setwd("C:/R files BHMRA")

attach("DS\_8\_6.RData")

library(rstan)

library(loo)

**# Diffuse prior on predictor fixed effects**

prenatal.stan <- "

data {

int<lower=1> N; // total number of observations

int Y[N]; // response variable

int<lower=1> K; // number of population-level effects

matrix[N, K] X; // centered population-level design matrix

vector[K] X\_means; // column means of X before centering

// data for group-specific effects of comm

int<lower=1> J\_1[N];

int<lower=1> N\_1;

int<lower=1> K\_1;

vector[N] Z\_1;

// data for group-specific effects of momr

int<lower=1> J\_2[N];

int<lower=1> N\_2;

int<lower=1> K\_2;

vector[N] Z\_2;

}

transformed data {

}

parameters {

vector[K] beta; // fixed predictor effects

real t\_Intercept; // Shifted Intercept

real<lower=0> sd\_1; // group-specific standard deviation

vector[N\_1] z\_1; // standardised community-specific effects

real<lower=0> sd\_2; // group-specific standard deviation

vector[N\_2] z\_2; // standardised mother-specific effects

}

transformed parameters {

// group-specific effects

vector[N\_1] r\_1;

// group-specific effects

vector[N\_2] r\_2;

vector[N] eta;

r\_1 = sd\_1 \* (z\_1);

r\_2 = sd\_2 \* (z\_2);

// linear predictor

eta = X \* beta + t\_Intercept;

for (n in 1:N) {

eta[n] = eta[n] + r\_1[J\_1[n]] \* Z\_1[n] + r\_2[J\_2[n]] \* Z\_2[n];

}

}

model {

// prior specifications

beta ~ normal(0, 100);

sd\_1 ~ student\_t(3, 0, 10);

z\_1 ~ normal(0, 1);

sd\_2 ~ student\_t(3, 0, 10);

z\_2 ~ normal(0, 1);

// likelihood contribution

Y ~ bernoulli\_logit(eta);

}

generated quantities {

vector[N] log\_lik;

real b\_Intercept; // population-level intercept

b\_Intercept = t\_Intercept - dot\_product(X\_means, beta);

for (n in 1:N)

log\_lik[n] = bernoulli\_logit\_lpmf(Y[n] | eta[n]);

}"

**# Compile and Estimation**

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

fit <- sampling(sm, data =DS\_8\_6, pars = c("beta","log\_lik"), iter = 2000, chains = 2,seed= 12345)

log\_lik <- extract\_log\_lik(fit)

loo1 <- loo(log\_lik)

**# Horseshoe prior on predictors**

prenatalHS.stan <- "

data {

int<lower=1> N; // total observations

int Y[N]; // response variable

int<lower=1> K; // number of fixed effects predictors

matrix[N, K] X; // centered fixed effects predictor matrix

vector[K] X\_means; // column means of X

// data for community effects

int<lower=1> J\_1[N];

int<lower=1> N\_1;

int<lower=1> K\_1;

vector[N] Z\_1;

// data for mother effects

int<lower=1> J\_2[N];

int<lower=1> N\_2;

int<lower=1> K\_2;

vector[N] Z\_2;

}

transformed data {

}

parameters {

vector[K] beta; // fixed predictor effects

// horseshoe shrinkage parameters

vector<lower=0>[K] hs\_local;

real<lower=0> hs\_global;

real t\_Intercept; // shifted Intercept

real<lower=0> sd\_1; // group-specific standard deviation

vector[N\_1] z\_1; // standardised community-specific effects

real<lower=0> sd\_2; // group-specific standard deviation

vector[N\_2] z\_2; // standardised mother-specific effects

}

transformed parameters {

vector<lower=0>[K] kappa;

// community effects

vector[N\_1] r\_1;

// mother effects

vector[N\_2] r\_2;

vector[N] eta;

r\_1 = sd\_1 \* (z\_1);

r\_2 = sd\_2 \* (z\_2);

// measures of shrinkage

for (i in 1:K) kappa[i] = 1/(1+hs\_local[i]\*hs\_local[i]);

// linear predictor

eta = X \* beta + t\_Intercept;

for (n in 1:N) {

eta[n] = eta[n] + r\_1[J\_1[n]] \* Z\_1[n] + r\_2[J\_2[n]] \* Z\_2[n];

}

}

model {

// priors

hs\_local ~ student\_t(1, 0, 1);

hs\_global ~ cauchy(0, 1);

beta ~ normal(0, hs\_local \* hs\_global);

sd\_1 ~ student\_t(3, 0, 10);

z\_1 ~ normal(0, 1);

sd\_2 ~ student\_t(3, 0, 10);

z\_2 ~ normal(0, 1);

// likelihood contribution

Y ~ bernoulli\_logit(eta);

}

generated quantities {

vector[N] log\_lik;

real b\_Intercept; // population-level intercept

b\_Intercept = t\_Intercept - dot\_product(X\_means, beta);

for (n in 1:N)

log\_lik[n] = bernoulli\_logit\_lpmf(Y[n] | eta[n]);

}"

**# Compile and Estimation**

smHS <- stan\_model(model\_code=prenatalHS.stan)

fitHS <- sampling(smHS, data =DS\_8\_6, pars = c("beta","kappa","log\_lik"), iter = 2000,

chains = 2,seed= 12345)

**# Fit**

log\_lik <- extract\_log\_lik(fitHS)

loo2 <- loo(log\_lik)

**# Posterior means and medians, kappa statsitics**

kappaMCMC = as.matrix(fitHS,pars="kappa")

kappa.mn=apply(kappaMCMC,2,mean)

kappa.mdn=apply(kappaMCMC,2,quantile,0.5)