HW 4 Sample solution

Solution courtesy of Rayleigh

November 14, 2017

Problem 1

```
library(lars)
data("diabetes")

d_x <- diabetes$x
d_y <- diabetes$y
solve(t(d_x) %*% d_x, t(d_x) %*% d_y)
```

```
##           [,1]
## age  -10.01220
## sex -239.81909
## bmi  519.83979
## map  324.39043
## tc  -792.18416
## ldl  476.74584
## hdl  101.04457
## tch  177.06418
## ltg  751.27932
## glu   67.62539
```

Problem 2

```
library(glmnet)

lasso_results <- glmnet(d_x, d_y, intercept=FALSE)
coef(lasso_results, s = 0.237)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)    .
## age            -0.2139686
## sex          -227.2727887
## bmi           526.6020553
## map           315.0293065
## tc           -221.0884867
## ldl            21.5337979
## hdl          -142.6427595
## tch           108.4340885
## ltg           539.5482418
## glu            64.6927909
```

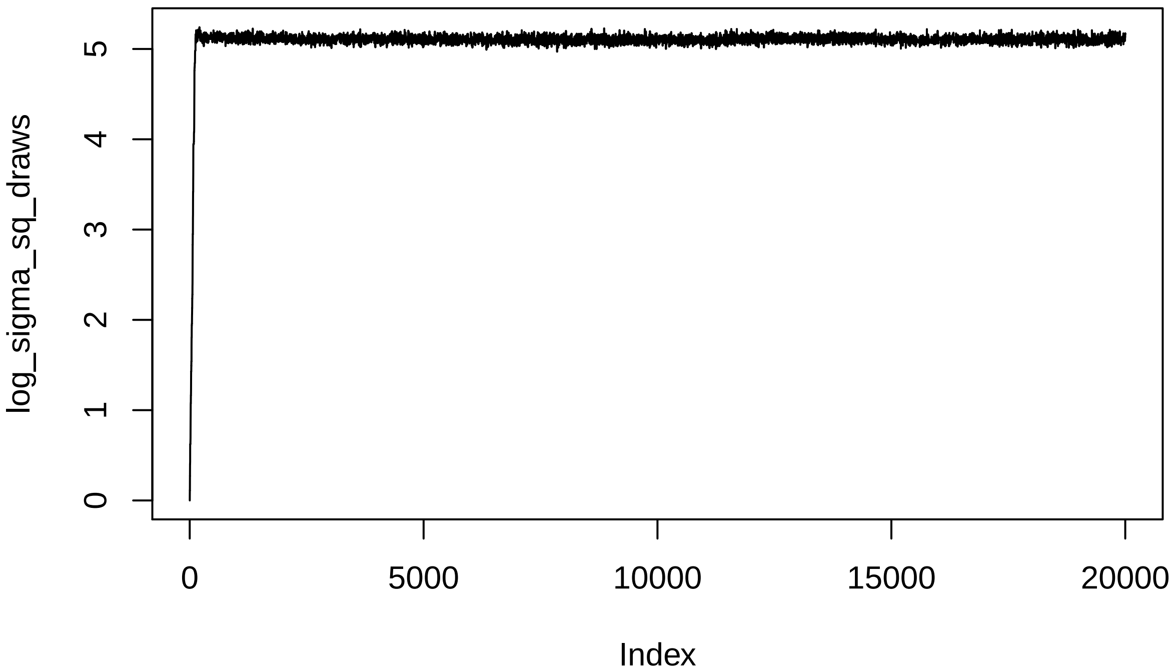
Problem 3

Note that if u = $\log(\sigma^2)$, $\sigma^2 = e^u$ and $\frac{\partial \sigma^2}{\partial u} = e^u$. As $\frac{1}{\sigma^2} = e^{-u}$, by change of variables, u $\propto$ 1.

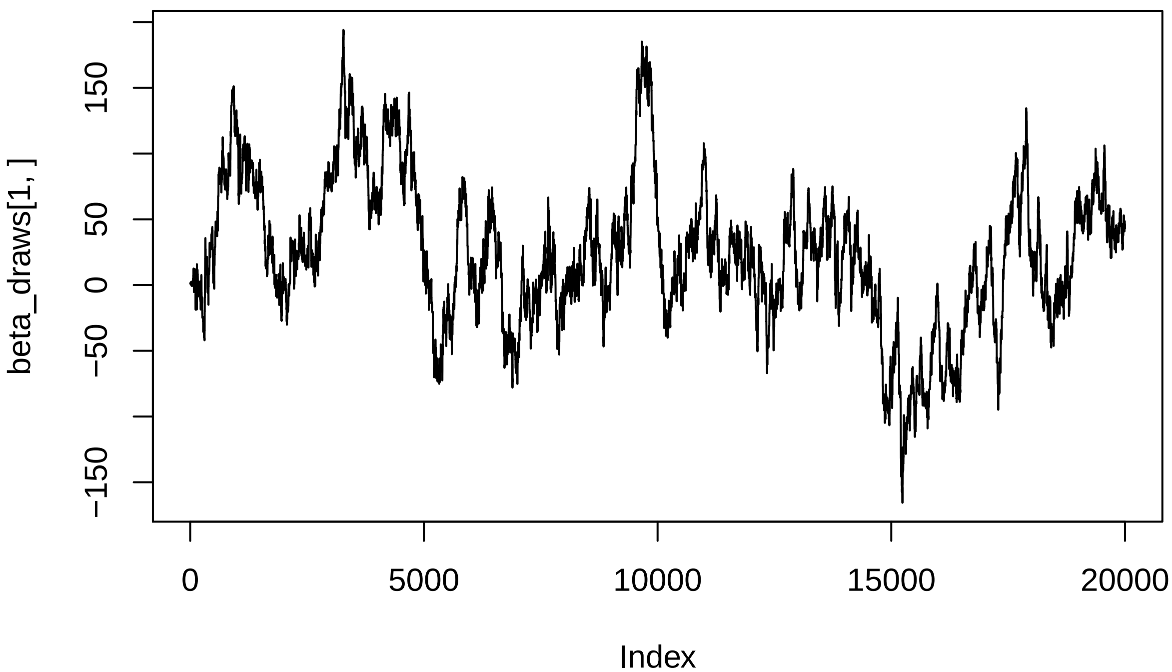
```
library(mvtnorm)
calc_pi <- function(x, y, beta, log_sigma_sq, lambda, p) {
  sum(dnorm(y, x %*% beta, rep(exp(log_sigma_sq), length(y)), log = T)) +
    p * log(lambda / 2) - p/2 * log_sigma_sq - lambda / sqrt(exp(log_sigma_sq)) * sum(abs(beta))
}

p <- ncol(d_x)
lambda = 0.237
beta_draws <- matrix(1, nrow = p)
log_sigma_sq_draws <- c(0)
num_draws = 20000
accept = 0
set.seed(12345)
for (i in 1:num_draws) {
  log_sigma_sq = rnorm(1, log_sigma_sq_draws[i], 0.1)
  beta <- t(rmvnorm(1, beta_draws[, i], exp(log_sigma_sq) * lambda * diag(p)))
  accept_ratio = calc_pi(d_x, d_y, beta, log_sigma_sq, lambda, p) -
                 calc_pi(d_x, d_y, beta_draws[, i], log_sigma_sq_draws[i], lambda, p)
  if (runif(1) <= exp(accept_ratio)) {
    accept = accept + 1
    beta_draws <- cbind(beta_draws, beta)
    log_sigma_sq_draws <- c(log_sigma_sq_draws, log_sigma_sq)
  } else {
    beta_draws <- cbind(beta_draws, beta_draws[, i])
    log_sigma_sq_draws <- c(log_sigma_sq_draws, log_sigma_sq_draws[i])
  }
}

#Acceptance ratio
accept / num_draws
```

```
## [1] 0.36135
```

```
#Trace plots
plot(log_sigma_sq_draws, type = "l")
```




```r
plot(beta_draws[1,], type = "l")
```



The acceptance ratio is acceptable. However, as seen in the trace plots, the betas don't look like they converged yet even though $\log(\sigma^2)$ looks to have converged. Unfortunately, this doesn't improve even when increasing the number of iterations to 50,000. However, for the sake of comparison, I'll throw out the first 10,000 draws. Then,

```r
rowMeans(beta_draws[, 10002:20001])
```

```
##  [1]    5.170015   -7.246206  277.394352   30.214661   18.259949
##  [6]   -5.055242 -180.124486   19.609346  370.308211   46.520984
```

Problem 4

Note that we can marginalize out μ because

$$\begin{aligned}
p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma^2) &\propto \int_\mu p(\mathbf{y}, \mu \mid \mathbf{X}, \beta, \sigma^2) d\mu \\
&= \int_\mu p(\mathbf{y} \mid \mu, \mathbf{X}, \beta, \sigma^2) p(\mu) d\mu \\
&\propto \int_\mu \prod_{i=1}^n \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp(-\frac{1}{2\sigma^2}(y_i - (\mu + X_i\beta))^2) d\mu \\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \int_\mu \exp(-\frac{1}{2\sigma^2}\sum_i^n (y_i - (\mu + X_i\beta))^2) d\mu \\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \int_\mu \exp(-\frac{1}{2\sigma^2}\sum_i^n (\mu - (y_i - X_i\beta))^2) d\mu \\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \int_\mu \exp(-\frac{1}{2\sigma^2}\sum_i^n (\mu - (\overline{y} - \overline{X}\beta))^2 - \frac{1}{2\sigma^2}\sum_i^n ((y_i - X_i\beta) - (\overline{y} - \overline{X}\beta))^2) d\mu \\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp(-\frac{1}{2\sigma^2}\sum_i^n ((y_i - X_i\beta) - (\overline{y} - \overline{X}\beta))^2) \int_\mu \exp(-\frac{1}{2\sigma^2}\sum_i^n (\mu - (\overline{y} - \overline{X}\beta))^2) d\mu \\
&\propto \frac{1}{\sqrt{(2\pi\sigma^2)^{n-1}}} \exp(-\frac{1}{2\sigma^2}\sum_i^n ((y_i - X_i\beta) - (\overline{y} - \overline{X}\beta))^2)
\end{aligned}$$

Conditional distribution of σ^2

$$\begin{aligned}
p(\sigma^2 \mid \mathbf{y}, \mathbf{X}, \beta, \tau^2) &\propto p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma^2) p(\beta \mid \tau^2, \sigma^2) p(\sigma^2) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{\frac{n-1}{2}} \exp\left(-\frac{1}{\sigma^2}\left(-\frac{1}{2}\left(\mathbf{y} - X\beta - (\overline{y} - \overline{X}\beta)\,\mathbf{1}_n\right)^2\right)\right. \\
&\left(\frac{1}{\sigma^2}\right)^{\frac{p}{2}} \exp\left(-\frac{1}{\sigma^2}\left(\sum_{j=1}^p \frac{\beta_j^2}{2\tau_j^2}\right)\right)\left(\frac{1}{\sigma^2}\right) \\
&= \left(\sigma^2\right)^{-\frac{n+p-1}{2}-1} \exp\left(\frac{1}{\sigma^2}\left(-\frac{1}{2}\left(\left(\mathbf{y} - X\beta - (\overline{y} - \overline{X}\beta)\mathbf{1}_n\right)^2 + \sum_{j=1}^p \frac{\beta_j^2}{\tau_j^2}\right)\right)\right)
\end{aligned}$$

so $\sigma^2 \sim \Gamma^{-1}\left(\frac{n+p-1}{2}, -\frac{1}{2}\left(\left(\mathbf{y} - X\beta - (\overline{y} - \overline{X}\beta)\mathbf{1}_n\right)^2 + \sum_{j=1}^p \frac{\beta_j^2}{\tau_j^2}\right)\right)$.

Conditional distribution of beta

If D^τ represents the diagonal matrix such that $D^\tau_{j,j} = \tau_j^2$,

$$
\begin{aligned}
p(\beta \mid \mathbf{y}, \mathbf{X}, \sigma^2, \tau^2) &\propto p(\mathbf{y} \mid \mathbf{X}, \beta, \sigma^2) p(\beta \mid \tau^2, \sigma^2) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{y} - X\beta - (\overline{y} - \overline{X}\beta)\mathbf{1}_n\right)^2\right) \exp\left(-\frac{1}{2\sigma^2}\left(\beta^T (D^\tau)^{-1}\beta\right)\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}\left((\mathbf{y} - \overline{y}\mathbf{1}_n) - (X - \mathbf{1}_n\overline{X})\beta\right)^2\right) \exp\left(-\frac{1}{2\sigma^2}\left(\beta^T (D^\tau)^{-1}\beta\right)\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\left(-2(\mathbf{y} - \overline{y}\mathbf{1}_n)^T (X - \overline{X}\mathbf{1}_n)\beta + \beta^T (X - \mathbf{1}_n\overline{X})^T (X - \mathbf{1}_n\overline{X})\beta + \beta^T (D^\tau)^{-1}\beta\right)\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}\left(-2(\mathbf{y} - \overline{y}\mathbf{1}_n)^T (X - \overline{X}\mathbf{1}_n)\beta + \beta^T\left((X - \mathbf{1}_n\overline{X})^T (X - \mathbf{1}_n\overline{X}) + (D^\tau)^{-1}\right)\beta\right)\right) \\
&\propto \exp\left(-\frac{1}{2}(\beta - \mu_\beta)(\Sigma_\beta^{-1})(\beta - \mu_\beta)\right)
\end{aligned}
$$

We get the last line by completing the square and letting

$$
M = (X - \mathbf{1}_n\overline{X})^T (X - \mathbf{1}_n\overline{X}) + (D^\tau)^{-1}
$$

$$
\Sigma_\beta^{-1} = \frac{1}{\sigma^2} M
$$

$$
\mu_\beta = M^{-1}(X - \overline{X}\mathbf{1}_n)^T (\mathbf{y} - \overline{y}\mathbf{1}_n)
$$

Hence, $\beta \sim N(\mu_\beta, \Sigma_\beta)$.

Conditional distribution of τ^2

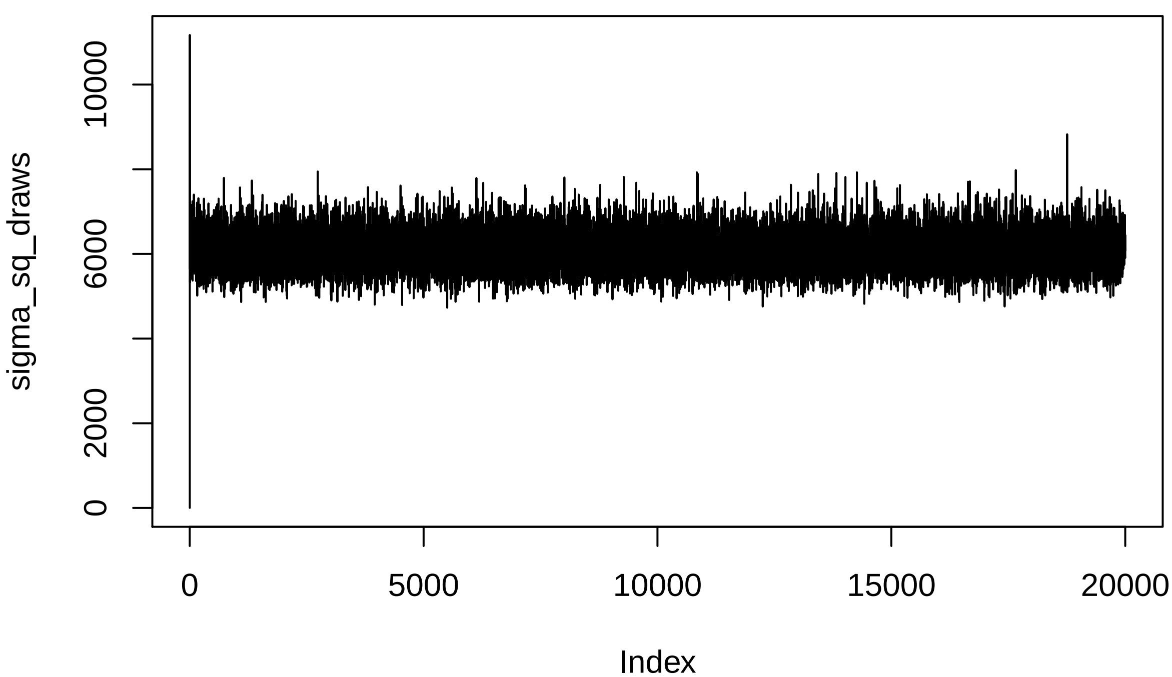
Note that τ_j is conditionally independent given β and σ^2. Hence,

$$
\begin{aligned}
p(\tau_j^2 \mid \mathbf{y}, \mathbf{X}, \sigma^2, \beta) &\propto p(\beta_j \mid \tau_j^2, \sigma^2) p(\tau_j^2) \\
&\propto \frac{1}{\sqrt{\tau_j^2}} \exp\left(-\frac{\beta_j^2}{2\tau_j^2\sigma^2} - \frac{\lambda^2\tau_j^2}{2}\right) \\
&= \frac{1}{\sqrt{\tau_j^2}} \exp\left(\frac{1}{2}\left(-\frac{\beta_j^2}{\sigma^2}\frac{1}{\tau_j^2} - \frac{\lambda^2}{\frac{1}{\tau_j^2}}\right)\right) \\
&= \frac{1}{\sqrt{\tau_j^2}} \exp\left(-\frac{1}{2}\left(\frac{\frac{\beta_j^2}{\sigma^2}\left(\frac{1}{\tau_j^2}\right)^2 + \lambda^2}{\frac{1}{\tau_j^2}}\right)\right) \\
&\propto \frac{1}{\sqrt{\tau_j^2}} \exp\left(-\frac{1}{2}\left(\frac{\frac{\beta_j^2}{\sigma^2}\left(\frac{1}{\tau_j^2} - \frac{\sigma\lambda}{\beta_j}\right)^2}{\frac{1}{\tau_j^2}}\right)\right) \\
&\propto \frac{1}{\sqrt{\tau_j^2}} \exp\left(-\frac{1}{2}\left(\frac{\lambda^2\left(\frac{1}{\tau_j^2} - \frac{\sigma\lambda}{\beta_j}\right)^2}{\frac{\sigma^2\lambda^2}{\beta_j^2}\frac{1}{\tau_j^2}}\right)\right)
\end{aligned}
$$

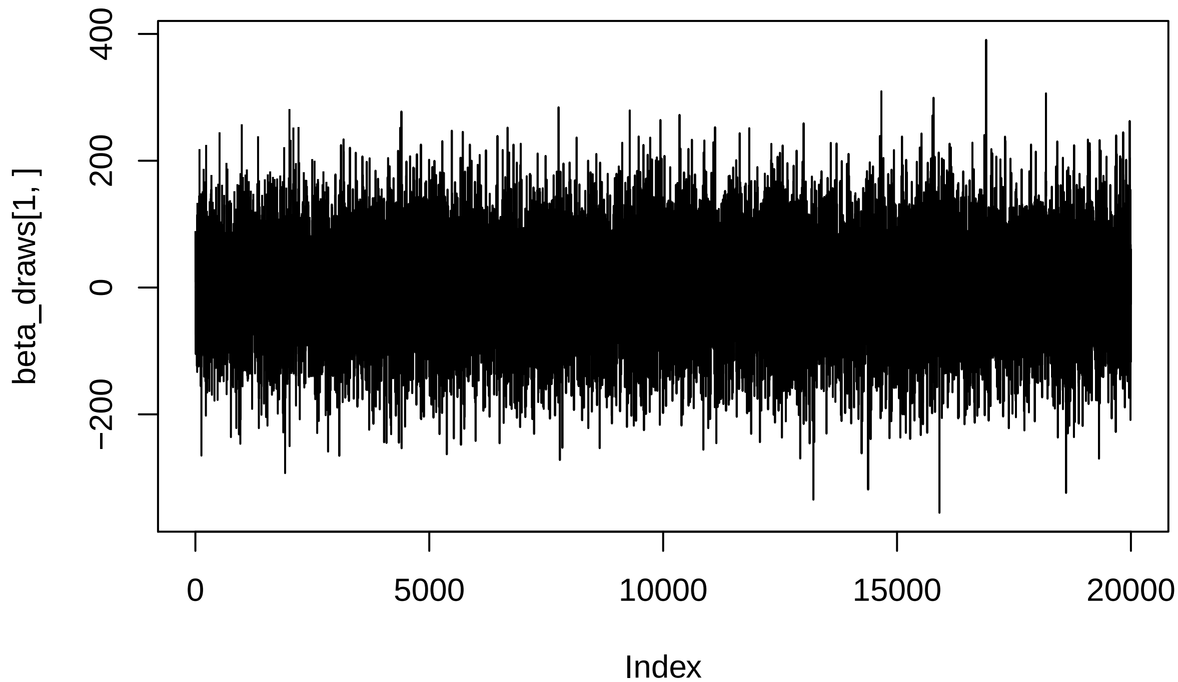
This is an inverse-Gaussian kernel for $\frac{1}{\tau_j^2}$ with $\mu = \frac{\lambda\sigma}{\beta_j}$ and $\lambda' = \lambda^2$.

```
if (!require(statmod)) {
  install.packages("statmod")
  library(statmod)
```

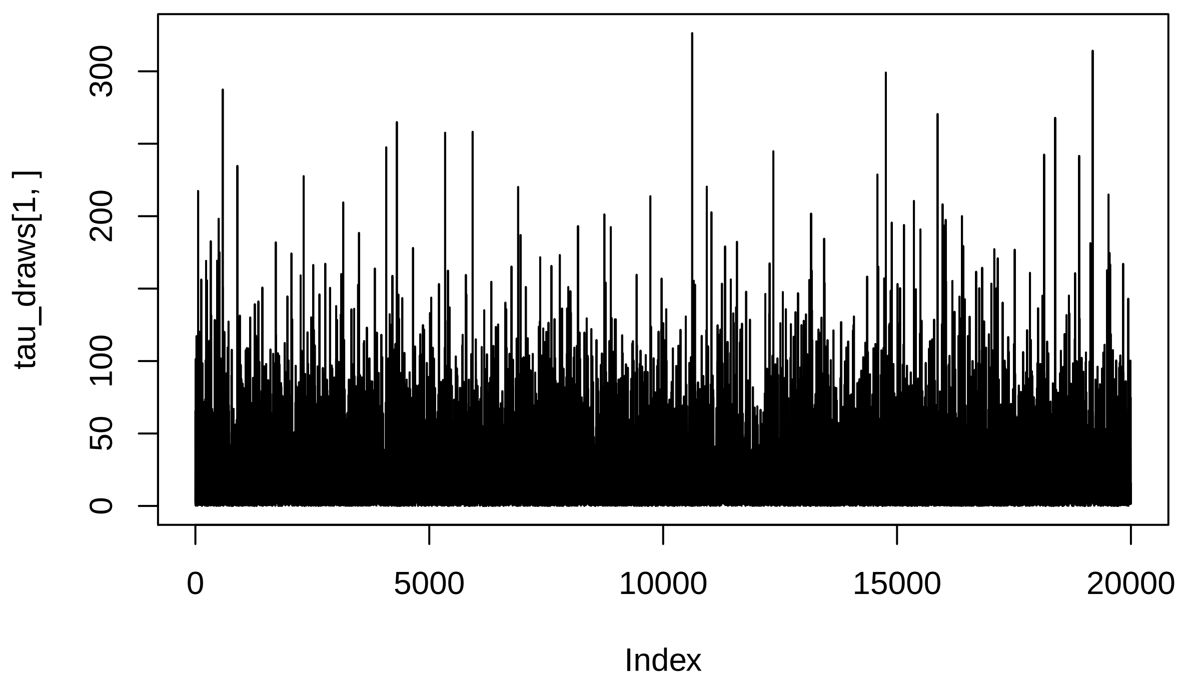
```
}

cd_x <- apply(d_x, 2, function(col) {col - mean(col)})
cd_y <- d_y - mean(d_y)

n = length(cd_y)
p <- ncol(d_x)
lambda = 0.237
sigma_sq_draws <- c(1)
beta_draws <- matrix(1, nrow = p)
tau_draws <- beta_draws
num_draws = 20000
accept = 0

set.seed(12345)
for (i in 1:num_draws) {
  sigma_sq_draws <- c(sigma_sq_draws,
                      1 / rgamma(1, (n + p - 1) / 2,
                                 sum((cd_y - cd_x %*% beta_draws[,i])^2) +
                                 sum(beta_draws[, i]^2 / tau_draws[, i]))))
  m <- (t(cd_x) %*% cd_x + diag(1 / tau_draws[, i]))
  beta_draws <- cbind(beta_draws,
                      t(rmvnorm(1, solve(m, t(cd_x) %*% cd_y), solve(m) *
                      sigma_sq_draws[i + 1])))
  tau_draws <- cbind(tau_draws,
                     unlist(sapply(beta_draws[, (i + 1)], function(beta) {
                       1 / rinvgauss(1, sqrt(lambda^2 * sigma_sq_draws[i + 1] / beta^2), lambda^2)
                     })))
}

#Trace plots
plot(sigma_sq_draws, type = "l")
```



```r
plot(beta_draws[1,], type = "l")
```




```r
plot(tau_draws[1,], type = "l")
```



Looking at trace plots, it looks to have converged. (Not all were included in this file.) Then, I'll throw out the first 10,000 draws.

```r
rowMeans(beta_draws[, 10002:20001])
```

```
##  [1]   -2.179532 -206.036991  520.877028  302.247567 -161.711967
##  [6]  -12.125208 -161.022880   98.387032  511.033030   65.383270
```

Problem 5

It is interesting that the regression coefficients for regression and LASSO are similar for sex, bmi, map, and glu. It is also interesting to note that the Bayesian LASSO returns values similar to LASSO except for the LDL covariate. Metropolis-Hastings doesn't give values that are close to either LASSO algorithms because the MCMC chains fail to converge, partly due to the poor proposal design.