Multiple linear regression for the HOMES data

Chapter 4.2: Linear Regression

We use the data from Barberan (2015), downloaded from http://figshare.com/articles/1000homes/1270900. The data are dust samples from the ledges above doorways from $n = 1,059$ homes (after removing samples with missing data) in the continental US. Bioinformatics processing detects the presence or absence of 763 species (technically operational taxonomic units) of fungi. The response is the log of the number of fungi species present in the sample, which is a measure of species richness. The objective is to determine which factors influence a home's species richness. For each home, eight covariates are included in this example: longitude, latitude, annual mean temperature, annual mean precipitation, net primary productivity (NPP), elevation, the binary indicator that the house is a single-family home, and the number of bedrooms in the home. These covariates are all centered and scaled to have mean zero and variance one.

The Bayesian multiple linear regression model is

$$Y_i \sim \text{Normal}(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j, \sigma^2).$$

Below we compare three different prior for the slopes $\beta_1, \ldots, \beta_p$,

1. Uninformative Gaussian: $\beta_j \sim \text{Normal}(0, 1000)$
2. Gaussian shrinkage: $\beta_j \sim \text{Normal}(0, \sigma_b^2)$ with $\sigma_b^2 \sim \text{InvGamma}(0.1, 0.1)$
3. Bayesian LASSO: $\beta_j \sim \text{DE}(0, \sigma_b^2)$ with $\sigma_b^2 \sim \text{InvGamma}(0.1, 0.1)$

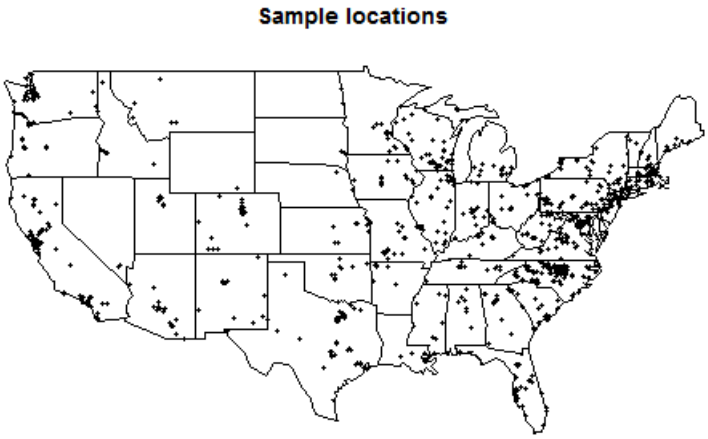
In all cases, we use uninformative conjugate priors for the intercept $\beta_0 \sim \text{Normal}(0, 1000)$ and variance $\sigma^2 \sim \text{InvGamma}(0.1, 0.1)$.

Load and plot the data

```
library(rjags)

 # Load data

 load("S:\\Documents\\My Papers\\BayesBook\\Data\\Microbiome\\homes.RData")

 lat      <- homes[,4]
 long     <- homes[,5]
 temp     <- homes[,6]
 precip   <- homes[,7]
 NPP      <- homes[,8]
 elev     <- homes[,9]
 house    <- ifelse(homes[,10]=="One-family house detached from any other house",1,0)
 bedrooms <- as.numeric(homes[,11])
```

```
## Warning: NAs introduced by coercion
```

```
 city     <- homes[,2]
 state    <- homes[,3]

 OTU      <- as.matrix(OTU)
 nspecies <- rowSums(OTU>0)
 Y        <- log(nspecies)
 X        <- cbind(long,lat,temp,precip,NPP,elev,house,bedrooms)
 names    <- c("Longitude","Latitude",
               "Temperature","Precipitation","NPP",
               "Elevation","Single-family home",
               "Number of bedrooms")

# Remove observations with missing data

 junk     <- is.na(rowSums(X))
 Y        <- Y[!junk]
 X        <- X[!junk,]
 city     <- city[!junk]
 state    <- state[!junk]

# Standardize the covariates

 X        <- as.matrix(scale(X))

# Plot the sample locations

library(maps)
map("state")
points(homes[,5],homes[,4],pch=19,cex=.5)
title("Sample locations")
```

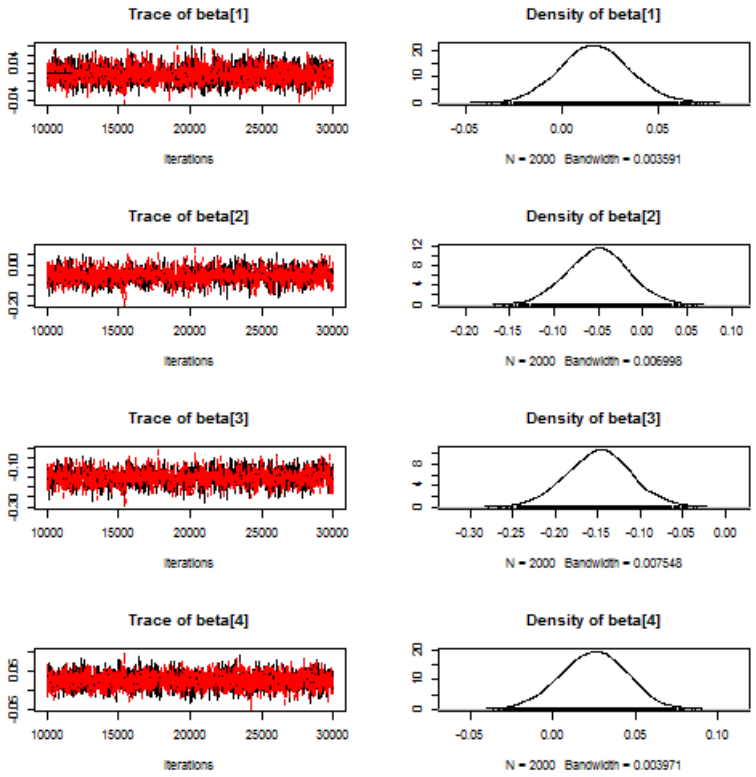


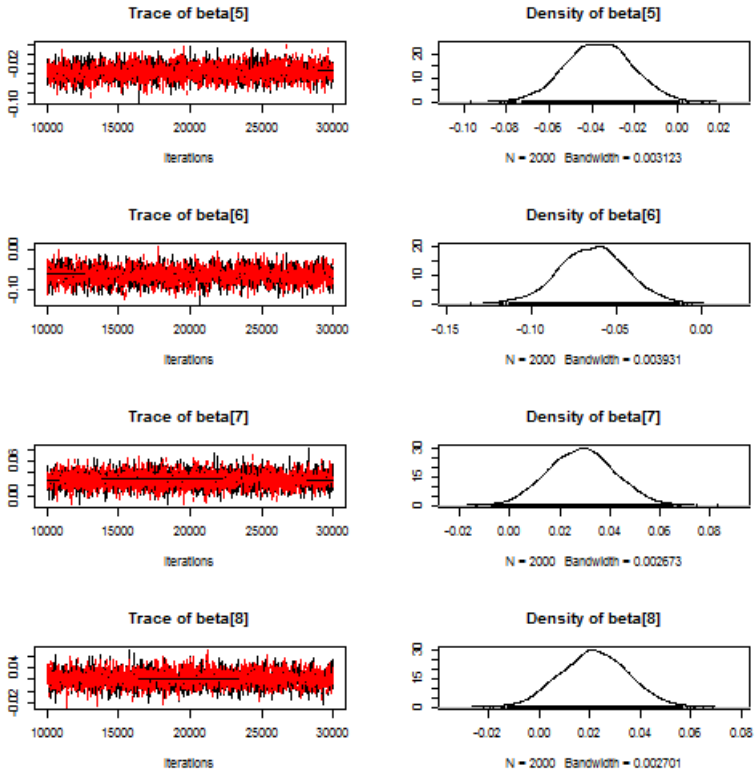
Put the data in JAGS format

```
n         <- length(Y)
p         <- ncol(X)

data    <- list(Y=Y,X=X,n=n,p=p)
params <- c("beta")

burn      <- 10000
n.iter    <- 20000
thin      <- 10
n.chains <- 2
```

(1) Fit the uninformative Gaussian model

```
model_string <- textConnection("model{
  # Likelihood
   for(i in 1:n){
     Y[i] ~ dnorm(alpha+inprod(X[i,],beta[]),taue)
   }
  # Priors
   for(j in 1:p){
     beta[j] ~ dnorm(0,0.001)
   }
   alpha ~ dnorm(0,0.001)
   taue  ~ dgamma(0.1, 0.1)
}")

model <- jags.model(model_string,data = data, n.chains=n.chains,quiet=TRUE)
update(model, burn, progress.bar="none")
samples1 <- coda.samples(model, variable.names=params, thin=thin, n.iter=n.iter, progress.bar="none")

plot(samples1)
```






```
round(effectiveSize(samples1),1)
```

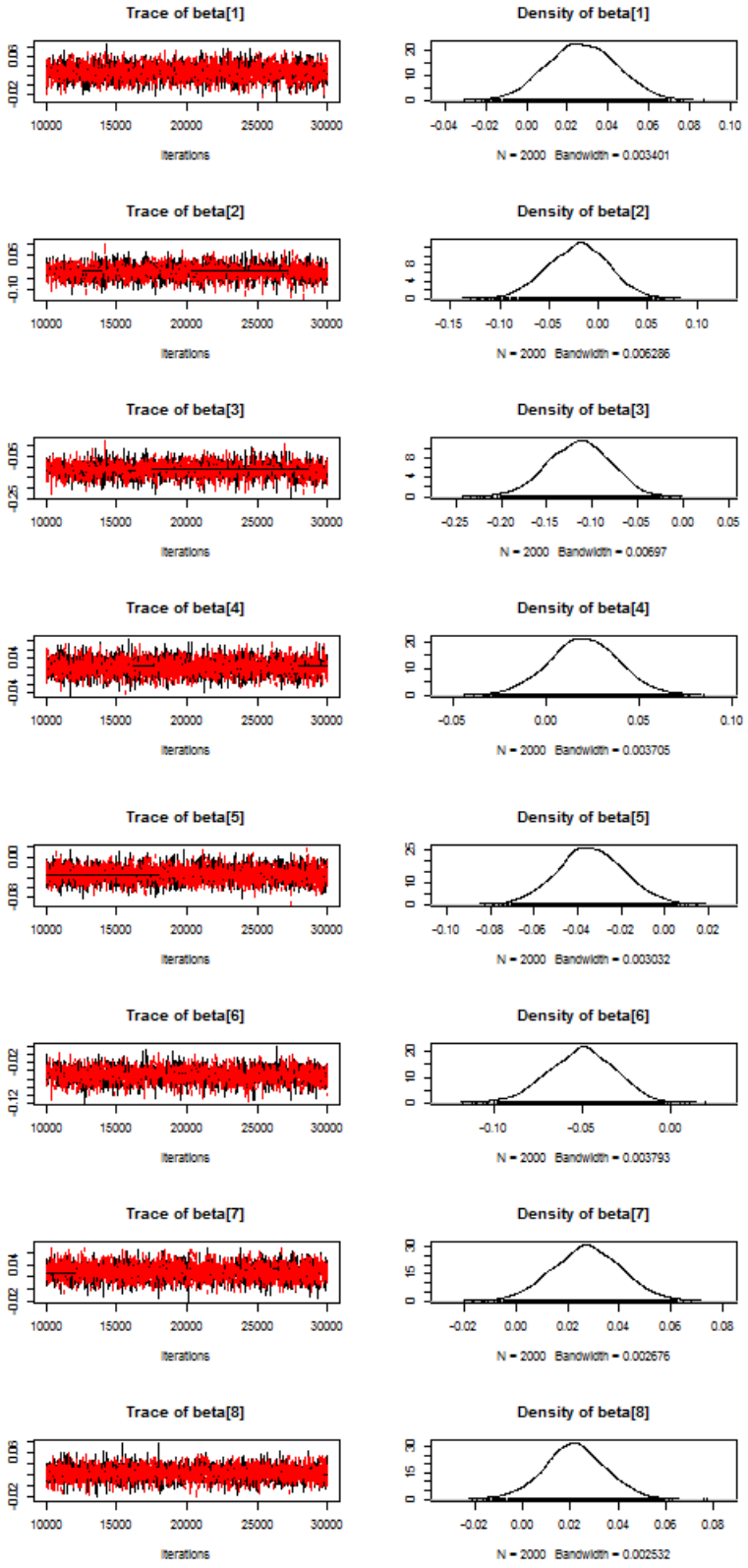
```
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]
##  2699.7  1675.6  1710.5  3827.0  4000.0  2217.1  4000.0  3853.3
```

```
sum                     <- summary(samples1)
rownames(sum$statistics) <- names
rownames(sum$quantiles)  <- names
sum$statistics           <- round(sum$statistics,3)
sum$quantiles            <- round(sum$quantiles,3)
sum
```

```
##
## Iterations = 10010:30000
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean    SD Naive SE Time-series SE
## Longitude            0.017 0.018    0.000          0.000
## Latitude            -0.050 0.035    0.001          0.001
## Temperature         -0.149 0.038    0.001          0.001
## Precipitation        0.025 0.020    0.000          0.000
## NPP                 -0.037 0.015    0.000          0.000
## Elevation           -0.064 0.019    0.000          0.000
## Single-family home   0.029 0.013    0.000          0.000
## Number of bedrooms   0.022 0.013    0.000          0.000
##
## 2. Quantiles for each variable:
##
##                        2.5%    25%    50%    75%  97.5%
## Longitude            -0.018  0.005  0.017  0.029  0.054
## Latitude             -0.119 -0.074 -0.050 -0.027  0.019
## Temperature          -0.225 -0.174 -0.148 -0.124 -0.073
## Precipitation        -0.015  0.012  0.025  0.039  0.063
## NPP                  -0.067 -0.047 -0.037 -0.026 -0.006
## Elevation            -0.103 -0.077 -0.064 -0.051 -0.025
## Single-family home    0.004  0.020  0.029  0.037  0.055
## Number of bedrooms   -0.004  0.013  0.022  0.031  0.048
```

(2) Fit the Gaussian shrinkage model

```
model_string <- textConnection("model{
  # Likelihood
   for(i in 1:n){
     Y[i] ~ dnorm(alpha+inprod(X[i,],beta[]),taue)
   }
  # Priors
   for(j in 1:p){
     beta[j] ~ dnorm(0,taue*taub)
   }
   alpha ~ dnorm(0,0.001)
   taue  ~ dgamma(0.1, 0.1)
   taub  ~ dgamma(0.1, 0.1)
}")

model <- jags.model(model_string,data = data, n.chains=n.chains,quiet=TRUE)
update(model, burn, progress.bar="none")
samples2 <- coda.samples(model, variable.names=params, thin=thin, n.iter=n.iter, progress.bar="none")

plot(samples2)
```




```
round(effectiveSize(samples2),1)
```

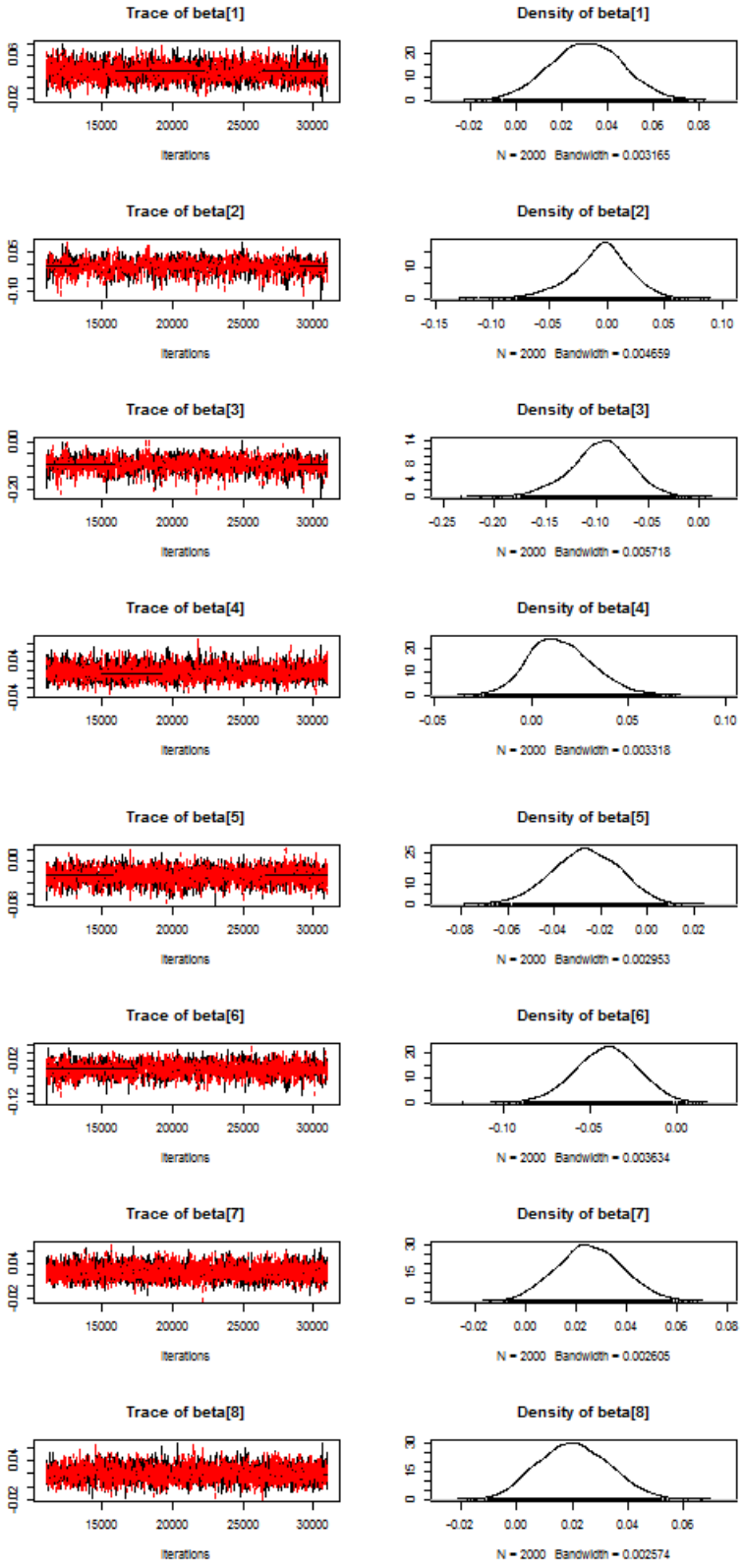
```
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]
##  2905.1  1933.7  1886.2  3645.9  3791.3  2577.3  3689.1  4135.9
```

```
sum                     <- summary(samples2)
rownames(sum$statistics) <- names
rownames(sum$quantiles)  <- names
sum$statistics           <- round(sum$statistics,3)
sum$quantiles            <- round(sum$quantiles,3)
sum
```

```
##
## Iterations = 10010:30000
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean    SD Naive SE Time-series SE
## Longitude            0.027 0.017    0.000          0.000
## Latitude            -0.021 0.031    0.000          0.001
## Temperature         -0.115 0.035    0.001          0.001
## Precipitation        0.020 0.018    0.000          0.000
## NPP                 -0.034 0.015    0.000          0.000
## Elevation           -0.050 0.019    0.000          0.000
## Single-family home   0.028 0.013    0.000          0.000
## Number of bedrooms   0.022 0.013    0.000          0.000
##
## 2. Quantiles for each variable:
##
##                        2.5%    25%    50%    75%  97.5%
## Longitude            -0.006  0.016  0.027  0.039  0.060
## Latitude             -0.083 -0.041 -0.020  0.001  0.039
## Temperature          -0.185 -0.138 -0.114 -0.092 -0.051
## Precipitation        -0.016  0.008  0.020  0.033  0.057
## NPP                  -0.063 -0.044 -0.034 -0.024 -0.004
## Elevation            -0.089 -0.063 -0.050 -0.037 -0.016
## Single-family home    0.002  0.019  0.028  0.037  0.054
## Number of bedrooms   -0.004  0.014  0.022  0.030  0.047
```

(3) Fit the Bayesian LASSO model

```
model_string <- textConnection("model{
  # Likelihood
   for(i in 1:n){
     Y[i] ~ dnorm(alpha+inprod(X[i,],beta[]),taue)
   }
  # Priors
   for(j in 1:p){
     beta[j] ~ ddexp(0,taue*taub)
   }
   alpha ~ dnorm(0,0.001)
   taue  ~ dgamma(0.1, 0.1)
   taub  ~ dgamma(0.1, 0.1)
}")

model <- jags.model(model_string,data = data, n.chains=n.chains,quiet=TRUE)
update(model, burn, progress.bar="none")
samples3 <- coda.samples(model, variable.names=params, thin=thin, n.iter=n.iter, progress.bar="none")

plot(samples3)
```




```
round(effectiveSize(samples3),1)
```

```
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]
##  2493.6  1452.3  1363.4  3988.4  3857.0  2107.0  4000.0  4000.0
```

```
sum                     <- summary(samples3)
rownames(sum$statistics) <- names
rownames(sum$quantiles)  <- names
sum$statistics           <- round(sum$statistics,3)
sum$quantiles            <- round(sum$quantiles,3)
sum
```

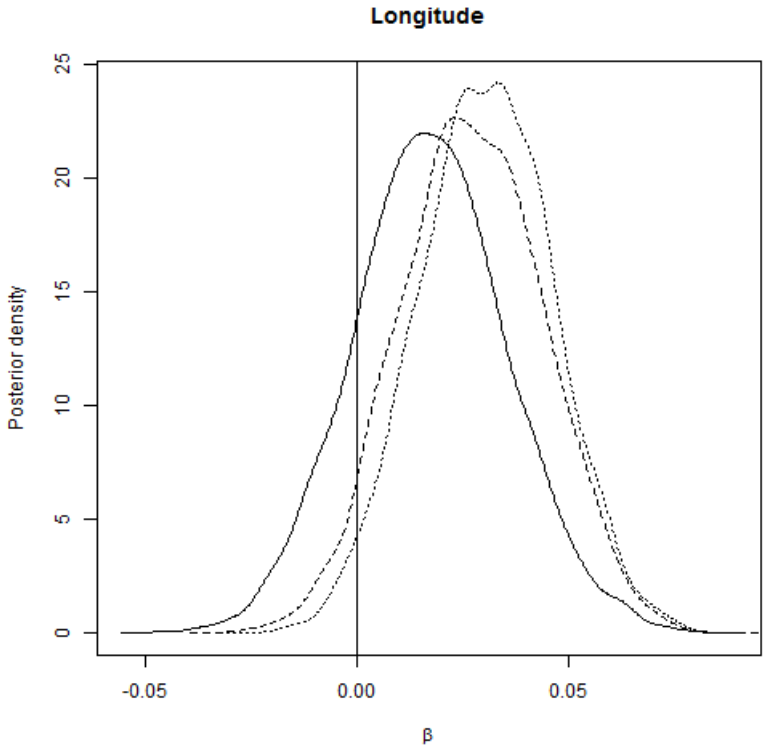
```
##
## Iterations = 11010:31000
## Thinning interval = 10
## Number of chains = 2
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean    SD Naive SE Time-series SE
## Longitude            0.030 0.016        0          0.000
## Latitude            -0.006 0.026        0          0.001
## Temperature         -0.096 0.030        0          0.001
## Precipitation        0.015 0.016        0          0.000
## NPP                 -0.026 0.015        0          0.000
## Elevation           -0.040 0.018        0          0.000
## Single-family home   0.026 0.013        0          0.000
## Number of bedrooms   0.021 0.013        0          0.000
##
## 2. Quantiles for each variable:
##
##                        2.5%    25%    50%    75%  97.5%
## Longitude             0.000  0.020  0.031  0.041  0.061
## Latitude             -0.063 -0.021 -0.004  0.010  0.042
## Temperature          -0.160 -0.114 -0.095 -0.076 -0.040
## Precipitation        -0.014  0.004  0.014  0.026  0.049
## NPP                  -0.056 -0.036 -0.026 -0.016  0.001
## Elevation            -0.077 -0.052 -0.040 -0.028 -0.006
## Single-family home    0.001  0.017  0.025  0.034  0.051
## Number of bedrooms   -0.003  0.012  0.020  0.029  0.046
```

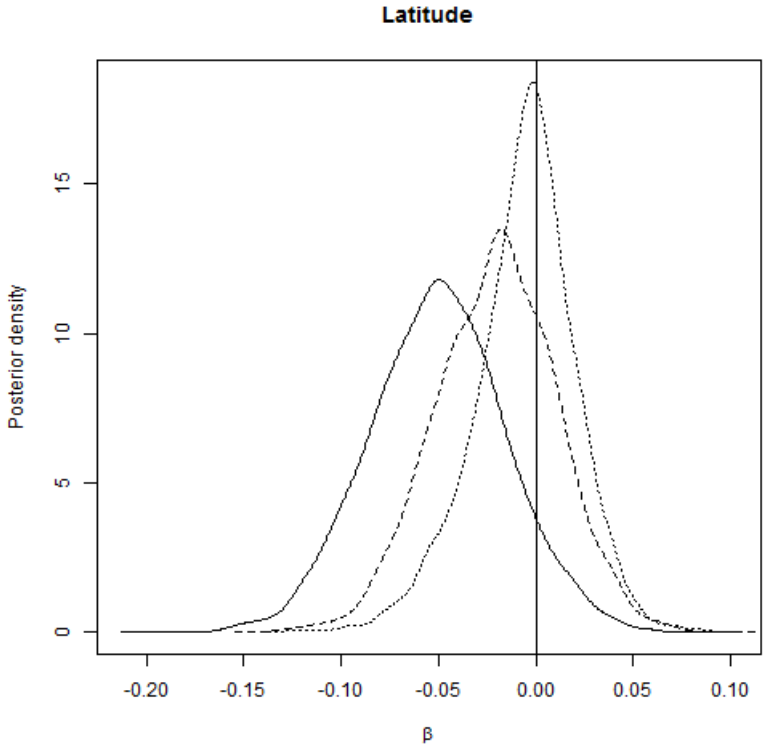
Compare the three fits

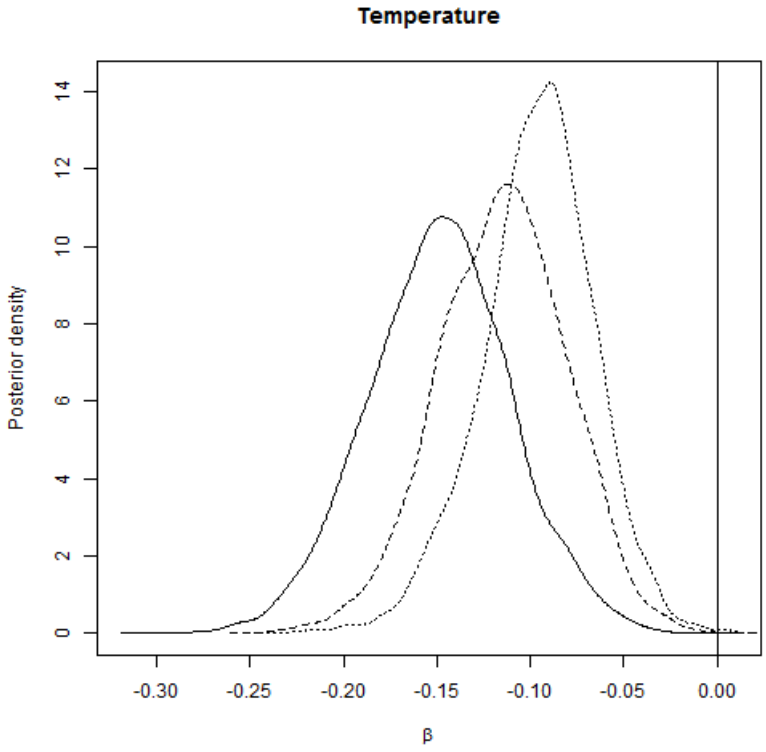
The plots below show the posterior for each covariate's slope for the three models:

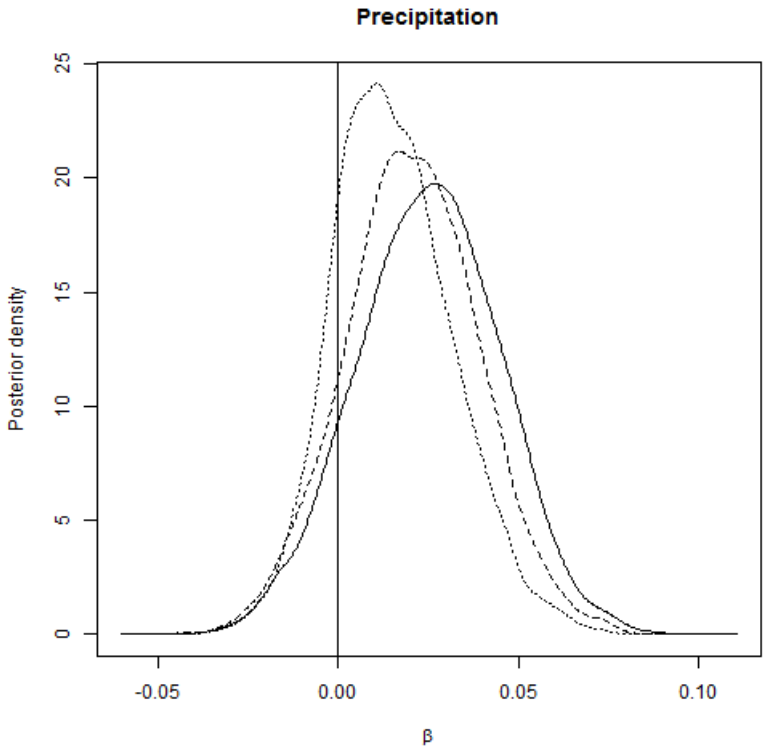
1. Uninformative Gaussian is the solid line
2. Gaussian shrinkage is the dotted line
3. Bayesian LASSO is the dashed line

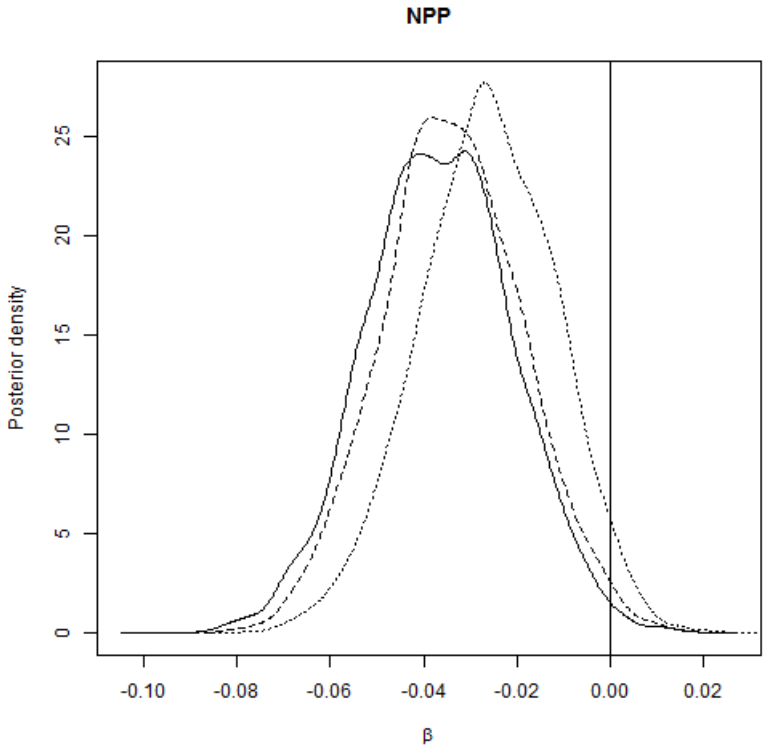
```
for(j in 1:p){

 # Collect the MCMC iteration from both chains for the three priors

 s1 <- c(samples1[[1]][,j],samples1[[2]][,j])
 s2 <- c(samples2[[1]][,j],samples2[[2]][,j])
 s3 <- c(samples3[[1]][,j],samples3[[2]][,j])

 # Get smooth density estimate for each prior

 d1 <- density(s1)
 d2 <- density(s2)
 d3 <- density(s3)

 # Plot the density estimates

 mx <- max(c(d1$y,d2$y,d3$y))

 plot(d1$x,d1$y,type="l",ylim=c(0,mx),xlab=expression(beta),ylab="Posterior density",main=names[j])
 lines(d2$x,d2$y,lty=2)
 lines(d3$x,d3$y,lty=3)
 abline(v=0)
}
```

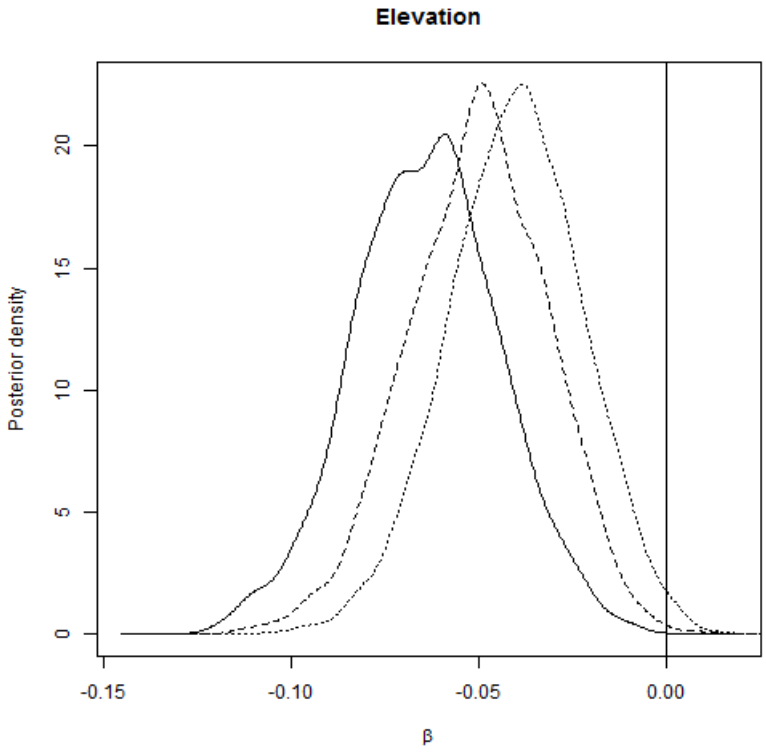

Longitude
Posterior density
0
5
10
15
20
25
-0.05
0.00
0.05
β

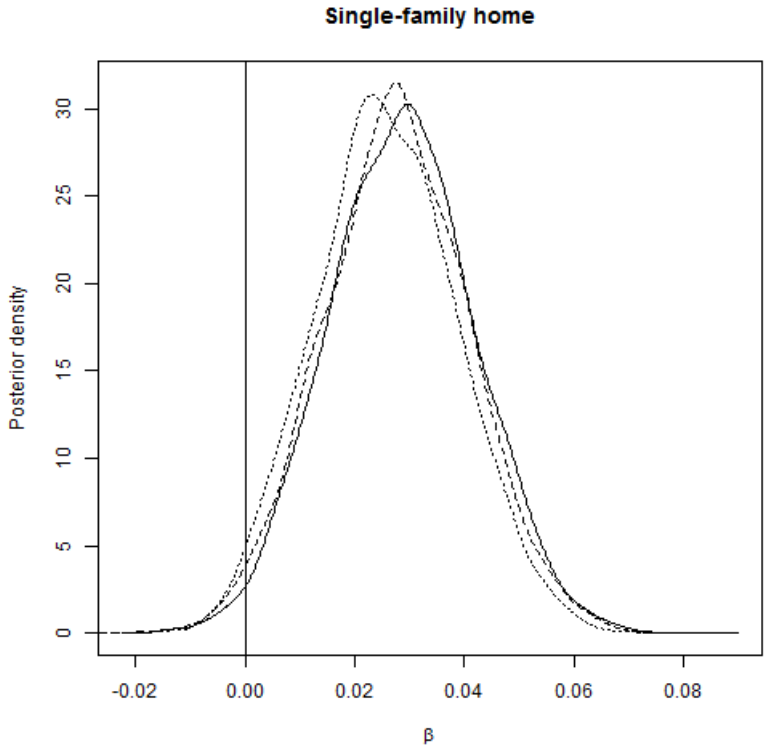

Latitude
Posterior density
0
5
10
15
-0.20
-0.15
-0.10
-0.05
0.00
0.05
0.10
β

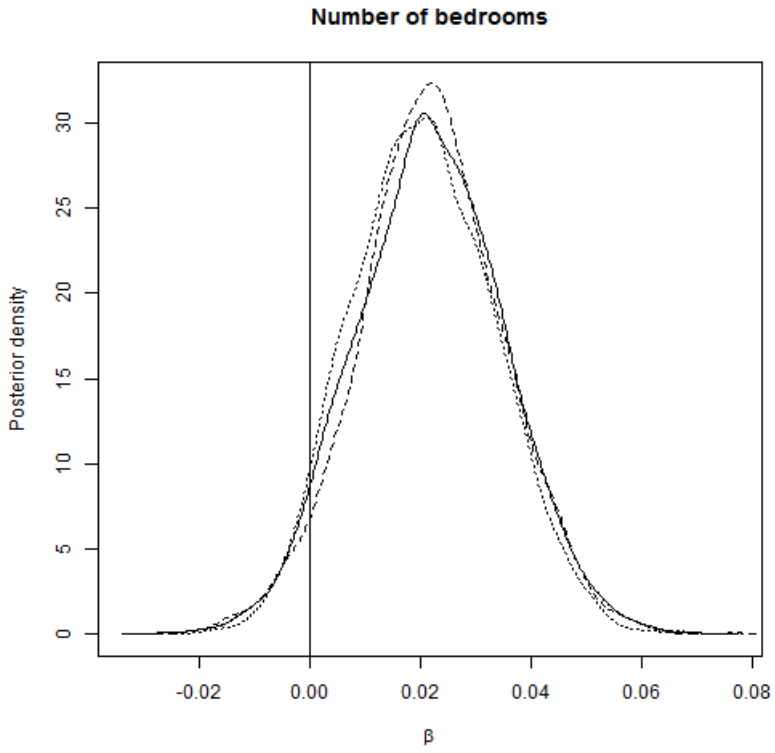

Temperature
Posterior density
0
2
4
6
8
10
12
14
-0.30
-0.25
-0.20
-0.15
-0.10
-0.05
0.00
β


Precipitation
Posterior density
0
5
10
15
20
25
-0.05
0.00
0.05
0.10
β


NPP
Posterior density
0
5
10
15
20
25
-0.10
-0.08
-0.06
-0.04
-0.02
0.00
0.02
β


Elevation
Posterior density
0
5
10
15
20
-0.15
-0.10
-0.05
0.00
β





Since n is so much bigger than p, the prior has little effect on the posterior. In all three models temperature, NPP, elevation, and single-family home are the most important predictors.