Portfolio 8

Rachel Wood

2023-04-26

For this portfolio, we use the Pima Indians Diabetes dataset:

```
library(mlbench)
data("PimaIndiansDiabetes")
head(PimaIndiansDiabetes)
```

```
##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35       0 33.6    0.627  50      pos
## 2        1      85       66      29       0 26.6    0.351  31      neg
## 3        8     183       64       0       0 23.3    0.672  32      pos
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74       0       0 25.6    0.201  30      neg
```

where we model the response y_i is the `diabetes` variable and model it using a logistic regression:

$$\mathbb{P}_{\alpha,\beta}(Y_i = 1) = \frac{1}{1 + e^{-\alpha - \beta^T x_i}}$$

which gives the likelihood function

$$L_n(\alpha, \beta) = \prod_{i=1}^{n} \mathbb{P}_{\alpha,\beta}(Y_i = y_i)$$

Then the posterior is

$$\pi(\alpha, \beta | y) \propto L_n(\alpha, \beta)\pi(\alpha, \beta)$$

with the log posterior given by

$$\log \pi(\alpha, \beta | y) = \sum_{i=1}^{n} \left(y_i \log \frac{1}{1 + e^{-\alpha - \beta^T x_i}} + (1 - y_i) \log \frac{e^{-\alpha - \beta^T x_i}}{1 + e^{-\alpha - \beta^T x_i}} \right) + \pi(\alpha, \beta)$$

where $\pi(\alpha, \beta)$ is the prior.

1. Choosing a proposal distribution

Before applying the MH algorithm, we need to choose a proposal distribution Q. For this we use

$$Q(z, dz') = \mathcal{N}_{p+1}(z, c\mathbf{\Sigma}_n)$$

for tuning parameter $c > 0$ and

$$\mu_n = \underset{(\alpha,\beta)\in\mathbb{R}^p}{\arg\max} \log \pi(\alpha, \beta | y), \qquad \mathbf{\Sigma}_n = -(\mathbf{H}_n(\mu_q))^{-1}$$

Since $\mathbf{\Sigma}_n$ can be seen as an estimate of the variance of the covariance matrix of (α, β) under the posterior. We can then use the `glm()` function to get this:

```
data <- PimaIndiansDiabetes
y <- as.numeric((data$diabetes))-1
X <- scale(data[,1:8])

fit <- glm(y ~ X, family = "binomial")
z_0 <- fit$coefficients
Sigma_n <- summary(fit)$cov.scaled
```

2. Implementing the MH algorithm

For the acceptance probability at each iteration, we also need a function to compute the posterior likelihood for the new and old parameter values. For this we choose a flat prior for simplicity:

```
library(mvtnorm)
post_likelihood <- function(par, X, y){

  alpha <- par[1]
  beta <- par[2:9]
  p <- 1 - 1 / (1 + exp(alpha + beta%*%t(X)))
  likelihood <- exp(sum(dbinom(y, size=1, prob=p, log=TRUE)) )

  return(likelihood)
}
```

Since we now have a proposal distribution and a posterior likelihood, we can now run the MH algorithm:

```
logistic_MH <- function(z_0,c, sigma_n, tmax, X, y){
  zs <- matrix(NA, nrow =tmax, ncol = length(z_0))
  z_current <- z_0
  z_current_ll <- post_likelihood(z_current, X,y)
  sigma_prime <- c * sigma_n
  accepted <- 0
  for (i in 1:tmax){
    z_new <- rmvnorm(1, z_current, sigma_prime)
    z_new_ll <- post_likelihood(z_new, X, y)
    alpha <- log(z_new_ll) +  dmvnorm(z_current, z_new, sigma_prime, log = TRUE)- log(z_current_ll) - dm
    if (runif(1) < min(1,exp(alpha))){

      z_current <- z_new
      z_current_ll <- z_new_ll
      accepted <- accepted + 1

    }
    zs[i,] <- z_current
  }
  return(list(zs = zs, acceptance_rate = accepted/tmax))
}
```

3. Convergence

We now run the algorithm with $c = 1$ and test for convergence.

```
mh<- logistic_MH(z_0, 1, Sigma_n, 10000, X, y)
```

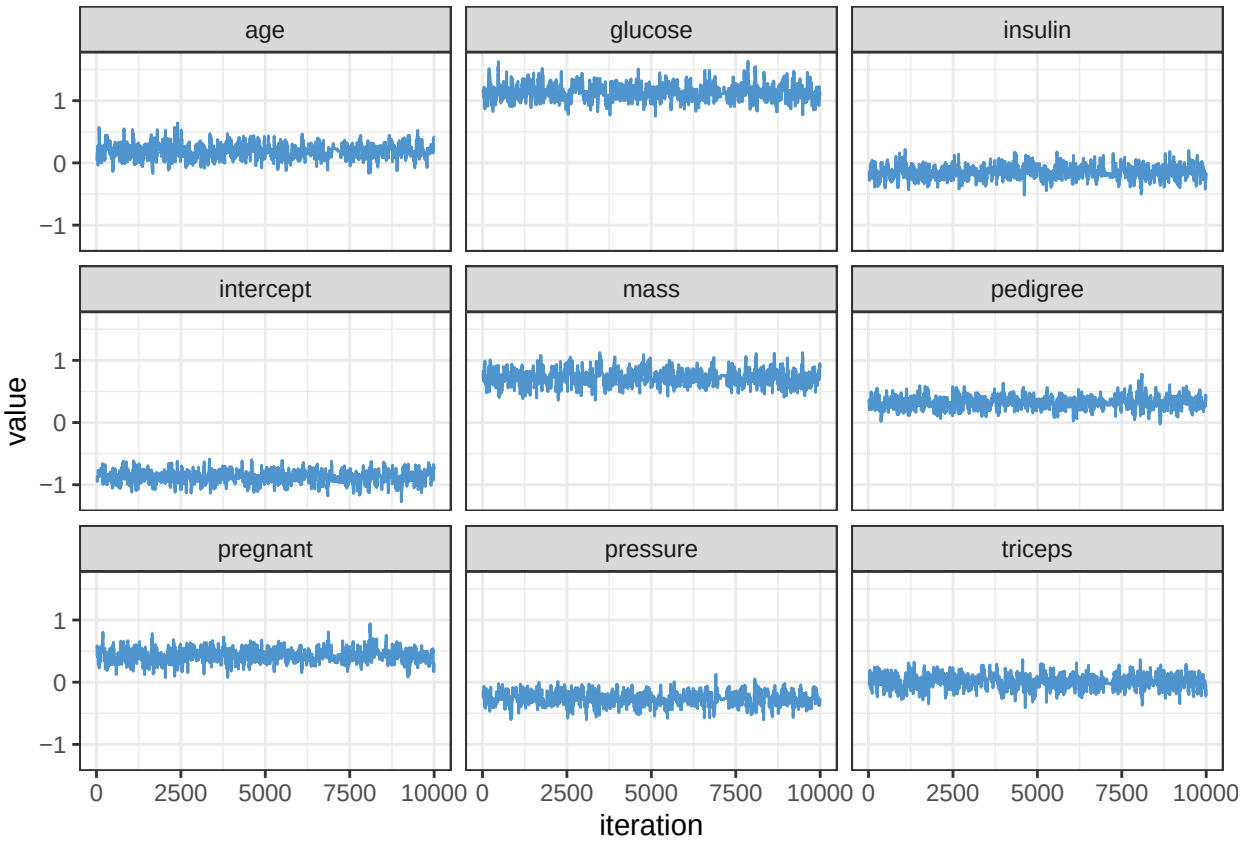
We get the acceptance rate to be:

```
mh$acceptance_rate
```

```
## [1] 0.1718
```

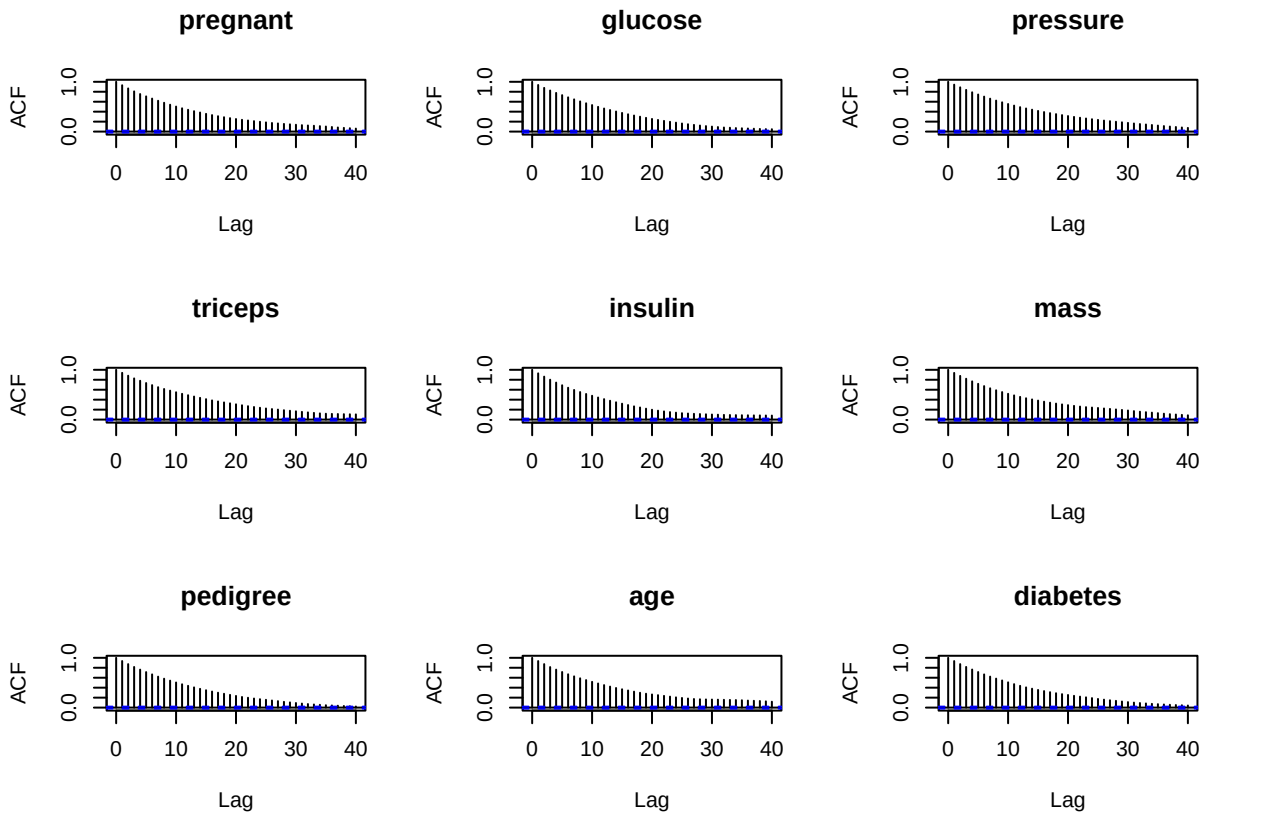
We then produce trace plots:

```
library(dplyr)
library(tidyr)
library(ggplot2)

colnames(mh$zs) <- c("intercept", colnames(data)[1:8])
zs_plot <- mh$zs %>%
  as_tibble() %>%
  mutate(iteration = 1:10000) %>%
  pivot_longer(cols = 1:9)

ggplot(zs_plot, aes(iteration, value)) +
  geom_line(colour = "steelblue3") +
  facet_wrap(vars(name))
```



We also produce some autocorrelation plots:

```
par(mfrow = c(3,3))
for (i in 1:9){
  acf(mh$zs[,i], main = colnames(data)[i])
}
```



From these results, we see the acceptance ratio is quite low and the autocorrelation of our samples decreases quite slowly.

4. Modifying Q

We then try the above again with a smaller c in the hopes that this will improve our convergence:

```
mh_new<- logistic_MH(z_0, 0.75, Sigma_n, 10000, X, y)
```

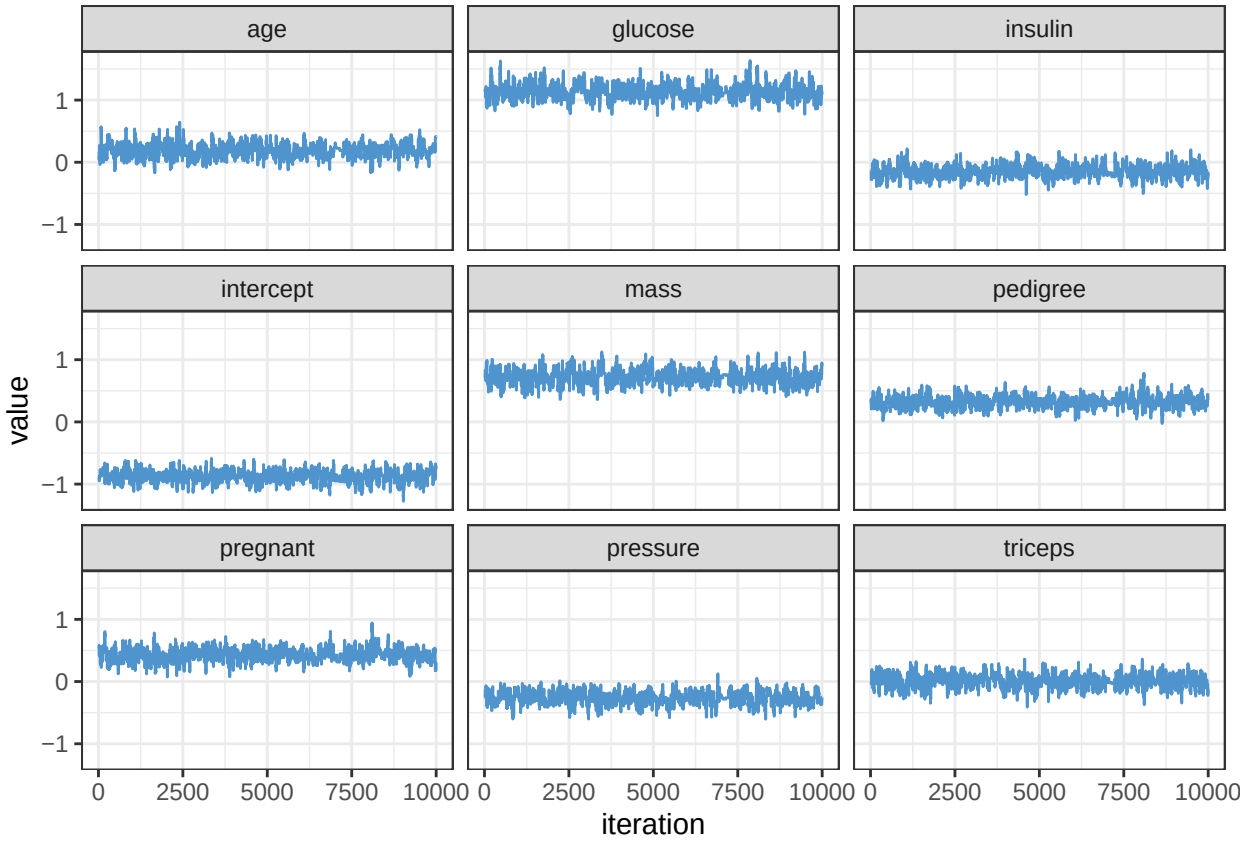
The new acceptance rate is then:

```
mh_new$acceptance_rate
```

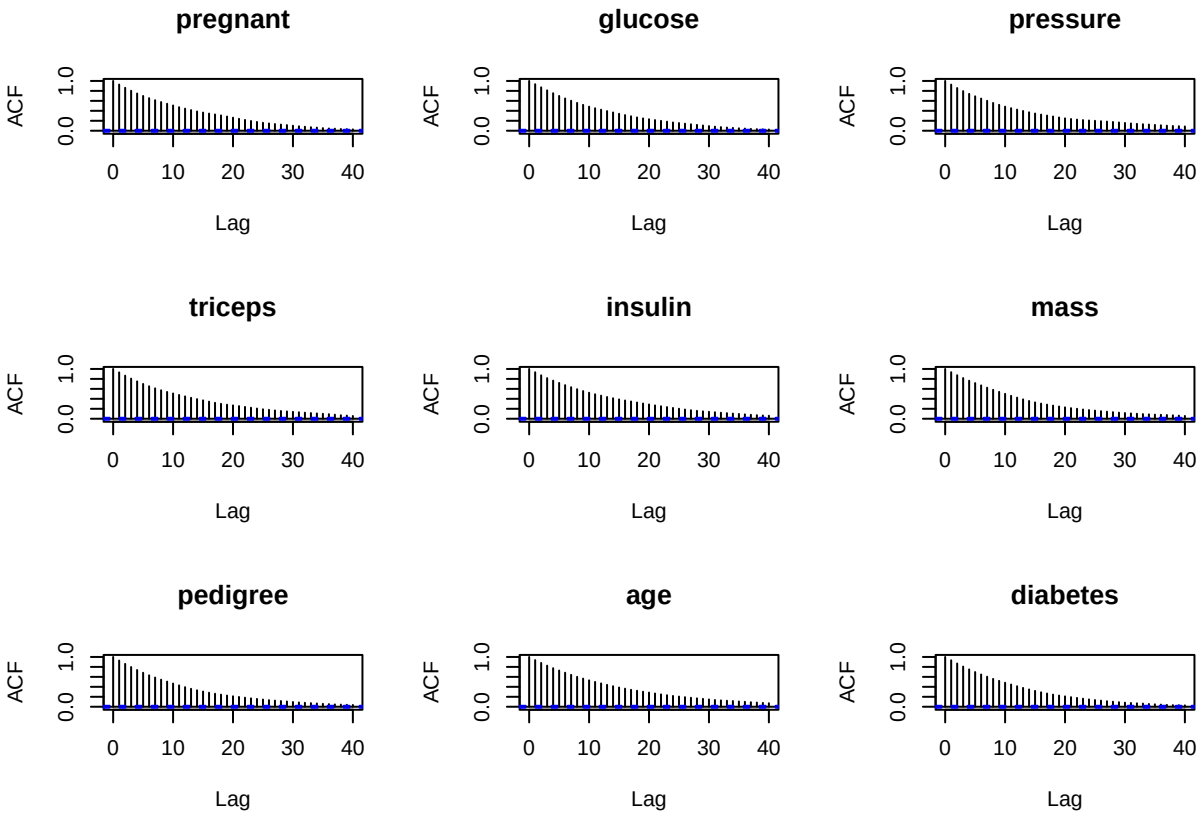
```
## [1] 0.228
```

We then produce trace plots and autocorrelation plots as before:

```
library(dplyr)
library(tidyr)
library(ggplot2)

colnames(mh_new$zs) <- c("intercept", colnames(data)[1:8])
zs_plot <- mh$zs %>%
  as_tibble() %>%
  mutate(iteration = 1:10000) %>%
  pivot_longer(cols = 1:9)

ggplot(zs_plot, aes(iteration, value)) +
  geom_line(colour = "steelblue3") +
  facet_wrap(vars(name))
```




```
par(mfrow = c(3,3))
for (i in 1:9){
  acf(mh_new$zs[,i], main = colnames(data)[i])
}
```



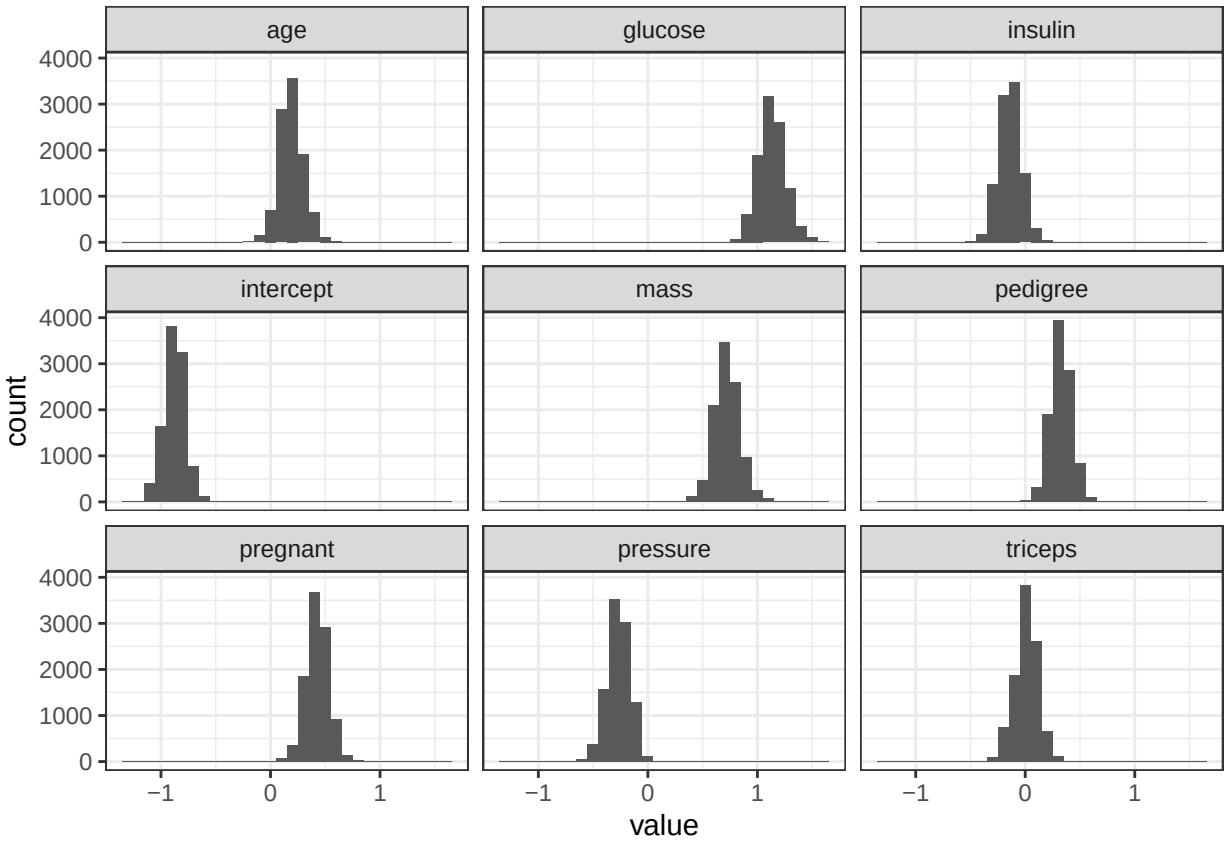
We can see from these results the acceptance rate is higher and the autocorrelation of samples decreases more quickly.

5. Marginal posterior distributions

We can now also plot the approximate marginal posteriors for each parameter using histograms:

```
ggplot(zs_plot, aes(value)) +
  geom_histogram() +
  facet_wrap(vars(name))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


age
glucose
insulin
intercept
mass
pedigree
pregnant
pressure
triceps
count
value
0
1000
2000
3000
4000
−1
0
1