STA365HW5

Stephen Shannon

April 3, 2020

```
library(rstan)

## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.19.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend
calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

## For improved execution time, we recommend calling
## Sys.setenv(LOCAL_CPPFLAGS = '-march=native')
## although this causes Stan to throw an error on a few processors.

library(ggplot2)
library(dplyr)

library(bayesplot)

## Warning: package 'bayesplot' was built under R version 3.6.3

## This is bayesplot version 1.7.1

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting
```

Question 1) Generating 3 data sets with p = 10, p = 50, p = 100, with n = 100, sigma = 0.4.

```
n = 1:100
p = 10
beta_init = c(1,2,3)
sum = 0
count = 1:100
x10 <- list()
for (i in n){
  x_vec = c()
```

```r
  x_123 = c(cos(i), sin(i), tan(i))
  rnorm_list <- rnorm(p-3)
  x_vec = c(x_123, rnorm_list)
  x10[[i]] <- x_vec
}
remainder10 <- rep(0, 7)
beta10 <- c(beta_init, remainder10)

y10=c()
for (i in count){
  mean <- x10[[i]]%*%beta10
  y_sim <- rnorm(1, mean, 0.4)
  y10 <- c(y10, y_sim)
}


p = 50
x50 <- list()
for (i in n){
  x_vec = c()
  x_123 = c(cos(i), sin(i), tan(i))
  rnorm_list <- rnorm(p-3)
  x_vec = c(x_123, rnorm_list)
  x50[[i]] <- x_vec
}
remainder50 <- rep(0, 47)
beta50 <- c(beta_init, remainder50)

count = 1:100
y50=c()
for (i in count){
  mean <- x50[[i]]%*%beta50
  y_sim <- rnorm(1, mean, 0.4)
  y50 <- c(y50, y_sim)
}

p = 100
x100 <- list()
for (i in n){
  x_vec = c()
  x_123 = c(cos(i), sin(i), tan(i))
  rnorm_list <- rnorm(p-3)
  x_vec = c(x_123, rnorm_list)
  x100[[i]] <- x_vec
}
remainder100 <- rep(0, 97)
beta100 <- c(beta_init, remainder100)

count = 1:100
```

```r
y100=c()
for (i in count){
  mean <- x100[[i]]%*%beta100
  y_sim <- rnorm(1, mean, 0.4)
  y100 <- c(y100, y_sim)
}
```

Q2) Creating the stan model

For the prior on the shrinkage parameter, lambda, I chose Cauchy ~ (0,3), as this produces a heavy tailed distribution which is suitable for LASSO regression.

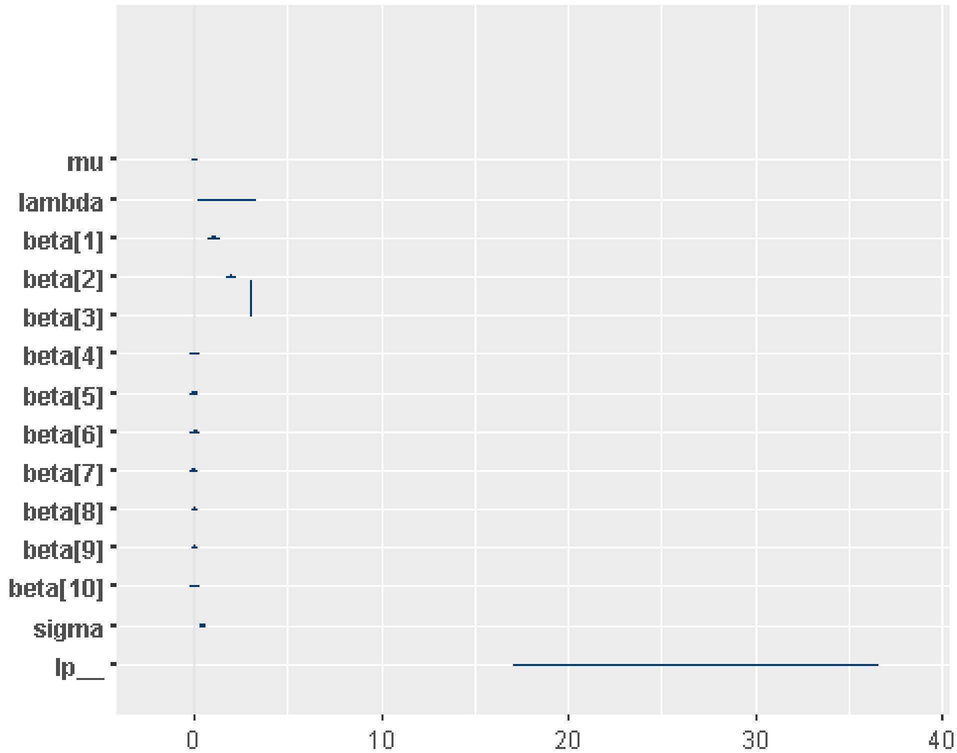
```stan
data {
  int<lower = 0> n; // number of observations
  int<lower = 0> p; // number of covariates
  matrix[n,p] x; // covariates are the rows!
  vector[n] y;
}

parameters {
  real mu;
  real<lower = 0> lambda;
  vector[p] beta;
  real<lower = 0> sigma;
}

model {
  y ~ normal(mu + x*beta, sigma);
  sigma ~ normal(0,1);
  for (i in 1:p) {
  beta[i] ~ double_exponential(mu, lambda);
  }
  mu ~ normal(0,1);
  lambda ~ cauchy(0,3);
}
```

P = 10 fit

```r
x_matrix <- t(sapply(x10, unlist))
stan_data <- list(n=100, p=10, x=x_matrix, y=y10)

fit <- sampling(homework5, data = stan_data)

##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
```

```
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.241 seconds (Warm-up)
## Chain 1:                0.087 seconds (Sampling)
## Chain 1:                0.328 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.216 seconds (Warm-up)
## Chain 2:                0.091 seconds (Sampling)
## Chain 2:                0.307 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
```

```
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.251 seconds (Warm-up)
## Chain 3:                0.117 seconds (Sampling)
## Chain 3:                0.368 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.24 seconds (Warm-up)
## Chain 4:                0.085 seconds (Sampling)
## Chain 4:                0.325 seconds (Total)
## Chain 4:

posterior <- as.matrix(fit)
plot_title <- ggtitle("Posterior distribution of Beta_j, p = 10")
mcmc_areas(posterior)
```

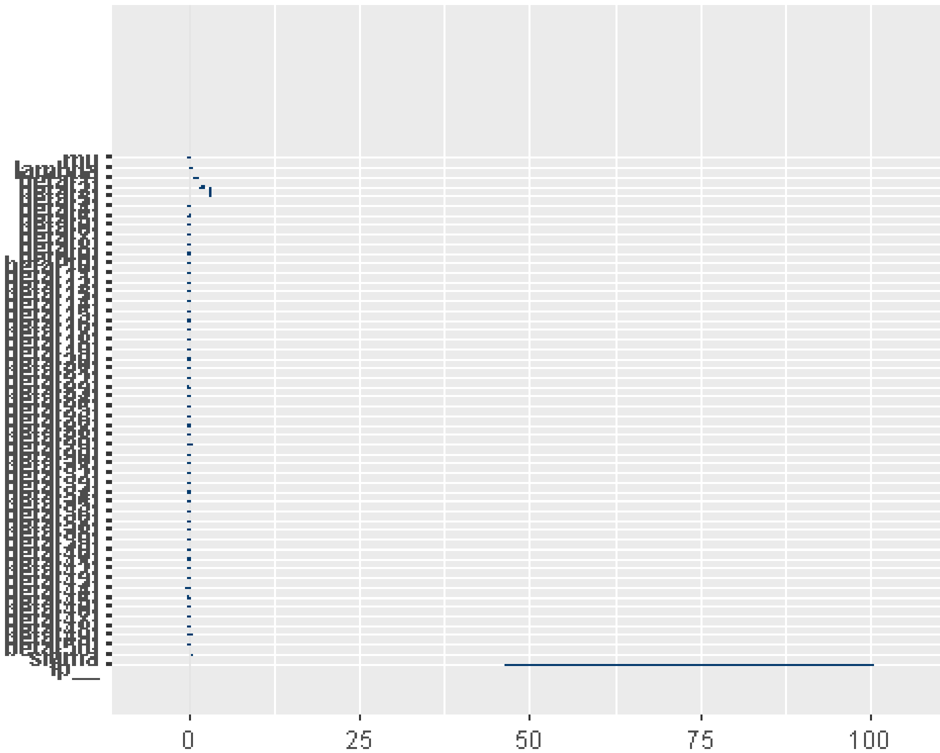


P = 50 fit

```
x_matrix <- t(sapply(x50, unlist))
stan_data <- list(n=100, p=50, x=x_matrix, y=y50)

fit <- sampling(homework5, data = stan_data)

##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.723 seconds (Warm-up)
## Chain 1:                0.419 seconds (Sampling)
## Chain 1:                1.142 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.73 seconds (Warm-up)
## Chain 2:                0.402 seconds (Sampling)
## Chain 2:                1.132 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.747 seconds (Warm-up)
## Chain 3:                0.448 seconds (Sampling)
## Chain 3:                1.195 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.8 seconds (Warm-up)
## Chain 4:                0.427 seconds (Sampling)
## Chain 4:                1.227 seconds (Total)
## Chain 4:

posterior <- as.matrix(fit)
plot_title <- ggtitle("Posterior distribution of Beta_j, p = 50")
mcmc_areas(posterior)
```



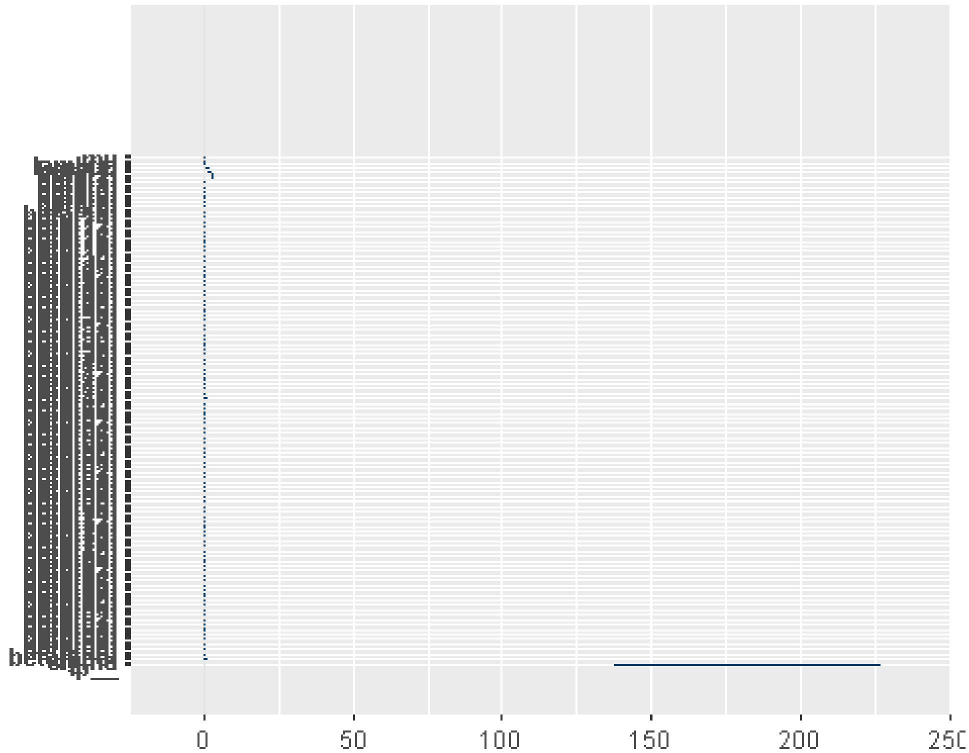
P = 100 fit

```
x_matrix <- t(sapply(x100, unlist))
stan_data <- list(n=100, p=100, x=x_matrix, y=y100)

fit <- sampling(homework5, data = stan_data)

##
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1: 
## Chain 1:  Elapsed Time: 2.513 seconds (Warm-up)
## Chain 1:                1.99 seconds (Sampling)
## Chain 1:                4.503 seconds (Total)
## Chain 1: 
## 
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 2).
## Chain 2: 
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would 
take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2: 
## Chain 2: 
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2: 
## Chain 2:  Elapsed Time: 2.694 seconds (Warm-up)
## Chain 2:                2.266 seconds (Sampling)
## Chain 2:                4.96 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would 
take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 2.589 seconds (Warm-up)
## Chain 3:                1.737 seconds (Sampling)
## Chain 3:                4.326 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'd2dc3b6e1a4db6f743b8418bcc8efc26' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 2.619 seconds (Warm-up)
## Chain 4:                2.214 seconds (Sampling)
## Chain 4:                4.833 seconds (Total)
## Chain 4: 
```

```r
posterior <- as.matrix(fit)
plot_title <- ggtitle("Posterior distribution of Beta_j, p = 100")
mcmc_areas(posterior)
```



In all 3 fits, the Bayesian LASSO does a pretty good job of sending almost all the beta's to 0. The Cauchy ~ (0,3) prior has shrunk all the 'zero' parameters to at least |-0.11|. The Cauchy prior correctly estimates beta 3 to be 3 in all 3 fits. However, it slightly underestimates beta 2 in all 3 models, estimating it to be about 1.9. In model p = 10, beta 1 is overestimated to be 1.09, but in models p = 50, 100, beta 1 is underestimated at 0.8. Overall, the Cauchy prior may have overshrunk the parameters, but it has succesfully sent most of the 0 parameters to 0.