Biostat 276 Project 1

Kelly Li

1/29/2022

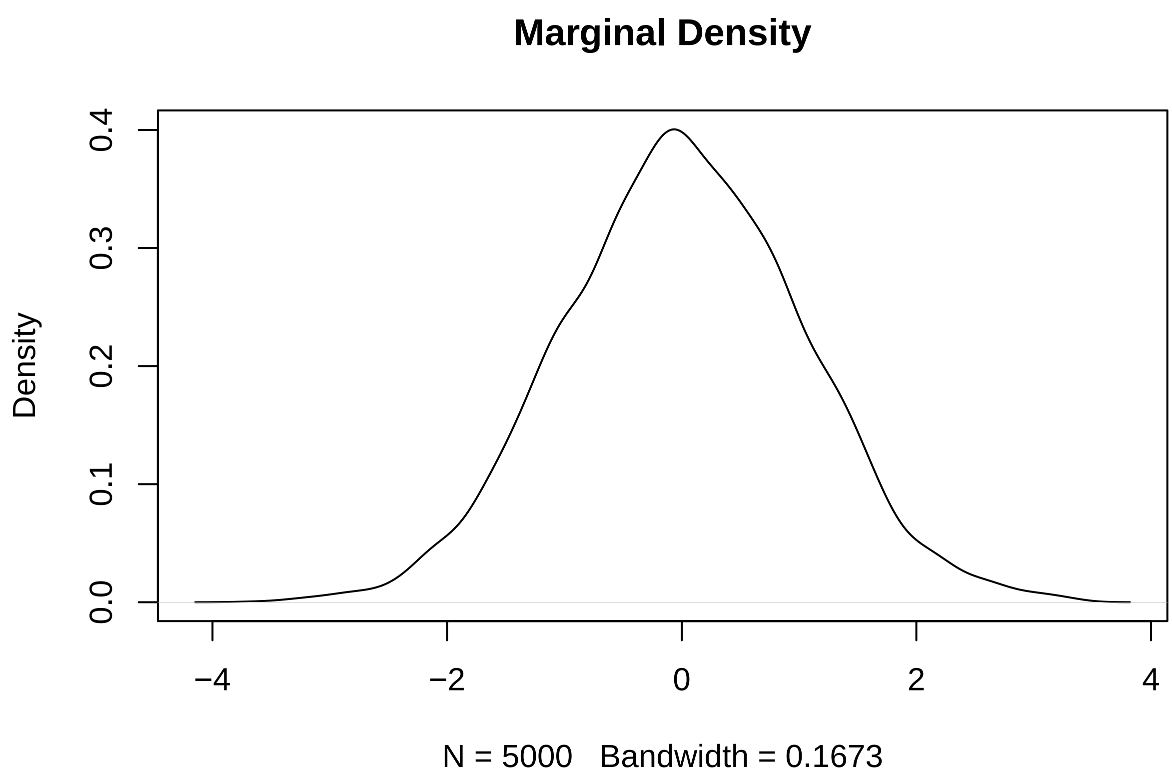
Sampling from the Banana Distribution

a)

Bayesian Adaptive Lasso

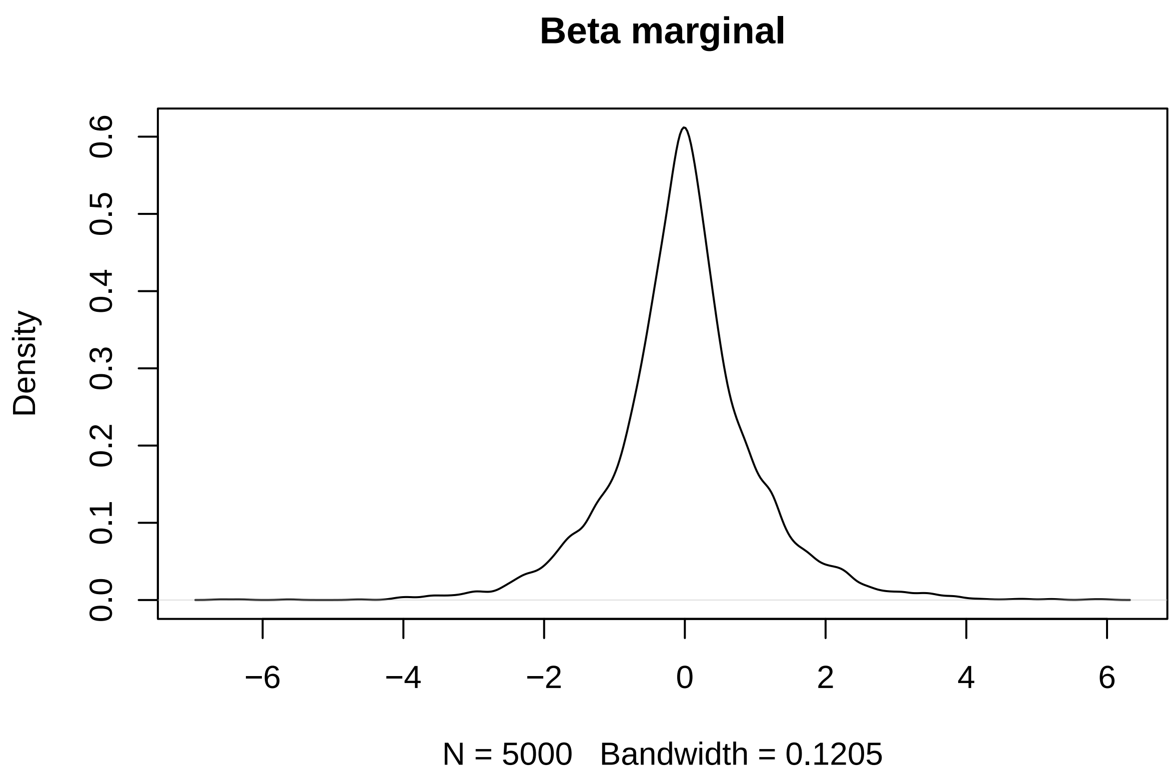
#a)

```
sima <- rnorm(5000, 0, 1)
plot(density(sima), main = "Marginal Density")
```



b)

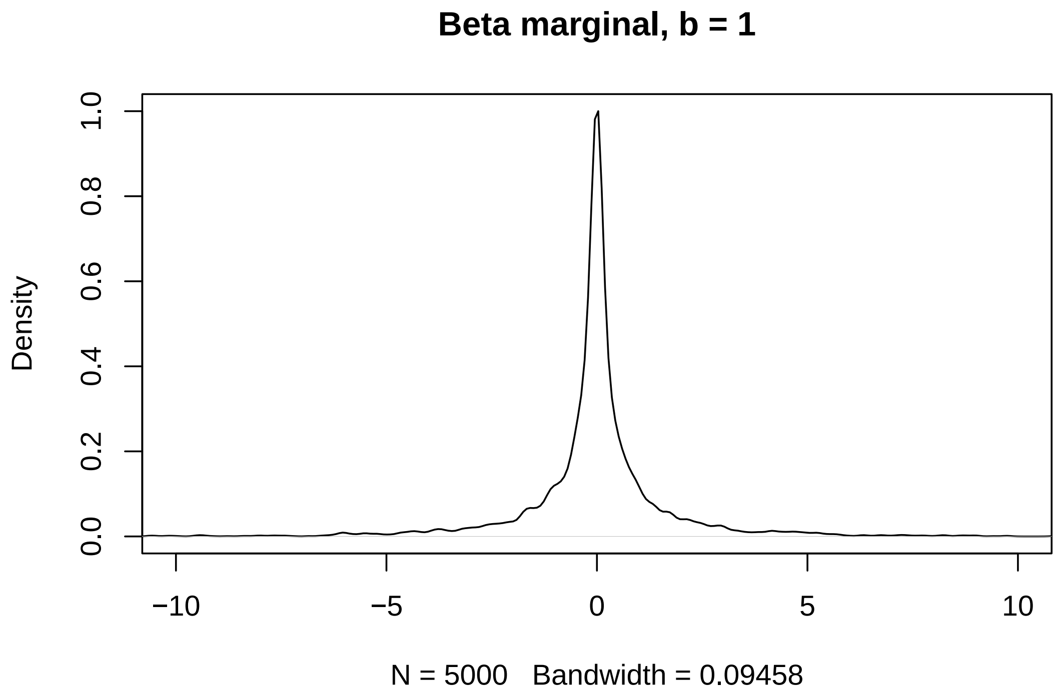
```r
lambda2 <- 2
tau2 <- rgamma(5000, shape = 1, rate = lambda2/2)
simb <- rnorm(5000, 0, sqrt(tau2))
plot(density(simb), main = "Beta marginal")
```

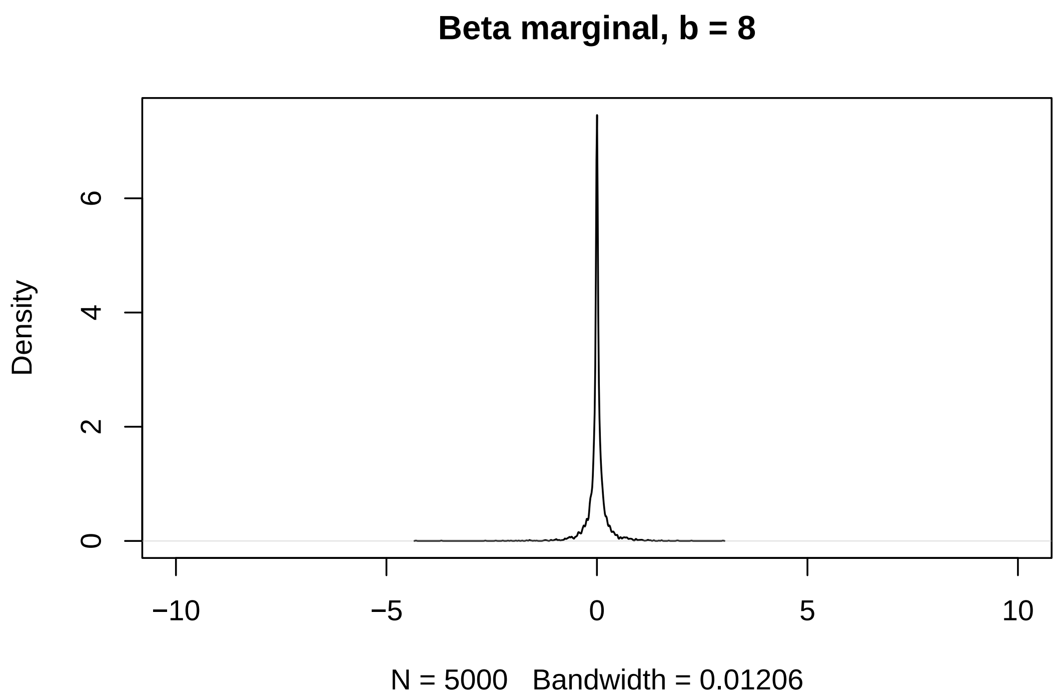


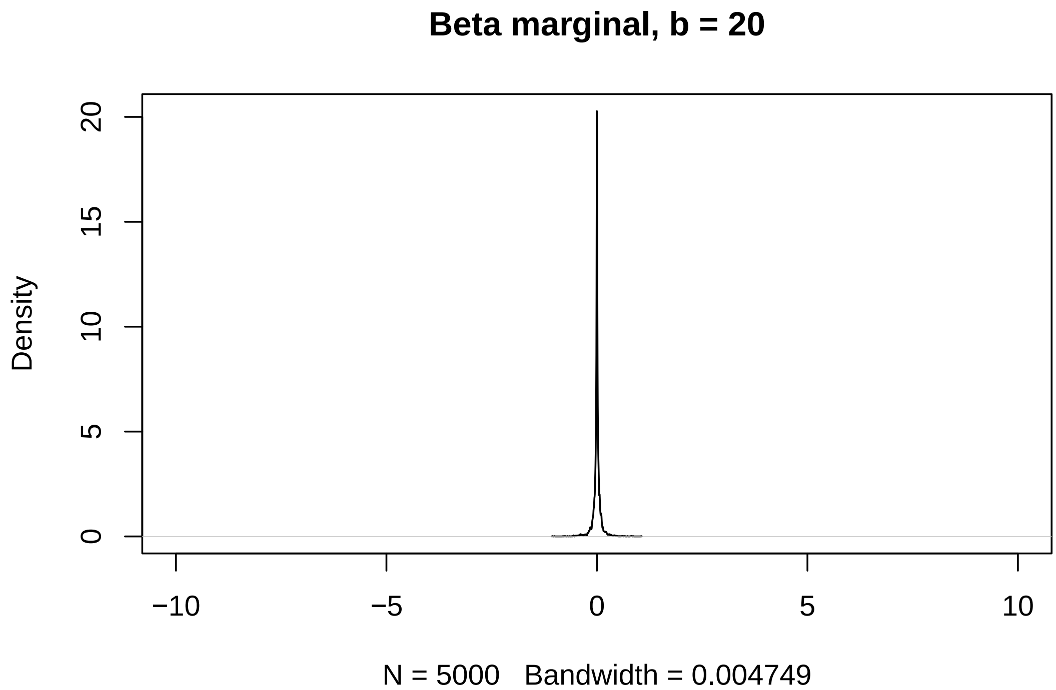
c)

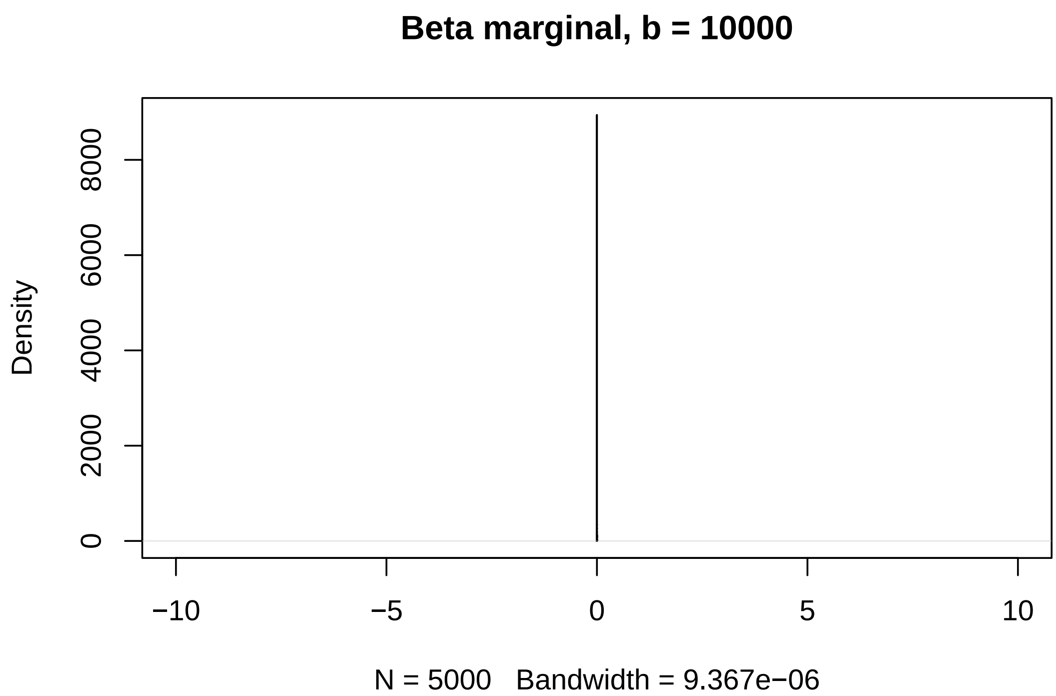
```r
bvec <- c(1, 8, 20, 10000)
marginalplot <- function(n, b){
  lambda <- 1/rgamma(n, 1, b)
  tau2 <- rgamma(n, shape = 1, rate = lambda^2/2)
  sim <- rnorm(n, 0, sqrt(tau2))
  plot <- plot(density(sim),
               main = paste0("Beta marginal, b = ", b),
               xlim = c(-10, 10))
  save_plot <- recordPlot()
  return(save_plot)
}

plots <- lapply(bvec, marginalplot, n = 5000)
```


Beta marginal, b = 1
Density
1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10
N = 5000 Bandwidth = 0.09458


Beta marginal, b = 8
Density
6
4
2
0
−10
−5
0
5
10
N = 5000 Bandwidth = 0.01206


Beta marginal, b = 20
Density
0
5
10
15
20
−10
−5
0
5
10
N = 5000   Bandwidth = 0.004749


Beta marginal, b = 10000
Density
0
2000
4000
6000
8000
−10
−5
0
5
10
N = 5000   Bandwidth = 9.367e−06

d)

We compute the full conditionals below for use in MCMC implementation:

$$p(\vec{\beta}, \tau_1^2, \dots, \tau_p^2, \sigma^2, \lambda^2|\vec{y}) \propto p(\vec{y}|\vec{\beta}, \tau_1^2, \dots, \tau_p^2, \sigma^2, \lambda^2)p(\vec{\beta}|\tau_1^2, \dots, \tau_p^2,)p(\tau_1^2, \dots, \tau_p^2, |\lambda^2)p(\lambda^2)p(\sigma^2)$$

$$\begin{aligned}
p(\sigma^2|\vec{y}) &\propto p(\vec{y}|\vec{\beta}, \tau_1^2, \dots, \tau_p^2, \beta, \lambda^2) \\
&\propto p(\vec{y}|\vec{\beta}, \tau_1^2, \dots, \tau_p^2, \sigma^2, \lambda^2)p(\sigma^2) \\
&\propto (\sigma^2)^{\frac{-n}{2}} \exp[\frac{-1}{2\sigma^2}(\vec{y} - X\vec{\beta})^T(\vec{y} - X\vec{\beta})](\sigma^2)^{-0.1-1} \exp(\frac{-0.1}{\sigma^2}) \\
&= \sigma^{2(-\frac{n}{2}+0.1)-1} \exp[-\frac{1}{\sigma^2}(0.1 + \frac{1}{2}(\vec{y} - X\vec{\beta})^T(\vec{y} - X\vec{\beta})] \\
&= \text{Inverse Gamma}(\frac{n}{2} + 0.1, 0.1 + \frac{1}{2}(\vec{y} - X\vec{\beta})^T(\vec{y} - X\vec{\beta}))
\end{aligned}$$

$$\begin{aligned}
p(\vec{\beta}|\vec{y}) &\propto p(\vec{y}|\vec{\beta}, \tau_1^2, \dots, \tau_p^2, \sigma^2, \lambda^2)(\vec{\beta}|\tau_1^2, \dots, \tau_p^2,) \\
&\propto \exp[-\frac{1}{2\sigma^2}(\vec{y} - X\vec{\beta})^T(\vec{y} - X\vec{\beta})] \exp[-\frac{1}{2}\Sigma^{-1}\vec{\beta}^T\vec{\beta}] \text{ for } \Sigma = \text{diag}(\tau_i^2) \\
&= \exp[-\frac{1}{2\sigma^2}(\vec{y}^T\vec{y} - 2\vec{y}^y(X\vec{(\beta)})^T + (X\vec{\beta})^T(X\vec{\beta})) - \frac{1}{2}\vec{\beta}^T\Sigma^{-1}\vec{\beta}) \\
&\propto \exp[-\frac{1}{2}(\vec{\beta}^T(\frac{X^TX}{\sigma^2} + \Sigma^{-1})\vec{\beta}) - \frac{1}{2\sigma^2}2(X\vec{\beta})^T\vec{y}] \\
&= N_p([\frac{X^TX}{\sigma^2} + \Sigma^{-1}]^{-1}\frac{X^T\vec{y}}{\sigma^2}, [\frac{X^TX}{\sigma^2} + \Sigma^{-1}]^{-1}) \\
p(\tau_i^2|\beta_i, \lambda^2, \sigma^2, \vec{y}) &\propto p(\beta_i|\tau_i^2)p(\tau_i^2|\lambda^2) \\
&\propto \frac{1}{\tau_i} \exp[-\frac{1}{2}(\frac{\beta_i}{\tau_i})^2 - \frac{\lambda^2}{2}\tau_i^2] \\
&= (\frac{1}{\tau_i^2})^{\frac{1}{2}} \exp[-\frac{1}{2}(\beta^2(\frac{1}{\tau_i^2}) + \lambda^2(\frac{1}{\frac{1}{\tau_i^2}})]
\end{aligned}$$

Let $u = \frac{1}{\tau_i^2}$ such that the Jacobian: $\frac{d}{du}(\frac{1}{u}) = -\frac{1}{u^2}$

$$\begin{aligned}
p(e|\beta_i, \lambda) &\propto u^{\frac{1}{2}} \exp[-\frac{\lambda^2}{2}(\frac{\beta_i^2}{\lambda^2}u + \frac{1}{u})]u^{-2} \\
&\propto e^{-\frac{3}{2}} \exp[-\frac{\lambda^2}{2}(\frac{\beta_i^2}{\lambda^2}u + \frac{1}{u})] \\
&= IG([\frac{\lambda^2}{\beta_i^2}]^{\frac{1}{2}}, \lambda^2)
\end{aligned}$$

$$\begin{aligned}
p(\lambda_i^2|\tau_i^2, \vec{\beta}, \sigma^2, \vec{y}) &\propto p(\vec{\tau}|\lambda^2)p(\lambda^2) \\
&\propto \Pi_{i=1}^p \exp[-\frac{\lambda^2}{2}\tau_i^2](\lambda^2)^{a-1} \exp[-b\lambda^2] \\
&\propto \exp[-(\frac{\sum_{i=1}^p \tau_i^2}{2} + b)\lambda^2](\lambda^2)^{a-1} \\
&= \text{Gamma}(a, \frac{\sum_{i=1}^p \tau_i^2}{2} + b)
\end{aligned}$$

e)

I will implement a Gibbs Sampler algorithm to sample from the posterior distributions.

```
gibbs <- function(y = y, X = Xdat,
                  lambda2, tau2,
                  n.sim = 1000, burn = 0.1,
                  a = 1, b = 1, beta, fixlambda = FALSE){
  #Chain information
  n.total <- n.sim*(1.0 + burn)
  n.burn <- n.sim*burn

  #Initializing matrices
  betamu.out <- matrix(NA, n.sim, 10)
  beta.out <- matrix(NA, n.sim, 10)
  sigma2.out <- c()

  #Data and Parameters
  n <- length(y)
  p <- ncol(X)
  XtX <- t(X) %*% X

  for(i in 1:n.total){
    rss <- t(y - X %*% beta)%*%(y - X %*% beta)
    #lambda2
    if(fixlambda == TRUE){
        lambda2 <- lambda2
    } else {
        lambda2 <- rgamma(1, shape = a, rate = sum(tau2)/2 + b)
    }
    #tau2
    for(a in 1:length(beta)){
      tau2[a] <- rinvgauss(1, sqrt(lambda2)/sqrt(beta[a]^2), lambda2)
    }
    tau2 <- 1/tau2
    #sigma2
    shape <- 0.1
    scale <- rss/2 + 0.1
    sigma2 <- 1/rgamma(1, shape = shape, scale = scale)
    #beta
    betavar <- solve(XtX/sigma2 + solve(diag(tau2)))
    betamu <- betavar %*% t(X) %*% y/sigma2
    beta <- rmvnorm(n=1, mean = betamu, sigma = betavar) %>% t()
    if(i > n.burn){
      i1 <- i - n.burn
      betamu.out[i1, ] <- betamu
      beta.out[i1, ] <- beta
      sigma2.out[i1] <- sigma2
    }
  }
 return(list(beta = beta.out, sigma2 = sigma2.out))
}

data("diabetes")
```

My MCMC	Glmnet
-10.01211	-9.057331
-239.81884	-238.879817
519.83971	520.885704
324.39017	323.449275
-792.18083	-680.104304
476.74398	390.311899
101.04372	48.691747
177.06395	159.307982
751.27807	710.278264
67.62525	67.555464

```
Xdat <- diabetes$x
y <- diabetes$y
betahat <- solve(t(Xdat) %*% Xdat) %*% t(Xdat) %*% y
out <- gibbs(y = y, X = Xdat, n = 1000, burn = 0.1,
             a = 1, b = 1, beta = betahat, lambda2 = 1, tau2 = rep(1, 10))

coeffun <- apply(out$beta, 2, function(x){quantile(x,c(0.16,0.5,0.84))})
coefme <- coeffun[2,]

fit.glm <- glmnet(Xdat, y, intercept = FALSE)
coefglm <- coef(fit.glm, s = min(fit.glm$lambda))
compare <- tibble("My MCMC" = coefme, "Glmnet" = matrix(coefglm[-1]))

kable(compare) %>% kable_styling(full_width = F)
```

f

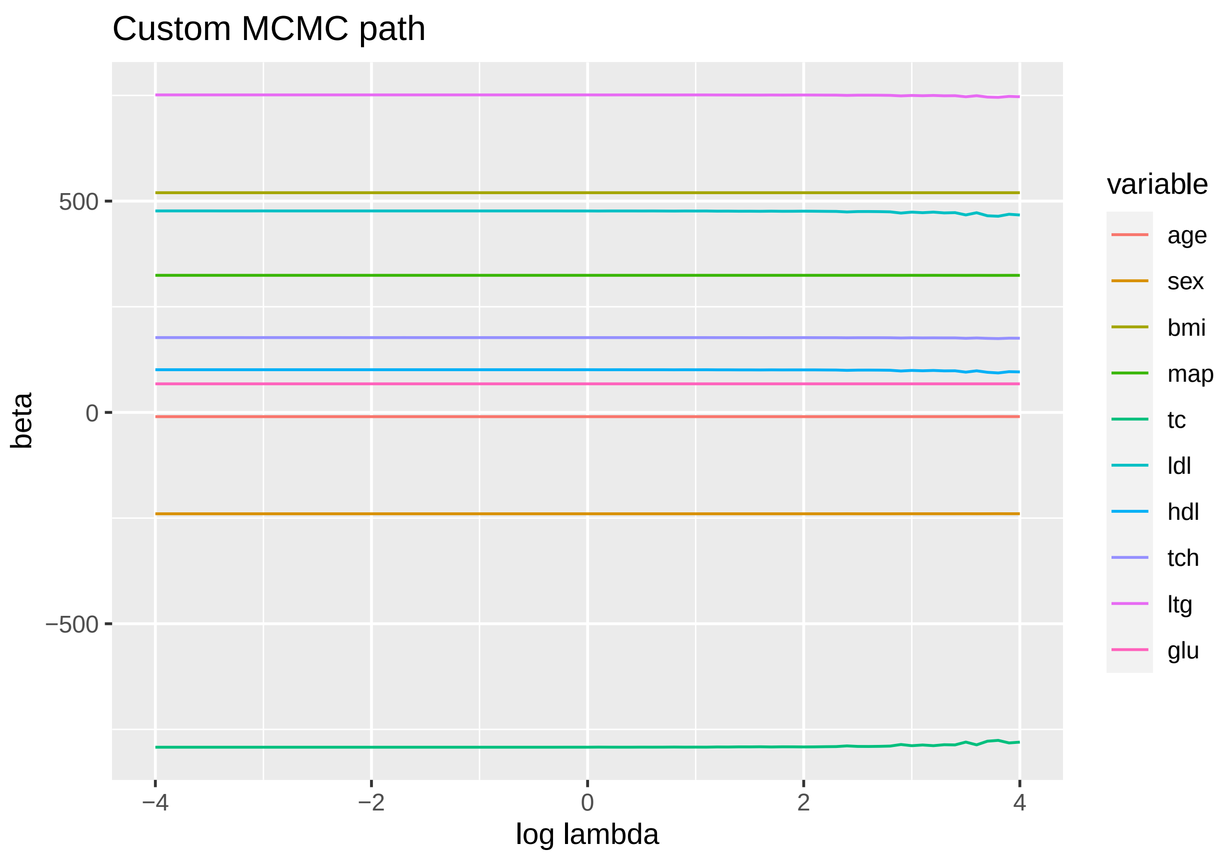
```
lambdas <- exp(seq(-4, 4, 0.1))^2
out <- lapply(lambdas, function(x)gibbs(y = y, X = Xdat,
                     n.sim = 1000, burn = 0.1,
                     a = 1, b = 1, beta = rep(0, 10),
                     lambda2 = x, tau2 = rep(1, 10),
                     fixlambda = TRUE))
betas <- lapply(out, `[`, c("beta"))
betas <- lapply(betas, sapply, colMedians)

df <- do.call("cbind", betas)
rownames(df) <- colnames(Xdat)
df <- t(df) %>% as.data.frame() %>% melt()
```

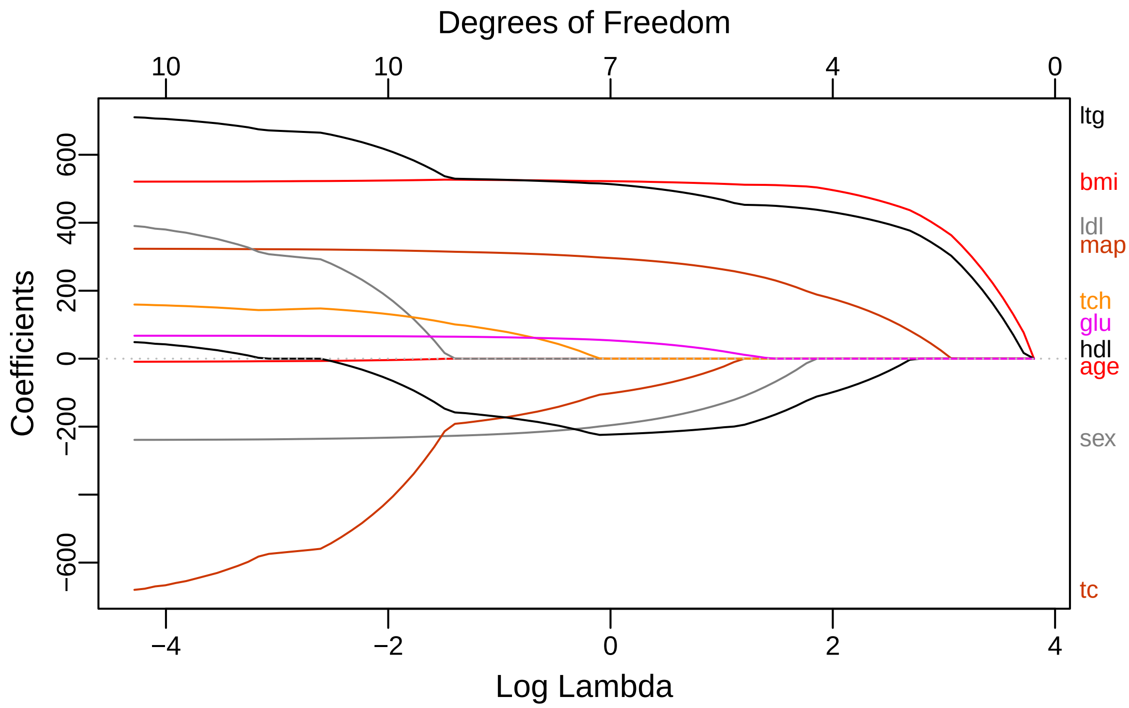
```
## No id variables; using all as measure variables
```

```
df$lambda <- rep(seq(-4, 4, 0.1), 10)

ggplot(data = df, aes(lambda, value, color = variable)) +
  geom_line() +
  ggtitle("Custom MCMC path") + xlab("log lambda") + ylab("beta")
```




```
plot_glmnet(fit.glm, xvar = "lambda")
```



g

```
df.ab <- data.frame(a = c(1, 5, 10, 1, 5, 10, 1, 5, 10),
                    b = c(1, 1, 1, 5, 5, 5, 10, 10, 10))
out <- apply(df.ab, 1, function(df.ab)gibbs(y = y, X = Xdat,
                     n.sim = 1000, burn = 0.1,
                     a = df.ab[1], b = df.ab[2], beta = betahat,
                     lambda2 = 1, tau2 = rep(1, 10)))
betas <- lapply(out, `[`, c("beta"))
betas <- lapply(betas, sapply, colMedians)
df <- do.call("cbind", betas)
colnames(df) <- paste0("(",df.ab$a,", ", df.ab$b, ")")
rownames(df) <- colnames(Xdat)
df %>% kable() %>% kable_styling(full_width = F)
```

	(1, 1)	(5, 1)	(10, 1)	(1, 5)	(5, 5)	(10, 5)	(1, 10)	(5, 10)	(10, 1
age	-10.01231	-10.01199	-10.01208	-10.01212	-10.01214	-10.01205	-10.01215	-10.01207	-10.012
sex	-239.81906	-239.81885	-239.81885	-239.81873	-239.81889	-239.81865	-239.81874	-239.81893	-239.818
bmi	519.83967	519.83963	519.83965	519.83951	519.83971	519.83963	519.83958	519.83970	519.839
map	324.39007	324.39010	324.39020	324.39026	324.39002	324.39004	324.39033	324.39013	324.389
tc	-792.18011	-792.18132	-792.17744	-792.18173	-792.18007	-792.18070	-792.18109	-792.18080	-792.181
ldl	476.74317	476.74350	476.74138	476.74262	476.74343	476.74290	476.74398	476.74297	476.744
hdl	101.04345	101.04361	101.04253	101.04353	101.04358	101.04274	101.04361	101.04295	101.043
tch	177.06392	177.06357	177.06342	177.06392	177.06370	177.06275	177.06379	177.06345	177.063
ltg	751.27798	751.27819	751.27775	751.27818	751.27780	751.27791	751.27820	751.27796	751.278
glu	67.62544	67.62536	67.62512	67.62523	67.62534	67.62535	67.62537	67.62536	67.625