistCF\_lpdf(real t, real kappa, real lambda, real theta) {

return( log(theta)+log(kappa)+log(lambda)+(kappa-1)\*log(t)

-2\*log(1+lambda\*t^kappa)-theta\*(1-1/(1+lambda\*t^kappa)) );}

real loglogistCF\_S\_lpdf(real t, real kappa, real lambda, real theta) {

return( -theta\*(1-1/(1+lambda\*t^kappa)) );}

}

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

vector[n] t; // response

int<lower=0,upper=1> d[n]; // event indicator(1=occurred, 0=censored)

int<lower=0> p; // total regression parameters, incl. intercept

int<lower=0> educ[n];

int<lower=0> sibs[n];

int<lower=0> white[n];

int<lower=0> immig[n];

int<lower=0> lowinc[n];

int<lower=0> city[n];

}

parameters {vector[p] beta;

real<lower=1> kappa; // shape parameter

real<lower=0> theta; // cure fraction parameter

}

transformed parameters {

real eta[n];

real lambda[n];

real lambdaT;

real p\_nochild;

real modeT; // modal age first maternity (13 years education, 3 siblings, white)

p\_nochild = exp(-theta); // rate of childlessness

lambdaT = exp(beta[1]+beta[2]\*13+beta[3]\*3+beta[4]);

modeT = ((kappa-1)/lambdaT)^(1/kappa);

for (i in 1:n) { eta[i]= beta[1]+beta[2]\*educ[i]+beta[3]\*sibs[i]

+beta[4]\*white[i]+beta[5]\*immig[i]

+beta[6]\*lowinc[i]+beta[7]\*city[i];

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

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

target += gamma\_lpdf(theta | 0.01, 0.01);

for (i in 1:n) {

if (d[i] == 1) { target += loglogistCF\_lpdf(t[i]|kappa, lambda[i],theta); }

else if (d[i] == 0) { target += loglogistCF\_S\_lpdf(t[i]|kappa, lambda[i],theta); }}}

generated quantities{real log\_lik[n];

for (i in 1:n) {

if (d[i] == 1) { log\_lik[i]= loglogistCF\_lpdf(t[i] | kappa, lambda[i],theta); }

else if (d[i] == 0) { log\_lik[i]= loglogistCF\_S\_lpdf(t[i] | kappa, lambda[i],theta); }}}

"

**# Compilation**

smLLCF <- stan\_model(model\_code=loglogistCF.stan)

**# Estimation**

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

summaryLLCF <- summary(fitLLCF, pars = c("beta", "kappa", "modeT","p\_nochild"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

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

**#**

**# Model with Cure Fraction and Frailty**

**#**

D= list(n=1508, p=7,d=d,t=t,educ=educ-mean(educ),sibs=sibs-mean(sibs),white=white,immig=immig,lowinc=lowinc,city=city)

loglogistCFF.stan ="

functions{

real loglogistCF\_lpdf(real t, real kappa, real lambda, real theta) {

return( log(theta)+log(kappa)+log(lambda)+(kappa-1)\*log(t)

-2\*log(1+lambda\*t^kappa)-theta\*(1-1/(1+lambda\*t^kappa)) );}

real loglogistCF\_S\_lpdf(real t, real kappa, real lambda, real theta) {

return( -theta\*(1-1/(1+lambda\*t^kappa)) );}

}

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

vector[n] t; // response

int<lower=0,upper=1> d[n]; // event indicator(1=occurred, 0=censored)

int<lower=0> p; // total regression parameters, incl. intercept

real educ[n];

real sibs[n];

int<lower=0> white[n];

int<lower=0> immig[n];

int<lower=0> lowinc[n];

int<lower=0> city[n];

}

parameters {vector[p] beta;

real<lower=1> kappa; // shape parameter

real<lower=0> theta; // cure fraction parameter

real u[n]; // frailty random effects

real<lower=0.001> sigma\_u; // sd frailty

}

transformed parameters {

real eta[n];

real lambda[n];

real lambdaT;

real p\_nochild;

real modeT; // modal age first maternity (13 years education, 3 siblings, white)

p\_nochild = exp(-theta); // rate of childlessness

lambdaT = exp(beta[1]+beta[2]\*(-0.35)+beta[3]\*(-0.66)+beta[4]);

modeT = ((kappa-1)/lambdaT)^(1/kappa);

for (i in 1:n) { eta[i]= beta[1]+beta[2]\*educ[i]+beta[3]\*sibs[i]

+beta[4]\*white[i]+beta[5]\*immig[i]

+beta[6]\*lowinc[i]+beta[7]\*city[i]+u[i]-u[1];

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

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

target += gamma\_lpdf(theta | 0.01, 0.01);

target += normal\_lpdf(sigma\_u |0, 1);

for (i in 1:n) { target += normal\_lpdf(u[i]| 0, sigma\_u);

if (d[i] == 1) { target += loglogistCF\_lpdf(t[i]|kappa, lambda[i],theta); }

else if (d[i] == 0) { target += loglogistCF\_S\_lpdf(t[i]|kappa, lambda[i],theta); }}}

generated quantities{real log\_lik[n];

for (i in 1:n) {

if (d[i] == 1) { log\_lik[i]= loglogistCF\_lpdf(t[i] | kappa, lambda[i],theta); }

else if (d[i] == 0) { log\_lik[i]= loglogistCF\_S\_lpdf(t[i] | kappa, lambda[i],theta); }}}

"

**# Compilation**

smLLCFF <- stan\_model(model\_code=loglogistCFF.stan)

**# Estimation**

inits1=list(beta=c(-15,0,0,0,0,0,0),kappa=3,theta=2.5,sigma\_u=0.1)

inits2=list(beta=c(-16,0,0,0,0,0,0),kappa=3.5,theta=2,sigma\_u=0.2)

inits= list(inits1,inits2)

fitLLCFF <- sampling(smLLCFF,data=D,iter=1500,warmup=250,chains=2,

seed= 12345)

summary(fitLLCFF, pars = c("beta", "kappa","theta","modeT",

"p\_nochild","sigma\_u"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# Fit**

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

**# random effects**

summaryLLCFF\_u <- summary(fitLLCFF, pars = c("u"), probs = c(0.025,0.05, 0.95, 0.975))$summary

**# 10 most negative and positive random effects**

sort(summaryLLCFF\_u[,1])[1:10]

sort(summaryLLCFF\_u[,1])[1499:1508]