

Biostat 276 Project 1

Kelly Li

1/29/2022

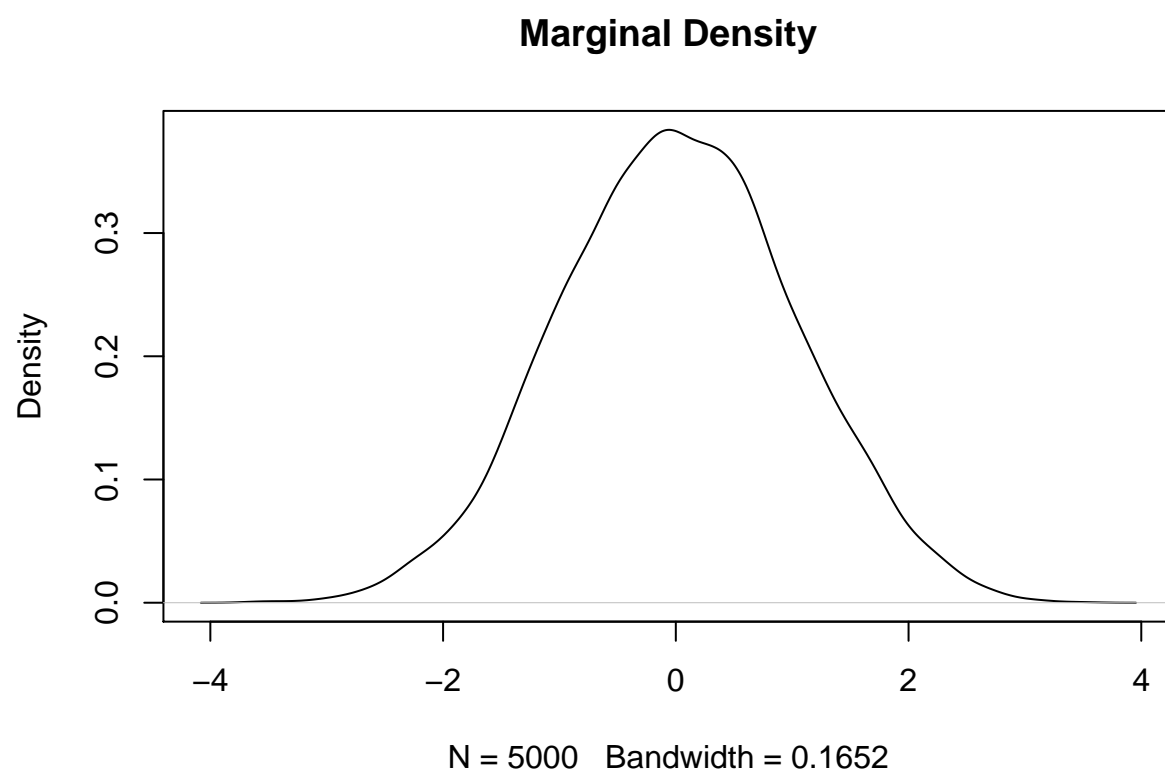
Sampling from the Banana Distribution

a)

Bayesian Adaptive Lasso

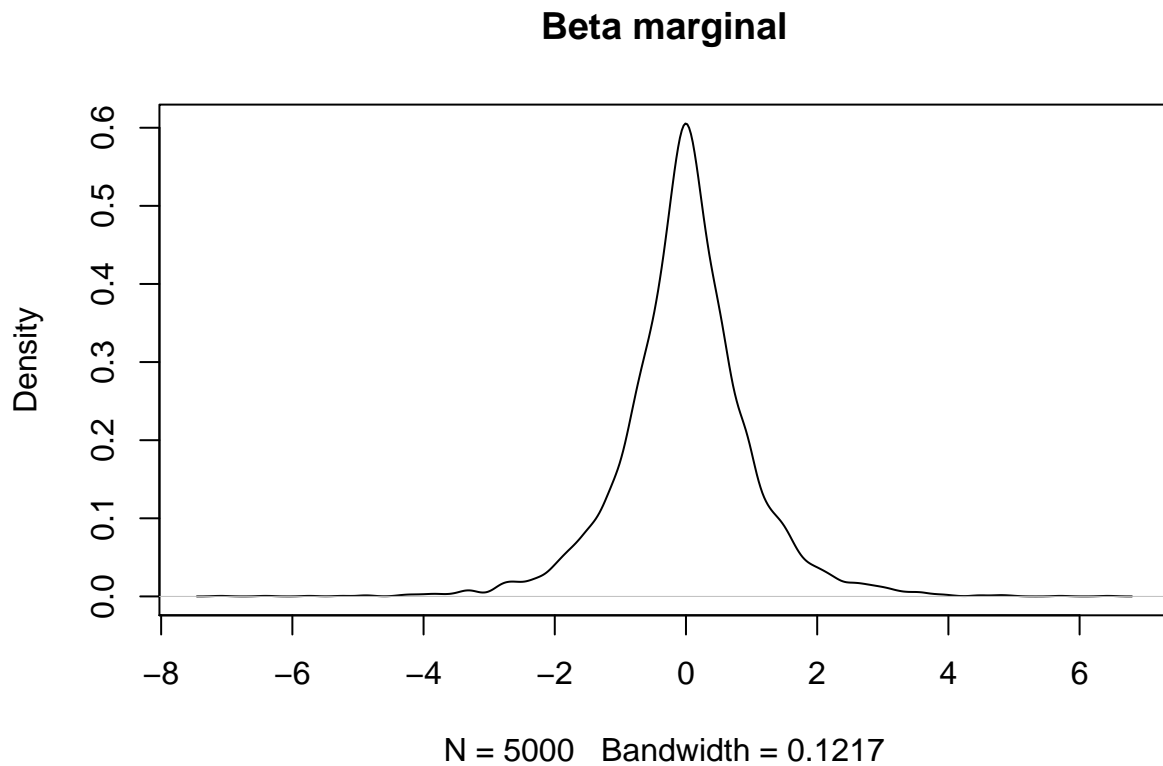
#a)

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



b)

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

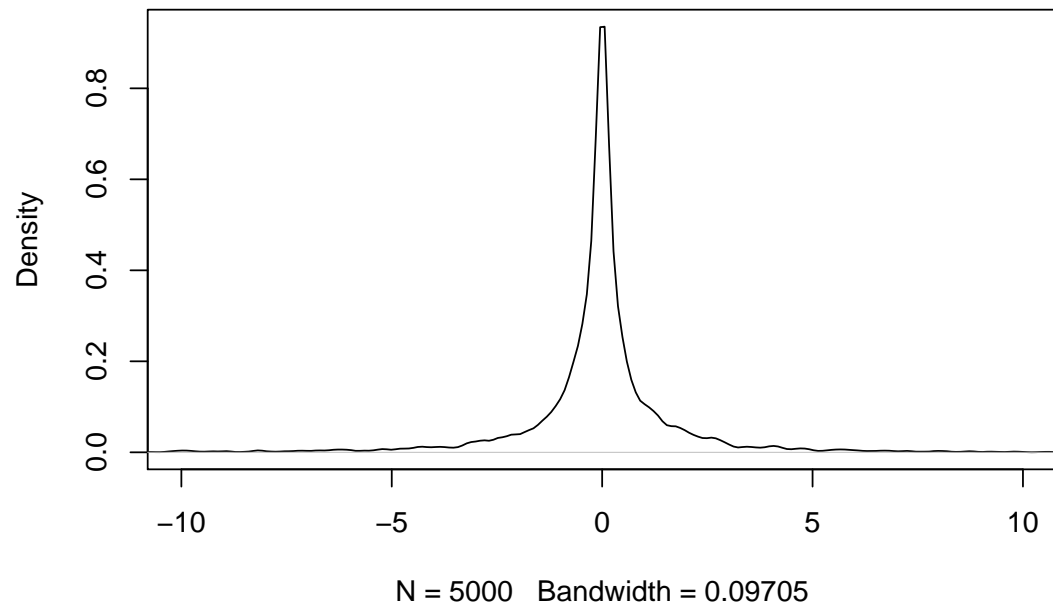


c)

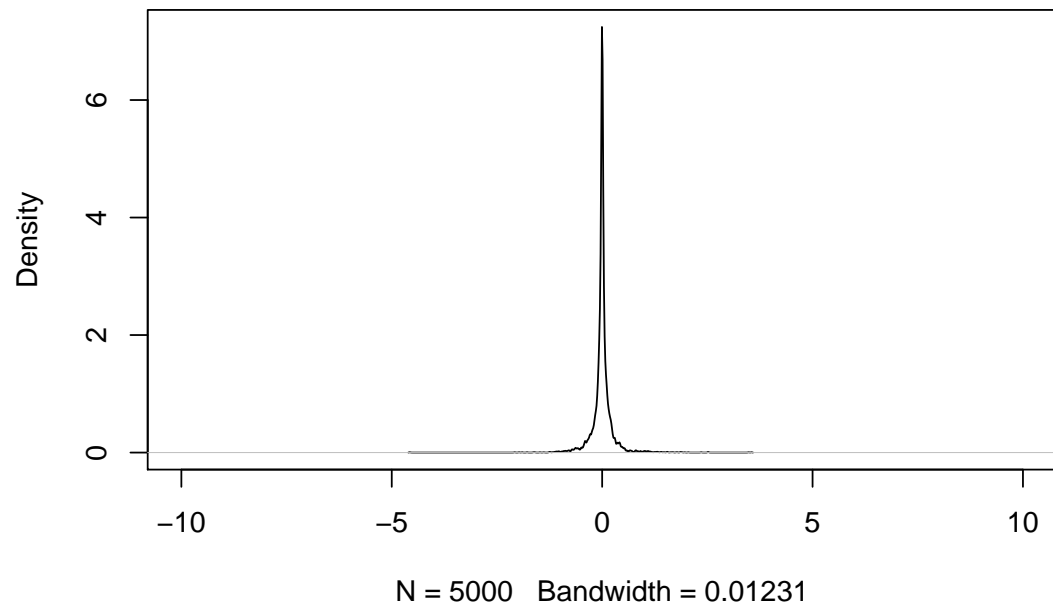
```
bvec <- c(1, 8, 20, 10000)
marginalplot <- function(n, b){
  lambda <- 1/rgamma(n, 1, b)
  tau2 <- rgamma(n, shape = 1, rate = lambda^2/2)
  sim <- rnorm(n, 0, sqrt(tau2))
  plot <- plot(density(sim),
               main = paste0("Beta marginal, b = ", b),
               xlim = c(-10, 10))
  save_plot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

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

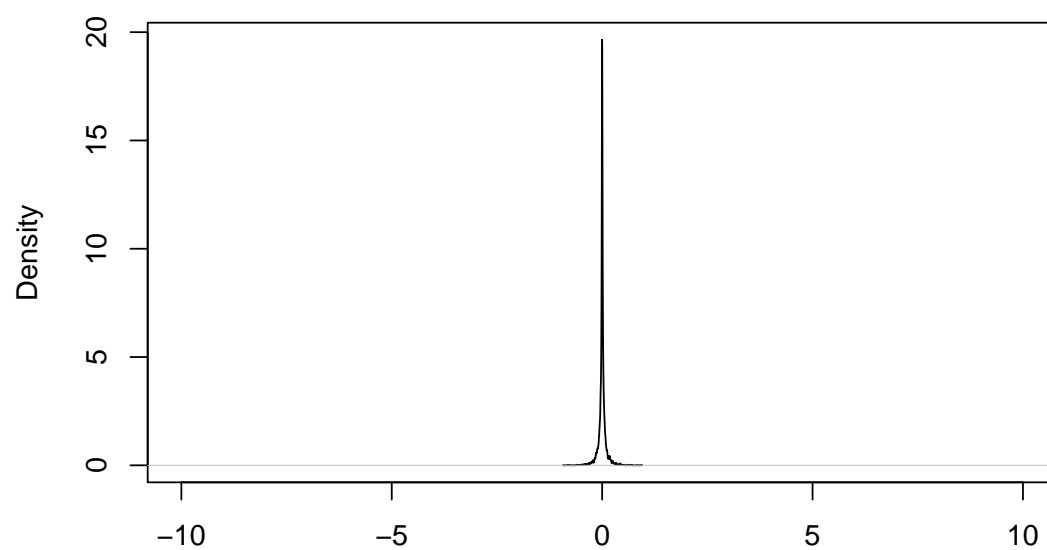
Beta marginal, $b = 1$



Beta marginal, $b = 8$

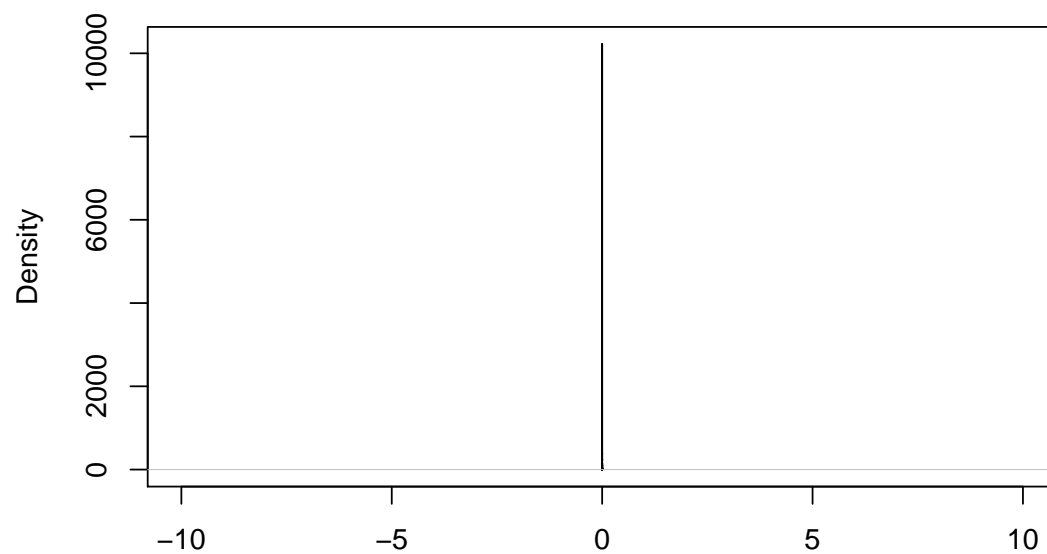


Beta marginal, $b = 20$



N = 5000 Bandwidth = 0.004802

Beta marginal, $b = 10000$



N = 5000 Bandwidth = 9.635e-06

e)

I will implement an MH + Gibbs Sampler algorithm to sample from the posterior distributions.

```
mh.gibbs <- function(y = y, X = Xdat,
                    n.sim = 1000, burn = 0.1,
                    a = 1, b = 1, beta, lambda2){
  #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(!is.na(lambda2)){
      lambda2 <- lambda2
    } else {
      lambda2 <- 1/rgamma(1, a, rate = b)
    }
    #tau2
    tau2 <- -(2/lambda2) * log(runif(10))
    #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")
Xdat <- diabetes$x
y <- diabetes$y
betahat <- solve(t(Xdat) %*% Xdat) %*% t(Xdat) %*% y
out <- mh.gibbs(y = y, X = Xdat, n = 10000, burn = 0.1,
```

My MCMC	Glmnet
-9.905013	-9.057331
-239.685839	-238.879817
519.840170	520.885704
324.356903	323.449275
-772.232192	-680.104304
460.963565	390.311899
92.112404	48.691747
175.636451	159.307982
743.874589	710.278264
67.628485	67.555464

```

a = 1, b = 1, beta = betahat, lambda2 = NA)

coefffun <- apply(out$beta, 2, function(x){quantile(x,c(0.16,0.5,0.84))})
coefme <- coefffun[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(-5, 5, 0.1))
out <- lapply(lambdas, function(x)mh.gibbs(y = y, X = Xdat,
      n.sim = 1000, burn = 0.1,
      a = 1, b = 1, beta = betahat, lambda2 = x))
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

```

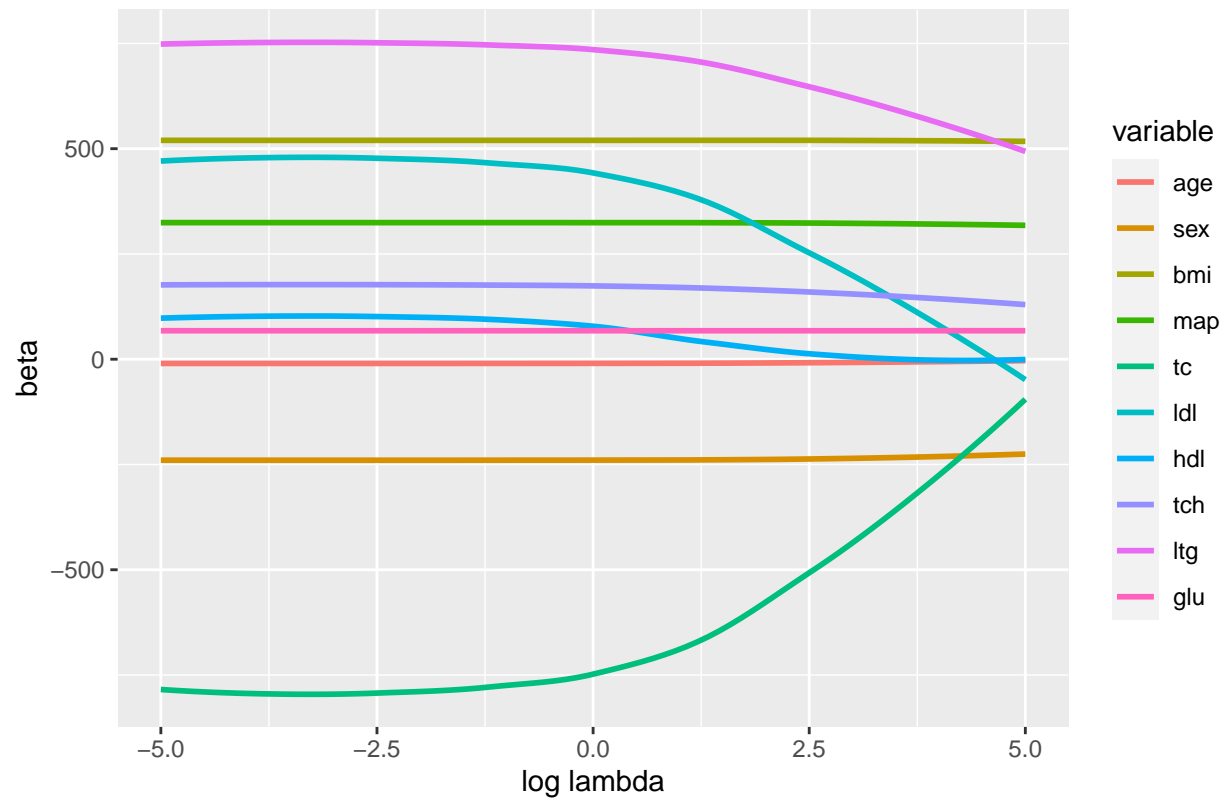
df$lambda <- rep(seq(-5, 5, 0.1), 10)

ggplot(data = df, aes(lambda, value, color = variable)) +
  geom_smooth(method = loess, se = FALSE) +
  ggtitle("Custom MCMC path") + xlab("log lambda") + ylab("beta")

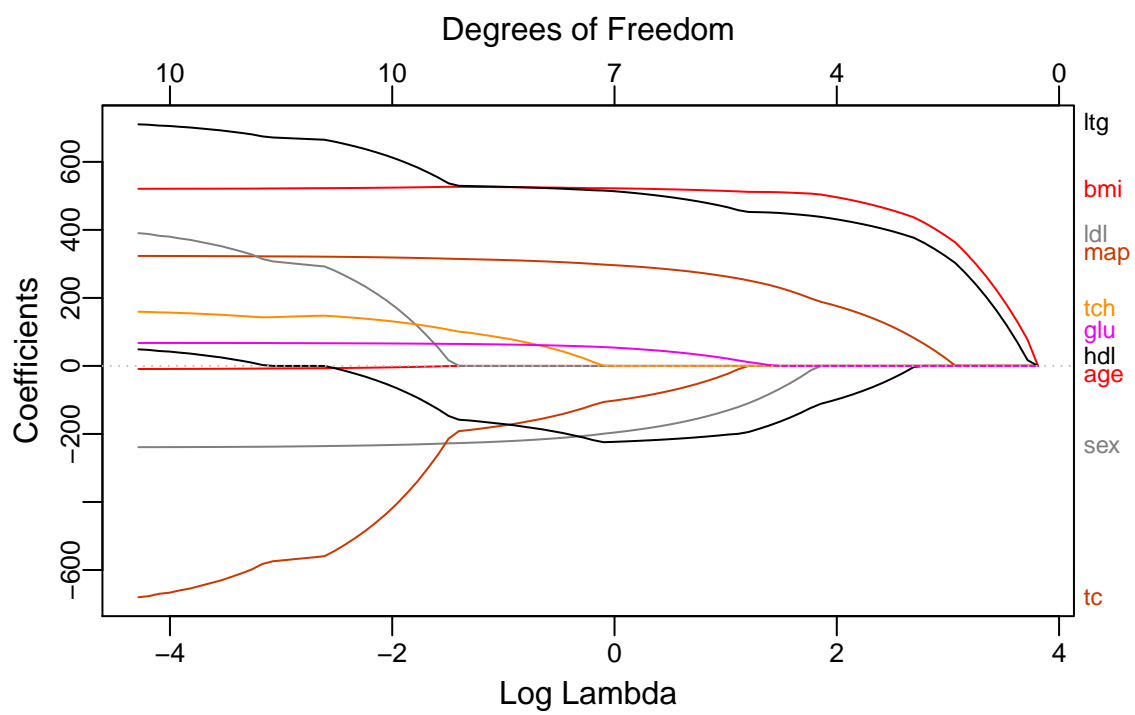
```

'geom_smooth()' using formula 'y ~ x'

Custom MCMC path



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



09