STAT 460
Bayesian Statistics
Final Project

Steve Hof

28/11/2020

The data from this project was supplied by the course instructor, Dr. Michelle Miranda. We first read in the data, then use it in the second half of the project.

Part I

Consider the multiple linear regression model:

$$y = \boldsymbol{X}\beta + \boldsymbol{E}$$

where y is a vector of size n containing the response variable, $\boldsymbol{X}$ is a matrix of size $n \times J$ of fixed covariates, and β is a vector of size J containing the coefficients that characterize the linear relationship between y and X. Let $\boldsymbol{E}$ be a vector of of size n of random noise terms. We assume $\boldsymbol{E} \sim N_n(0, \Sigma)$, with known $\Sigma = I_n$. Now assume that for each $j = 1, \ldots, J$

$$\begin{gathered} \beta_j \mid \delta_j, \tau, \epsilon \sim \delta_j N\left(0, \tau^2\right) + (1 - \delta_j) N(0, \epsilon) \\ \delta_j \mid \pi \sim \text{Bernoulli}(\pi) \\ \pi \mid a_\pi, b_\pi \sim \text{Beta}\left(\tfrac{a_\pi}{2}, \tfrac{b_\pi}{2}\right) \end{gathered}$$

Let $\theta = (\beta, \delta, \pi)$, then the prior distribution of θ is $p(\theta) = p\left(\pi \mid a_\pi, b_\pi\right) \prod_{j=1}^{J} p\left(\beta_j \mid \delta_j, \tau^2, \epsilon\right) p\left(\delta_j \mid \pi\right)$

Question (a)

Write down $p(\beta_j \mid \delta_j, \tau^2, \epsilon)$, the prior of β_j, up to a constant of proportionality.

solution:

$$\begin{aligned} p(\beta_j \mid \delta_j, \tau^2, \epsilon) &= \left(\frac{1}{\sqrt{2\pi\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\beta_j^2\right\}\right)^{\delta_j} \left(\frac{1}{\sqrt{2\pi\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\beta_j^2\right\}\right)^{(1-\delta_j)} \\ &\propto \left(\frac{1}{\tau} \exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}\right)^{\delta_j} \left(\frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}\right)^{(1-\delta_j)} \end{aligned}$$

Question (b)

Use (a) to find the full conditional distribution of β_j, i.e., $p\left(\beta_j \mid \delta_j, \tau^2, \epsilon, y\right)$.

Hint 1: consider two separate distributions, $p\left(\beta_j \mid \delta_j = 0, \tau^2, \varepsilon, y\right)$ and $p\left(\beta_j \mid \delta_j = 1, \tau^2, \varepsilon, y\right)$.

Hint 2: If it helps, use the fact that $y_i - \sum_{j=1}^{J} X_{ij}\beta_j = \tilde{y}_i - X_{ij}\beta_j$, where $\tilde{y}_i = y_i - \sum_{l \neq j} X_{il}\beta_l$

solution:

We start by using hint 1.

$$
\begin{aligned}
p(\beta_j \mid \delta_j = 0, \tau^2, \varepsilon, y) &= p(\beta_j \mid \delta_j = 0, \tau^2, \varepsilon) \prod_{i=1}^{n} p(y_i|\beta_j) \\
&\propto \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\} \prod_{i=1}^{n} p(y_i|\beta_j)
\end{aligned}
$$

To keep things more organized we will calculate the likelihood $\prod_{i=1}^{n} p(y_i|\beta_j)$ separately now, then continue on by plugging it into the above.

$$
\begin{aligned}
\prod_{i=1}^{n} p(y_i|\beta_j) &= \prod_{i=1}^{n} \det(\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(y_i - \sum_{j=1}^{J} X_{ij}\beta_j\right)^2\right\} \\
&\propto \exp\left\{\sum_{i=1}^{n}\left(-\frac{1}{2}\right)\left(y_i - \sum_{j=1}^{J} X_{ij}\beta_j\right)^2\right\} \\
&\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\tilde{y} - X_{ij}\beta_j)^2\right\} \text{(Hint 2)}
\end{aligned}
$$

Now, plugging the likelihood back into the above, we have

$$
\begin{aligned}
p(\beta_j \mid \delta_j = 0, \tau^2, \varepsilon, y) &\propto \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n} (\tilde{y} - X_{ij}\beta_j)^2\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2}\left(\frac{\beta_j^2}{\varepsilon} + \sum_{i=1}^{n} (\tilde{y} - X_{ij}\beta_j)^2\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 + \varepsilon\sum_{i=1}^{n} (\tilde{y} - X_{ij}\beta_j)^2\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 + \varepsilon\sum_{i=1}^{n} \left[\tilde{y}_i^2 - 2\tilde{y}_i X_{ij}\beta_j + X_{ij}^2\beta_j^2\right]\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 + \varepsilon\left[\sum_{i=1}^{n} \tilde{y}_i^2 - 2\sum_{i=1}^{n} \tilde{y}_i X_{ij}\beta_j + \sum_{i=1}^{n} X_{ij}^2\beta_j^2\right]\right)\right\} \\
&\propto \frac{1}{\sqrt{\varepsilon}} \exp\left\{\beta_j^2 + \varepsilon\left[-2\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j + \sum_{i=1}^{n} X_{ij}\beta_j^2\right]\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 - 2\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j + \varepsilon\beta_j^2\sum_{i=1}^{n} X_{ij}^2\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 + \varepsilon\beta_j^2\sum_{i=1}^{n} X_{ij}^2 - 2\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2\left(1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2\right) - 2\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j\right)\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2\right)\left[\beta_j^2 - \frac{2\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j}{1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2}\right]\right\} \\
&= \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2\right)\left[\left(\beta_j^2 - \frac{\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j}{1 + \varepsilon\sum_{i=1}^{n} X_{ij}}\right)^2 - \left(\frac{\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j}{1 + \varepsilon\sum_{i=1}^{n} X_{ij}}\right)^2\right]\right\} \\
&\propto \frac{1}{\sqrt{\varepsilon}} \exp\left\{-\frac{1}{2\varepsilon}\left(\beta_j^2 - \frac{\varepsilon\sum_{i=1}^{n} \tilde{y} X_{ij}\beta_j}{1 + \varepsilon\sum_{i=1}^{n} X_{ij}}\right)^2\left(1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2\right)\right\}
\end{aligned}
$$

The full conditional of β_j when $\delta_j = 0$ is, therefore, given by

$$
p(\beta_j \mid \delta_j = 0, \varepsilon, y) \sim \text{Normal}\left(\frac{\varepsilon\sum_{i=1}^{n} X_{ij}\tilde{y}_i}{1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2}, \varepsilon\left(1 + \varepsilon\sum_{i=1}^{n} X_{ij}^2\right)^{-1}\right)
$$

We then repeat the process for the full conditional of β_j with $\delta_j = 1$.

$$
\begin{aligned}
p(\beta_j \mid \delta_j = 0, \tau^2, \tau^2, y) &\propto \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\tilde{y} - X_{ij}\beta_j)^2\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2}\left(\frac{\beta_j^2}{\tau^2} + \sum_{i=1}^{n}(\tilde{y} - X_{ij}\beta_j)^2\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 + \tau^2\sum_{i=1}^{n}(\tilde{y} - X_{ij}\beta_j)^2\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 + \tau^2\sum_{i=1}^{n}\left[\tilde{y}_i^2 - 2\tilde{y}_i X_{ij}\beta_j + X_{ij}^2\beta_j^2\right]\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 + \tau^2\left[\sum_{i=1}^{n}\tilde{y}_i^2 - 2\sum_{i=1}^{n}\tilde{y}_i X_{ij}\beta_j + \sum_{i=1}^{n}X_{ij}^2\beta_j^2\right]\right)\right\} \\
&\propto \frac{1}{\sqrt{\tau^2}} \exp\left\{\beta_j^2 + \tau^2\left[-2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j + \sum_{i=1}^{n}X_{ij}\beta_j^2\right]\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 - 2\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j + \tau^2\beta_j^2\sum_{i=1}^{n}X_{ij}^2\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 + \tau^2\beta_j^2\sum_{i=1}^{n}X_{ij}^2 - 2\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2\left(1 + \tau^2\sum_{i=1}^{n}X_{ij}^2\right) - 2\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j\right)\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(1 + \tau^2\sum_{i=1}^{n}X_{ij}^2\right)\left[\beta_j^2 - \frac{2\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j}{1 + \tau^2\sum_{i=1}^{n}X_{ij}^2}\right]\right\} \\
&= \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(1 + \tau^2\sum_{i=1}^{n}X_{ij}^2\right)\left[\left(\beta_j^2 - \frac{\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j}{1 + \tau^2\sum_{i=1}^{n}X_{ij}}\right)^2 - \left(\frac{\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j}{1 + \tau^2\sum_{i=1}^{n}X_{ij}}\right)^2\right]\right\} \\
&\propto \frac{1}{\sqrt{\tau^2}} \exp\left\{-\frac{1}{2\tau^2}\left(\beta_j^2 - \frac{\tau^2\sum_{i=1}^{n}\tilde{y}X_{ij}\beta_j}{1 + \tau^2\sum_{i=1}^{n}X_{ij}}\right)^2\left(1 + \tau^2\sum_{i=1}^{n}X_{ij}^2\right)\right\}
\end{aligned}
$$

The full conditional of β_j when $\delta_j = 1$ is, therefore, given by

$$
p(\beta_j \mid \delta_j = 1, \tau^2, y) \sim \text{Normal}\left(\frac{\tau^2\sum_{i=1}^{n}X_{ij}\tilde{y}_i}{1 + \tau^2\sum_{i=1}^{n}X_{ij}^2}, \tau^2\left(1 + \tau^2\sum_{i=1}^{n}X_{ij}^2\right)^{-1}\right)
$$

Question (c)

Show that the full conditional distribution of δ_j is $\text{Bernoulli}\left(\dfrac{p_1}{p_0 + p_1}\right)$ with $p_1 = \pi \exp\left\{-\frac{1}{2\tau^2}\beta_j^2\right\}$ and $p_0 = \dfrac{(1-\pi)\tau}{\sqrt{\varepsilon}} \exp\left\{-\dfrac{1}{2\varepsilon}\beta_j^2\right\}$.

solution:

$$
\begin{aligned}
p(\delta_j \mid \beta_j, \tau^2, \varepsilon) &= p(\beta_j|\delta_j, \tau^2, \varepsilon)p(\delta_j|\pi) \\
&\propto \left(\frac{1}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}\right)^{\delta_j}\left(\frac{1}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}\right)^{(1-\delta_j)} \quad \pi^{\delta_j}(1-\pi)^{(1-\delta_j)} \\
&\propto \left(\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}\right)^{\delta_j}\left(\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}\right)^{(1-\delta_j)} \\
&\propto \left(\frac{\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}}{\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}+\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}\right)^{\delta_j}\left(\frac{\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}{\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}+\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}\right)^{(1-\delta_j)} \\
&= \left(\frac{\pi\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}}{\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}+\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}\right)^{\delta_j}\left(\frac{\frac{(1-\pi)\tau}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}{\frac{\pi}{\tau}\exp\left\{-\frac{\beta_j^2}{2\tau^2}\right\}+\frac{1-\pi}{\sqrt{\varepsilon}}\exp\left\{-\frac{\beta_j^2}{2\varepsilon}\right\}}\right)^{(1-\delta_j)}
\end{aligned}
$$

We have now shown that the full conditional distribution of δ_j is Bernoulli$\left(\dfrac{p_1}{p_0+p_1}\right)$ with $p_1 = \pi\exp\left\{-\frac{1}{2\tau^2}\beta_j^2\right\}$ and $p_0 = \dfrac{(1-\pi)\tau}{\sqrt{\varepsilon}}\exp\left\{-\dfrac{1}{2\varepsilon}\beta_j^2\right\}$.

Question (d)

Write down the full conditional distribution of π.

solution:

$$
\begin{aligned}
p\left(\pi \mid \delta_j, \frac{a_\pi}{2}, \frac{b_\pi}{2}\right) &\propto p\left(\pi|\frac{a_\pi}{2}, \frac{b_\pi}{2}\right)\prod_{j=1}^{J} p(\delta_j|\pi) \\
&\propto \pi^{\frac{a_\pi}{2}-1}(1-\pi)^{\frac{b_\pi}{2}-1}\prod_{j=1}^{J}\pi^{\delta_j}(1-\pi)^{(1-\delta_j)} \\
&\propto \pi^{\frac{a_\pi}{2}-1}(1-\pi)^{\frac{b_\pi}{2}-1}\pi^{\sum_{j=1}^{J}\delta_j}(1-\pi)^{\sum_{j=1}^{J}(1-\delta_j)} \\
&\propto \pi^{\sum_{j=1}^{J}\delta_j+\frac{a_\pi}{2}-1}(1-\pi)^{\sum_{j=1}^{J}(1-\delta_j)+\frac{b_\pi}{2}-1}
\end{aligned}
$$

Which means that

$$
\pi \mid \delta_j, \frac{a_\pi}{2}, \frac{b_\pi}{2} \sim \text{Beta}\left(\sum\delta_j + \frac{a_\pi}{2}, \sum(1-\delta_j) + \frac{b_\pi}{2}\right)
$$

Question (e)

Write down a Gibbs sampler algorithm to sample from the joint posterior distribution of θ.

solution:

With Gibbs sampling, the idea is to create a Markov Chain with stationary distribution equal to the full posterior, so that we can generate posterior samples from it. We go back and forth updating the parameters one at a time using the current value of all the other parameters. We start with the simplest conditional and move toward the most complicated.

For our case, in particular, the algorithm is

1. Choose starting values for β, δ, and π and initialize to those starting values
2. For each iteration, k up to number of iterations chosen:
   (a) Sample from $p(\pi \mid -)$ using the most recently sampled δ_j
   (b) Sample from $p(\delta_j \mid -)$ using the most recently sampled π and β_j's (or initial values of β_j if we are in iteration 1.
   (c) Sample from $p(\beta_j \mid -)$, calculated using the updated value of δ_j and the most recent values of $\beta_j, j = 1 \ldots, n$. Unless $j = 1$ or $j = n$, this will involve using β_1 to β_{j-1} from the current iteration of the algorithm, and β_{j+1} to β_n from the previous iteration of the algorithm.
3. Check convergence diagnostics by viewing trace plots, auto-correlation and $\hat{R}$ measurements, as well as effective sample size calculations.
4. Remove burn-in, and re-check convergence diagnostics.
5. If necessary, apply thinning and re-check convergence diagnostics.

Part II

Question (f)

Let $\varepsilon = 10^{-4}$ and $\tau^2 = 10^2$. Explain heuristically how the spike-and-slab prior allows for variable selection in the multiple linear regression model context.

solution:

Question (g)

Let $a_\pi = b_\pi = 1$, the bathtub prior distribution for π. Using ε and τ as in part (f), obtain posterior estimates for the coefficient β.

solution:

In order to make our code as readable as possible, we write helper functions to update π, δ_j, and β_j as well as an index generator function to assist with sampling from the proper past values of β_j.

```
indx_gen = function(j, M) {
  n = dim(M)[2]
  if (j == 1) return(M[1, ])
  if (j == n) return(M[2, ])
  return(c(M[2, 1 : j-1], M[1, j:n]))
}

update_pi = function(delta_j, a_pi, b_pi) {
  shape1 = sum(delta_j) + a_pi/2
  shape2 = sum(1-delta_j) + b_pi/2
  rbeta(n = 1, shape1, shape2)
}

update_delta = function(bet, pi, tau, eps, j) {
  p0 = ((1 - pi) * tau / sqrt(eps)) * exp(-1/(2*eps) * bet[j]^2)
  p1 = pi * exp(-1/(2*tau^2) * bet[j]^2)
  p = p1 / (p0 + p1)
  return(rbinom(n = 1, size = 1, prob = p))
```

```r
}

update_beta = function(delta_j, j, bet, tau, eps, mutop = 0, mubot = 0) {
  partial.mutop = mutop
  partial.mubot = mubot
  for(i in 1:dim(X)[1]) {
    partial.mutop = partial.mutop + X[i, j] *
      (y[i] - sum(setdiff(X[i,], X[i,j]) * setdiff(bet, bet[j])))
    partial.mubot = partial.mubot + X[i, j]^2
  }

  if(delta_j == 0) {
    bot = 1 + eps * partial.mubot
    mu = eps * partial.mutop / bot
    sd = sqrt(eps * bot^-1)
    return(rnorm(n = 1, mean = mu, sd = sd))

  }
  if(delta_j == 1) {
    bot = 1 + tau^2 * partial.mubot
    mu = tau^2 * partial.mutop / bot
    sd = sqrt(tau^2 * bot^-1)
    return(rnorm(n = 1, mean = mu, sd = sd))
  }
}
```

Our Gibbs sampler function follows the algorithm written for solution (e), except that we only maintain the most recent values for π and δ_j

```r
gibbs = function(n_iter, init, priors) {
  beta.out = matrix(data = NA, nrow = n_iter, ncol = dim(X)[2])
  delta.curr = init$delta_j
  beta.curr = init$beta
  beta.out[1, ] = beta.curr

  for(k in 2:n_iter) {
    pi.curr = update_pi(delta_j = delta.curr, a_pi = priors$a_pi, b_pi = priors$b_pi)

    for(j in 1:length(beta.curr)) {
      delta.curr = update_delta(bet = beta.out[k-1,], pi = pi.curr,
                                tau = priors$tau, eps = priors$eps, j = j)

      betas = indx_gen(j = j, M = beta.out[(k-1) : k, ])
      beta.curr[j] = update_beta(delta_j = delta.curr, j = j, bet = betas,
                                 tau = priors$tau, eps = priors$eps)

      beta.out[k, j] = beta.curr[j]
    }
  }
  return(beta.out)
}
```

Here we set up our priors and initialize with chosen starting values.

```r
priors = list()
init = list()
n_iter = 10000

model = lm(y ~ X - 1)
init$beta = model$coefficients
init$delta_j = 0

priors$a_pi = 1
priors$b_pi = 1
priors$tau = 10
priors$eps = 10^-4
```

Finally, we run the algorithm and show summary statistics for each β_j

```r
# post = gibbs(n_iter = n_iter, init = init, priors = priors)
post = readRDS(file = "post_1.rds")
colnames(post) = c("beta1", "beta2", "beta3", "beta4", "beta5",
                   "beta6", "beta7", "beta8", "beta9", "beta10")
tail(post)
```

```
##              beta1    beta2         beta3    beta4       beta5    beta6
##  [9995,] 0.6809122 2.519954 -0.0004358975 1.828270 0.011221094 1.301888
##  [9996,] 0.6730794 2.516862  0.0031754523 1.835281 0.011007978 1.304312
##  [9997,] 0.6741057 2.516831  0.0072263607 1.840275 0.011299413 1.304856
##  [9998,] 0.6868114 2.514227  0.0097722773 1.830462 0.009155944 1.310620
##  [9999,] 0.6754362 2.526089  0.0023899840 1.825766 0.013899248 1.311648
## [10000,] 0.6680775 2.524683  0.0029283353 1.830294 0.010601889 1.313442
##                 beta7    beta8    beta9    beta10
##  [9995,] -0.011697079 2.019323 1.491756 0.7655213
##  [9996,] -0.021612322 2.021356 1.492682 0.7596288
##  [9997,] -0.015886620 2.006744 1.504224 0.7483262
##  [9998,] -0.021002517 2.011659 1.495771 0.7628297
##  [9999,] -0.008654419 1.996997 1.497623 0.7695771
## [10000,] -0.012761882 2.004457 1.493148 0.7719868
```

Question (h)

Check for convergence of the MCMC chains using trace plots and compute $\hat{R}$.

solution:

First, we calculate $\hat{R}$ for our chain of sampled β values

```r
calc.rhat = function(m, nchain, J, chain) {
  rhat = numeric(J)
  for (j in 1:J) {
    psi.mean = mean(chain[, j])
    psi.bar = numeric(m)
    aux.w = numeric(m)
    for (k in 1:m) {
      sub.chain = chain[seq((k - 1) * nchain + 1, k * nchain, 1), j]
      psi.bar[k] = mean(sub.chain)
      aux.w[k] = (1 / (nchain - 1)) * sum((sub.chain - mean(sub.chain))^2)
    }
```

```r
    B = (nchain / (m - 1)) * (sum((psi.bar - psi.mean)^2))
    W = (1 / m) * sum(aux.w)
    VP = ((nchain - 1) / nchain) * W + (1 / nchain) * B
    rhat[j] = sqrt(VP / W)
  }
  return(rhat)
}

rhat.post = calc.rhat(m = 5, nchain = 100, J = 10, chain = post)
# TODO: add names for columns
rhat.post
```

```
##  [1] 2.857479 1.724200 1.050331 1.762468 2.050055 1.961335 1.041590 1.761001
##  [9] 1.264461 1.391706
```

The $\hat{R}$ values are not close enough to 1 for us to believe convergence has occurred, and our original chain shows a great deal of auto-correlation so we remove a burn-in of $2,000$ and thin the chain by a step-size of 10.

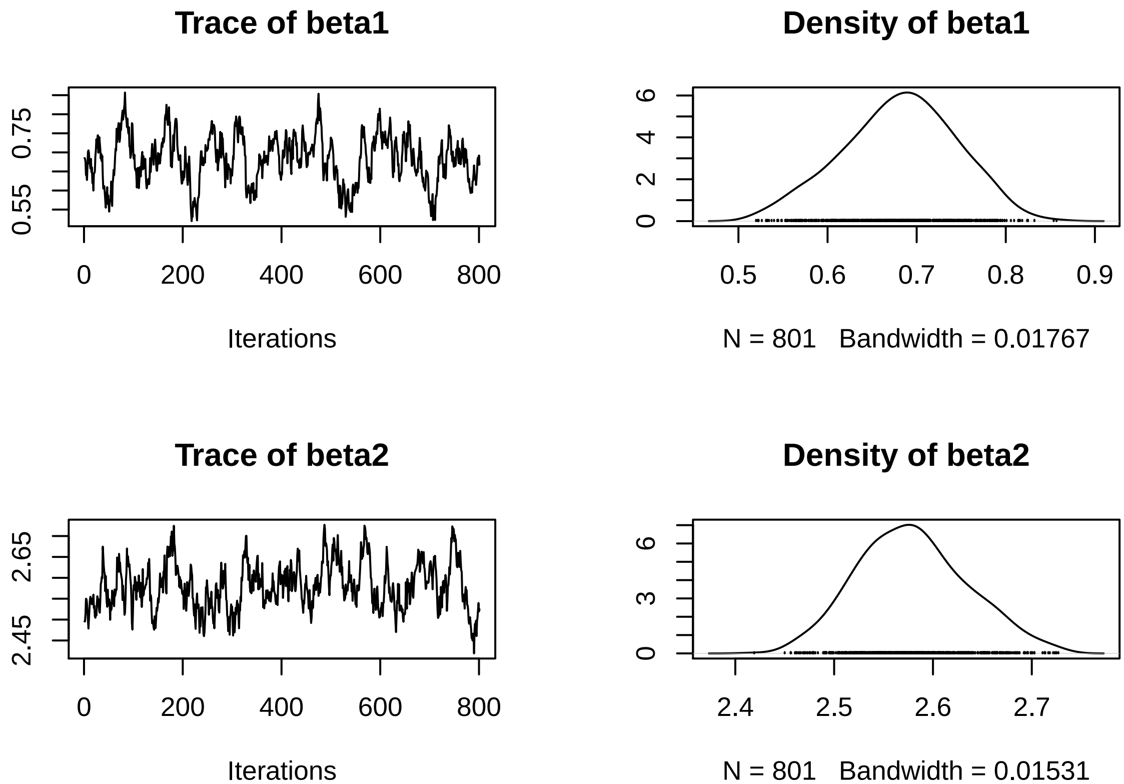
```r
burn.in = 2000
thin_interval = 10
thin_indx = seq(from = burn.in, to = length(post[, 1]), by = thin_interval)

thin.post = post[thin_indx, ]
colnames(thin.post) = c("beta1", "beta2", "beta3", "beta4", "beta5",
                        "beta6", "beta7", "beta8", "beta9", "beta10")
# burned.post = post[burn.in:n_iter, ]
# thin.post = Thin(x = post, By = 10)
rhat.thin.post = calc.rhat(m = 5, nchain = 100, J = 10, chain = thin.post)
# TODO: add names for columns
rhat.thin.post
```

```
##  [1] 1.0442415 1.0931170 0.9989601 1.0400011 1.0284213 1.0431538 1.0633191
##  [8] 1.0734171 1.0470726 1.0023907
```

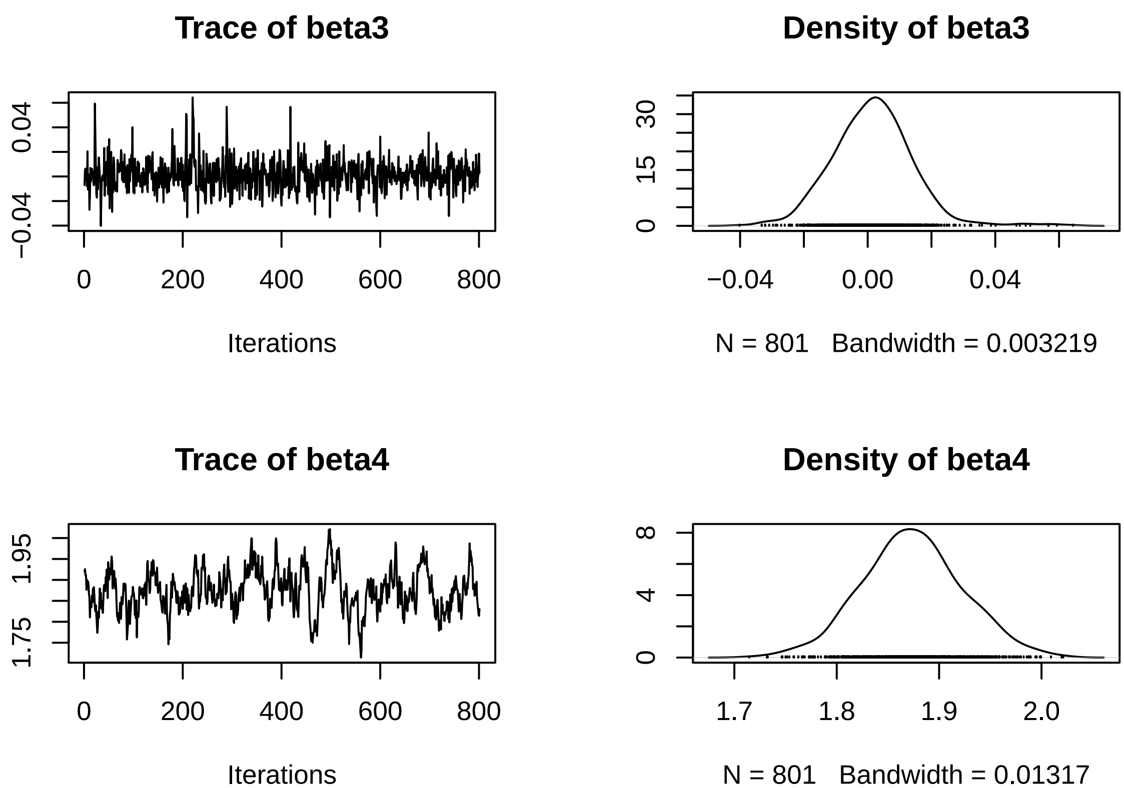
The $\hat{R}$ values for our thinned chain look much better. We now produce trace, density and auto-correlation plots to further assess convergence.

Trace plots and densities of β_j's:

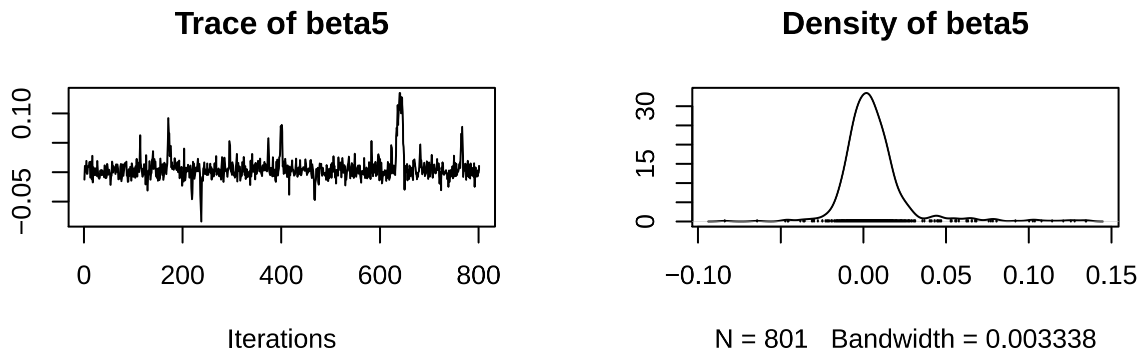
```r
plot(as.mcmc(thin.post[, 1:2]))
```



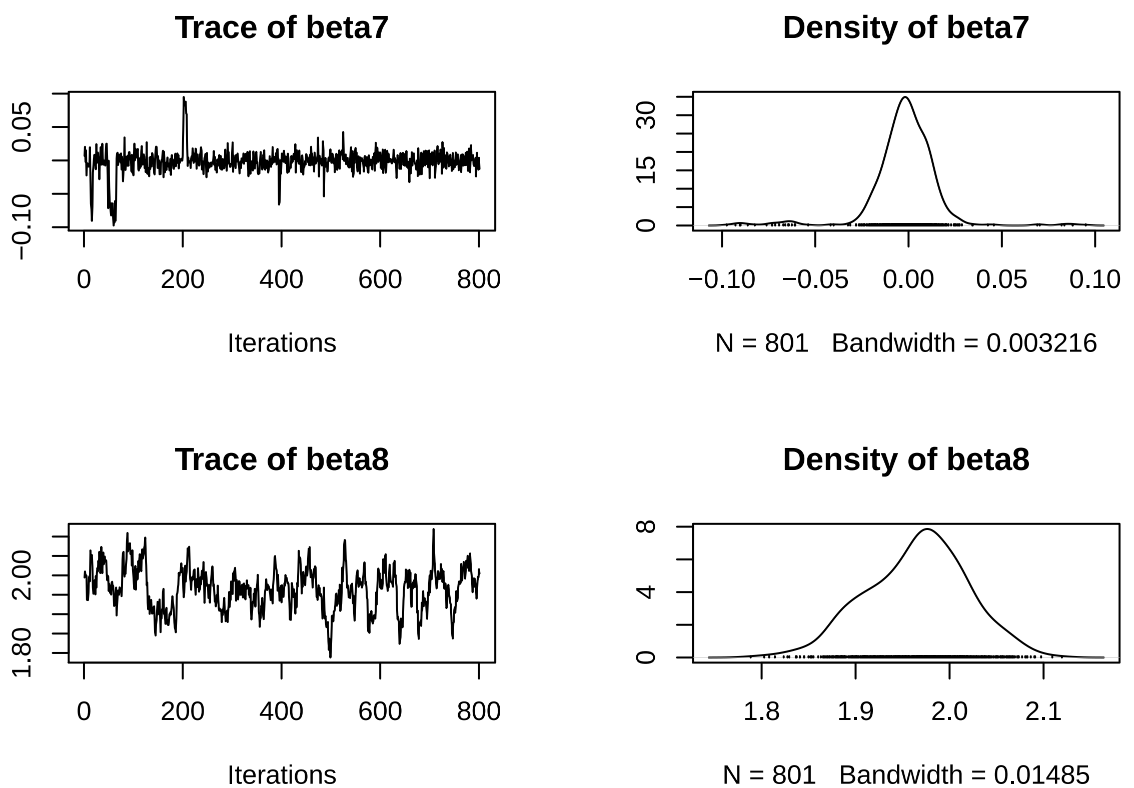

```
plot(as.mcmc(thin.post[, 3:4]))
```



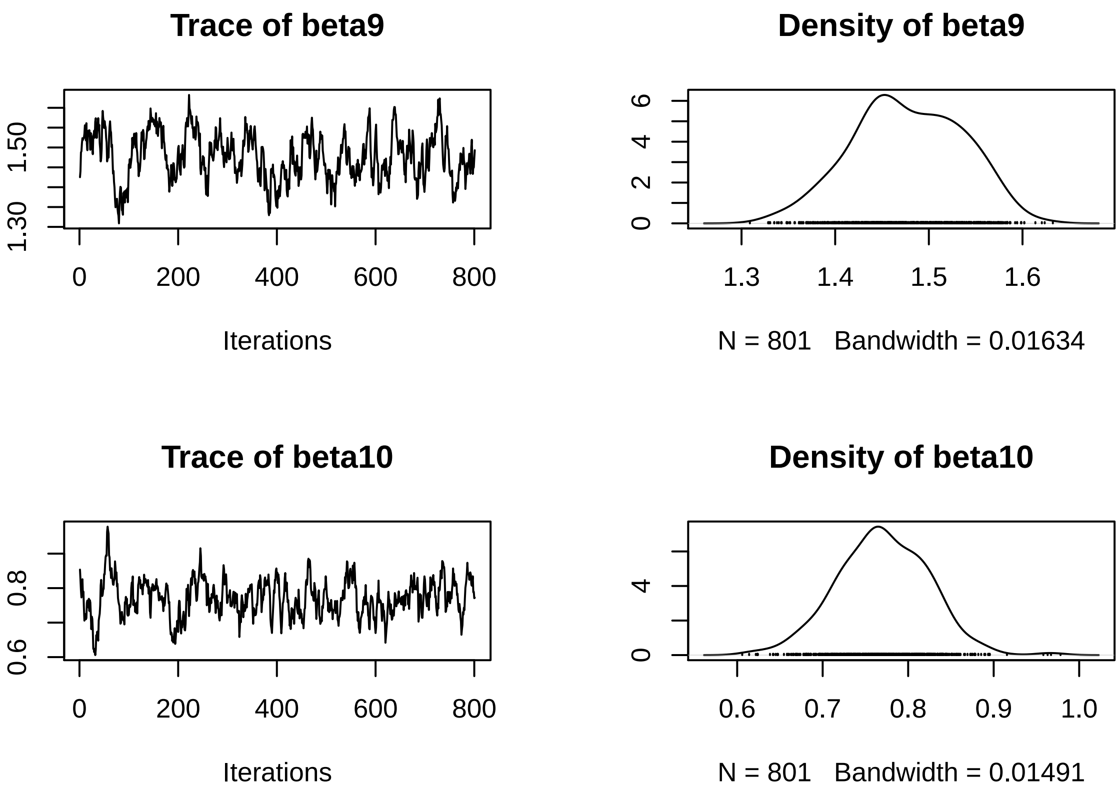
```r
plot(as.mcmc(thin.post[, 5:6]))
```




```r
plot(as.mcmc(thin.post[, 7:8]))
```



```r
plot(as.mcmc(thin.post[, 9:10]))
```



The trace plots show us that our sampler seams to be traversing the entire space and that the thinned posterior Markov chain maintains a consistent mean. This, coupled with the $\hat{R}$ values all being close to 1, makes us believe convergence has been achieved.

Question (i)

If the MCMC is converging, present the results including the posterior mean, posterior variance, and a 95% credible interval for each coefficient. Based on these results, which covariates are important to predict the response variable?

solution:

Summary statistics for the mean and standard deviation are given in the following output.

```r
summary(as.mcmc(thin.post))
```

```
##
## Iterations = 1:801
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 801
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD  Naive SE Time-series SE
## beta1    0.680955 0.06348 0.0022430       0.011259
## beta2    2.579083 0.05538 0.0019569       0.008668
```

```
## beta3   0.001531 0.01271 0.0004491       0.000398
## beta4   1.873839 0.04925 0.0017403       0.006746
## beta5   0.006118 0.02047 0.0007232       0.002256
## beta6   1.252682 0.05713 0.0020186       0.009523
## beta7  -0.002086 0.01819 0.0006428       0.001767
## beta8   1.965439 0.05334 0.0018846       0.007713
## beta9   1.477401 0.05871 0.0020746       0.009380
## beta10  0.770231 0.05358 0.0018933       0.007645
##
## 2. Quantiles for each variable:
##
##             2.5%       25%       50%      75%   97.5%
## beta1   0.55302  0.639137  0.684683 0.726048 0.79385
## beta2   2.47597  2.540816  2.576731 2.614505 2.69529
## beta3  -0.02118 -0.006560  0.001718 0.008935 0.02558
## beta4   1.77518  1.841573  1.873406 1.904996 1.97279
## beta5  -0.01947 -0.004172  0.003326 0.011900 0.06333
## beta6   1.15030  1.211115  1.249664 1.293718 1.36491
## beta7  -0.06075 -0.008611 -0.001207 0.006872 0.02473
## beta8   1.86023  1.929034  1.969758 2.001213 2.06445
## beta9   1.36343  1.438234  1.474662 1.523223 1.58213
## beta10  0.66578  0.733534  0.768778 0.808937 0.87513
```

```
# TODO: Add comments about header of summary not being accurate
```

The 95% Credible Intervals for each of the β_j's are given in the following output.

```
hdi(as.mcmc(thin.post))
```

```
##           beta1    beta2       beta3    beta4       beta5    beta6       beta7
## lower 0.5524856 2.488881 -0.02215572 1.772839 -0.02752419 1.153080 -0.02662521
## upper 0.7914541 2.702750  0.02407283 1.968164  0.04706820 1.367617  0.02742339
##          beta8   beta9    beta10
## lower 1.866417 1.36895 0.6581661
## upper 2.067160 1.58400 0.8612452
## attr(,"credMass")
## [1] 0.95
```

```
# TODO: see if Michelle needs variance rather than sd
```

Since the 95% Credible Intervals for β_3, β_5, and β_7 contain zero, we consider them insignificant to predict the response variable. All the other covariates are significant.

Question (j)

Sensitivity Analysis. Consider four different prior distributions for π by choosing values of a_π and b_π that change the shape of the beta distribution. Plot the prior of π for each of these values. Is the posterior distribution of β sensitive to these new prior distributions?

Question (k)

Model Checking. Generate $10,000$ replications of the data y^{rep} using the same x_i as the original data. Compare the posterior mean and median. Based on that, does the model generate predicted results similar to the observed data in the study?