

05-08-2023

Hengde Ouyang

2023-05-08

## Function

```
# Compute theta
THETA <- function(A,beta){
  n = dim(A)[1]
  Abeta_bar = A%*%beta-(matrix(1,n,n)%*%A%*%beta)/n
  theta = Abeta_bar/as.numeric(t(Abeta_bar)%*%Abeta_bar)
  theta[is.nan(theta)] = 0
  return(theta)
}
```

```
# Compute C-index type loss function
HarrellC <- function(Y, delta, tau,theta) {
  n <- length(Y)
  Wmat <- matrix(0, n, n)
  Thetamat <- matrix(0,n,n)
  for(i in 1:n) {
    Thetamat[i,] <- (theta[i]> theta)*1
    if(delta[i] != 0) {
      Wmat[i,] <- delta[i]*(Y[i] < Y)*(Y[i] < tau)
    }
  }

  Wmat <- Wmat/sum(Wmat)

  # Case that theta are all 0
  if (sum(Wmat*Thetamat) == 0){
    return (0.0001)
  }

  return(sum(Wmat*Thetamat))
}
```

```
library(mvtnorm)
```

```
MH_Sampling <- function(Y,delta,tau,A,beta0,sigma20,var.prop,m){
```

```

accept = 0
BETA = matrix(0,m,dim(A)[2])
beta = beta0

for (i in 1:m){

  # Sample beta
  beta.p = t(rmvnorm(1,beta,var.prop))

  theta.p = THETA(A,beta.p)
  theta = THETA(A,beta)

  # Compute log of MH ratio
  lrMH = log(HarrellC(Y, delta, tau,theta.p)) +
    sum(dnorm(beta.p,beta0,sigma20,log=T))-
    log(HarrellC(Y, delta, tau,theta)) -
    sum(dnorm(beta,beta0,sigma20,log=T))

  if (log(runif(1))<lrMH){
    beta = beta.p
    accept = accept + 1
  }
  BETA[i,] = beta
}

return(list(BETA=BETA,accept_rate=accept/m))
}

```

## Simulation

```

library(survival)
data(cancer)

Y = gbsg$rfstime
delta=gbsg$status
tau = 2500
A <- model.matrix(rfstime ~ age + meno + size + grade + er + hormon,data=gbsg)
beta0 = rep(0,dim(A)[2])
sigma20 = rep(100,dim(A)[2])

# "In a normal regression problem, the posterior variance of beta
# will be close to .. (Page 180) "
var.prop = var(Y)*solve(t(A)%*%A)
m = 5000

system.time({

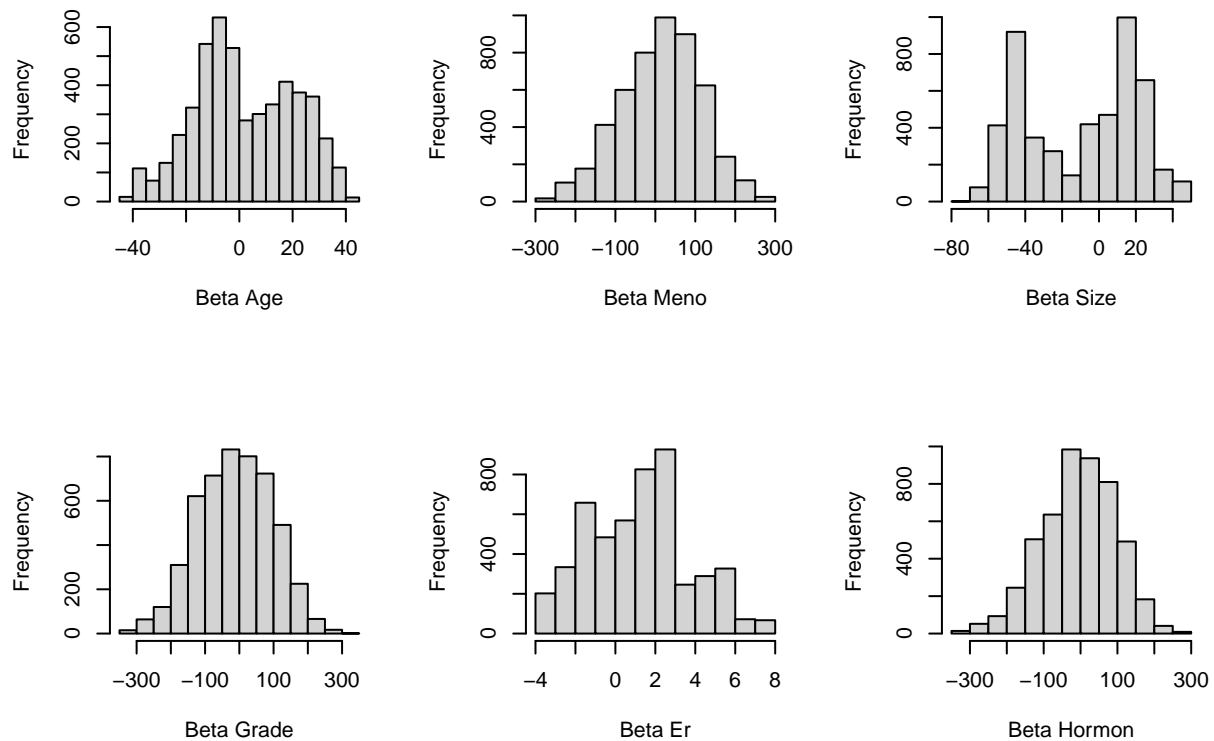
```

```
result1 = MH_Sampling(Y,delta,tau,A,beta0,sigma20,var.prop,m)
})
```

```
##    user  system elapsed
## 347.86   40.55   732.22
```

Result for 5000 iterations

```
par(mfrow=c(2,3))
hist(result1$BETA[,2],xlab = "Beta Age",main = "")
hist(result1$BETA[,3],xlab = "Beta Meno",main = "")
hist(result1$BETA[,4],xlab = "Beta Size",main = "")
hist(result1$BETA[,5],xlab = "Beta Grade",main = "")
hist(result1$BETA[,6],xlab = "Beta Er",main = "")
hist(result1$BETA[,7],xlab = "Beta Hormon",main = "")
```



This is the simulation result with acceptance rate:

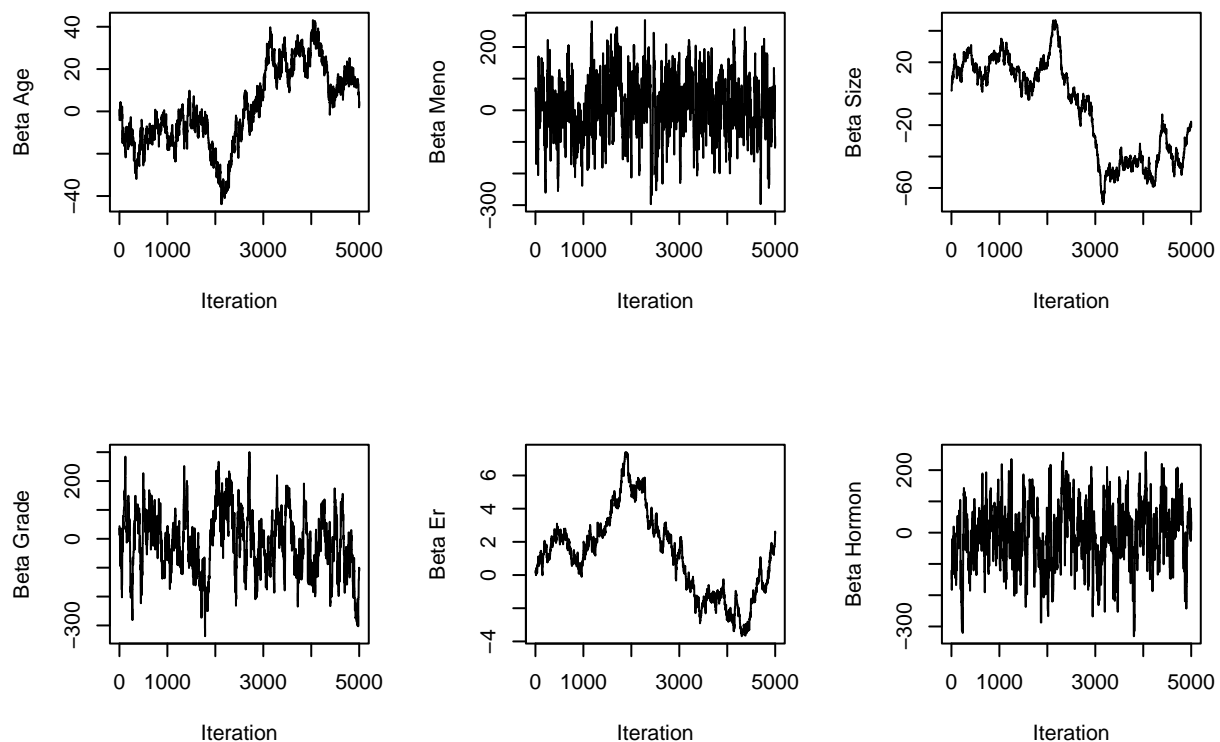
```
print(result1$accept_rate)
```

```
## [1] 0.3844
```

```

par(mfrow=c(2,3))
plot(1:5000,result1$BETA[,2],xlab = "Iteration",
     ylab = "Beta Age",type = "l")
plot(1:5000,result1$BETA[,3],xlab = "Iteration",
     ylab = "Beta Meno",type = "l")
plot(1:5000,result1$BETA[,4],xlab = "Iteration",
     ylab = "Beta Size",type = "l")
plot(1:5000,result1$BETA[,5],xlab = "Iteration",
     ylab = "Beta Grade",type = "l")
plot(1:5000,result1$BETA[,6],xlab = "Iteration",
     ylab = "Beta Er",type = "l")
plot(1:5000,result1$BETA[,7],xlab = "Iteration",
     ylab = "Beta Hormon",type = "l")

```



## Simulation 2

```

# Adjust the prior variance for
sigma20_2 = rep(1000,dim(A)[2])
result2 = MH_Sampling(Y,delta,tau,A,beta0,sigma20_2,var.prop,m)

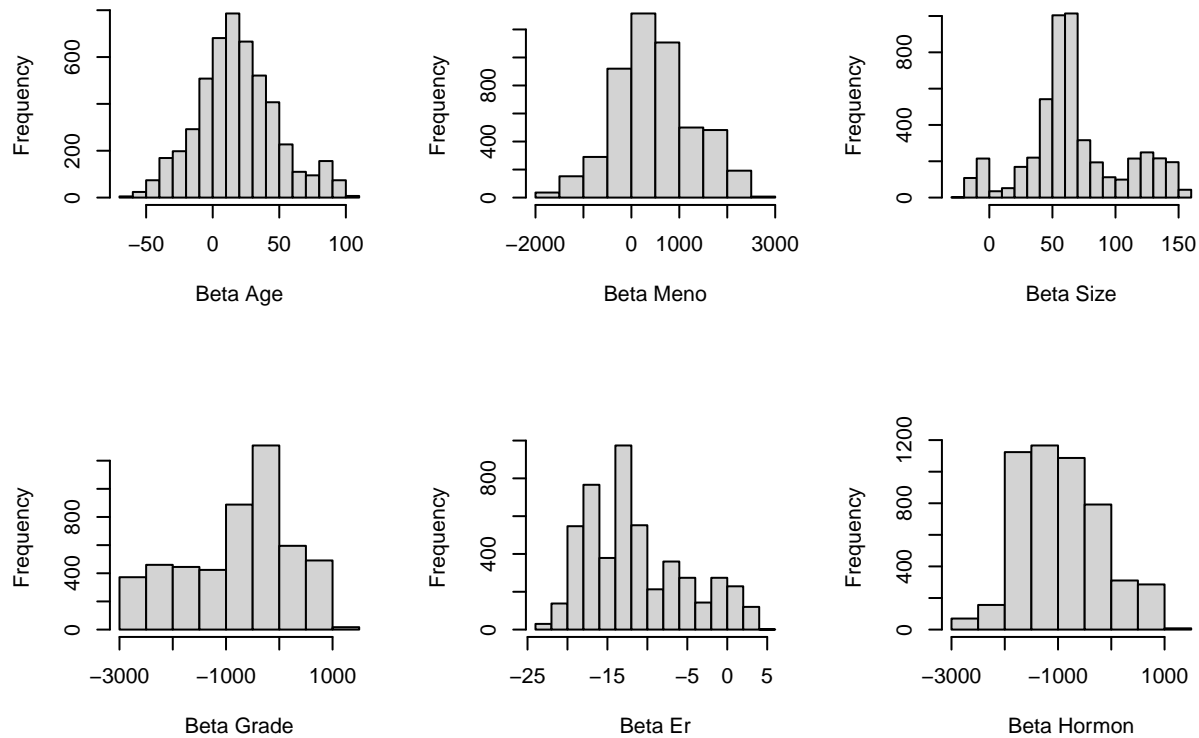
```

```

par(mfrow=c(2,3))
hist(result2$BETA[,2],xlab = "Beta Age",main = "")
hist(result2$BETA[,3],xlab = "Beta Meno",main = "")

```

```
hist(result2$BETA[,4],xlab = "Beta Size",main = "")
hist(result2$BETA[,5],xlab = "Beta Grade",main = "")
hist(result2$BETA[,6],xlab = "Beta Er",main = "")
hist(result2$BETA[,7],xlab = "Beta Hormon",main = "")
```

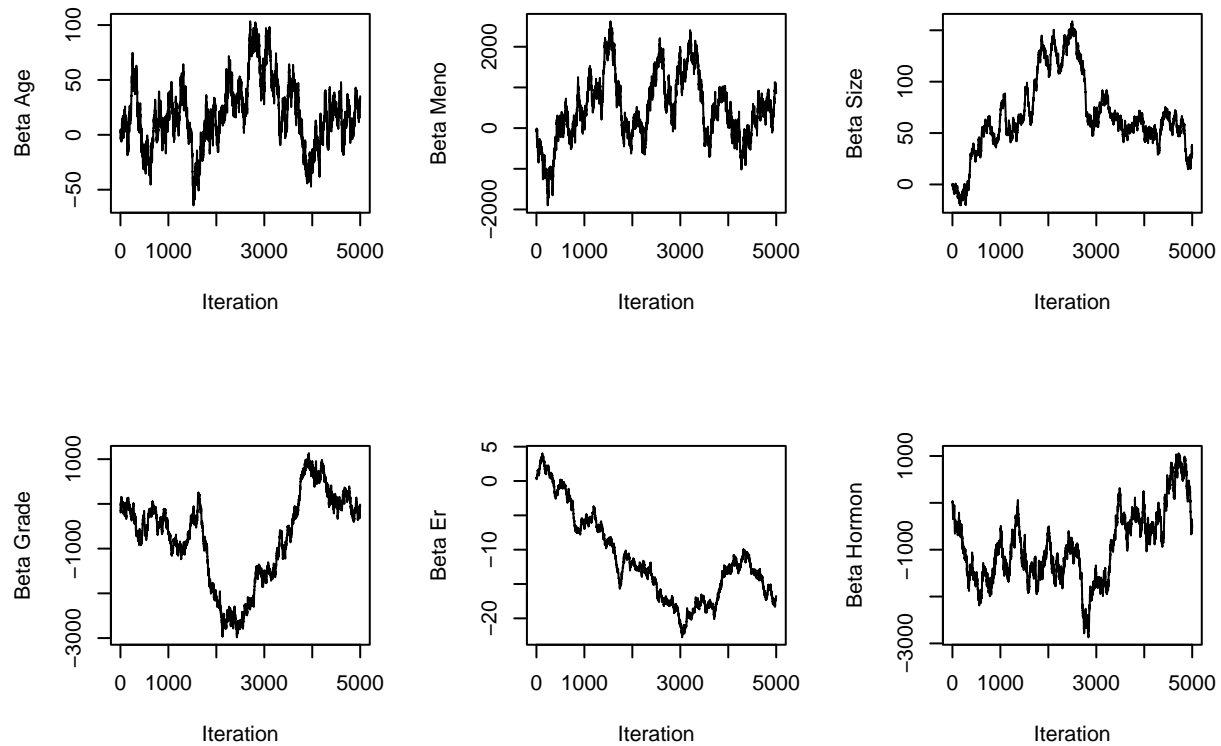


This is the simulation result with acceptance rate:

```
print(result2$accept_rate)
```

```
## [1] 0.9228
```

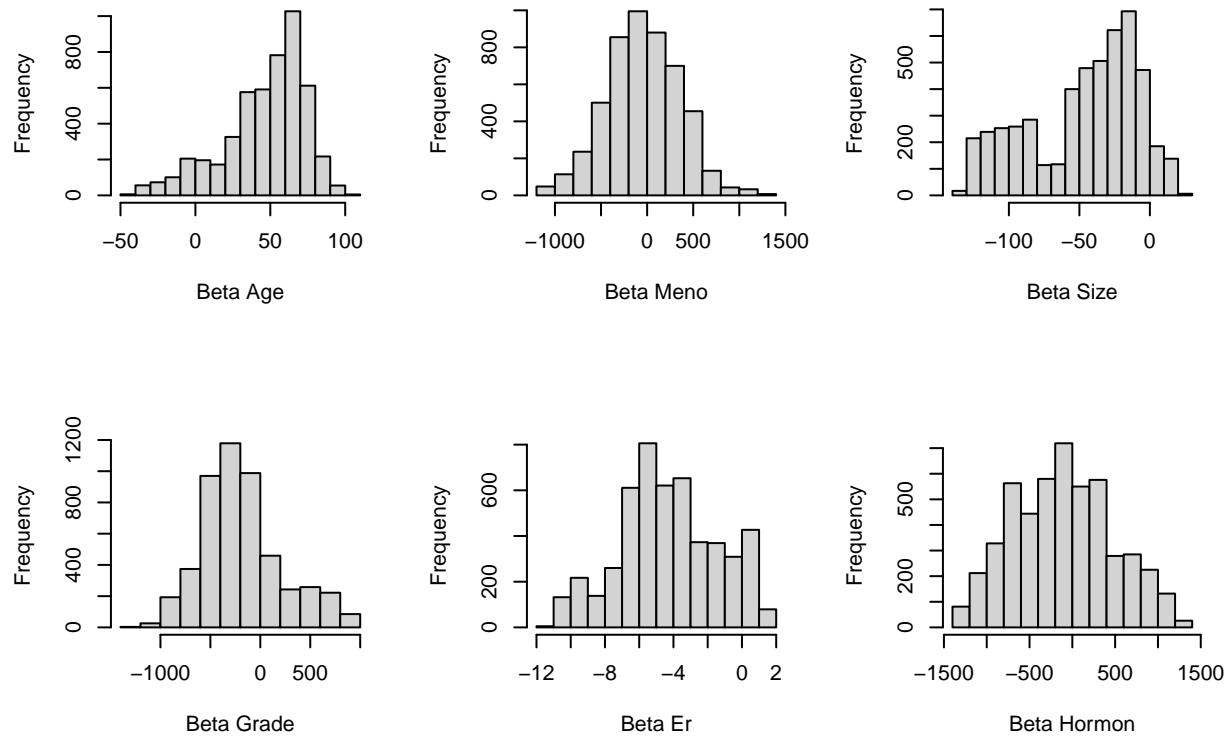
```
par(mfrow=c(2,3))
plot(1:5000,result2$BETA[,2],xlab = "Iteration",
     ylab = "Beta Age",type = "l")
plot(1:5000,result2$BETA[,3],xlab = "Iteration",
     ylab = "Beta Meno",type = "l")
plot(1:5000,result2$BETA[,4],xlab = "Iteration",
     ylab = "Beta Size",type = "l")
plot(1:5000,result2$BETA[,5],xlab = "Iteration",
     ylab = "Beta Grade",type = "l")
plot(1:5000,result2$BETA[,6],xlab = "Iteration",
     ylab = "Beta Er",type = "l")
plot(1:5000,result2$BETA[,7],xlab = "Iteration",
     ylab = "Beta Hormon",type = "l")
```



### Simulation 3

```
# Adjust the prior variance for
sigma20_3 = rep(500,dim(A)[2])
result3 = MH_Sampling(Y,delta,tau,A,beta0,sigma20_3,var.prop,m)
```

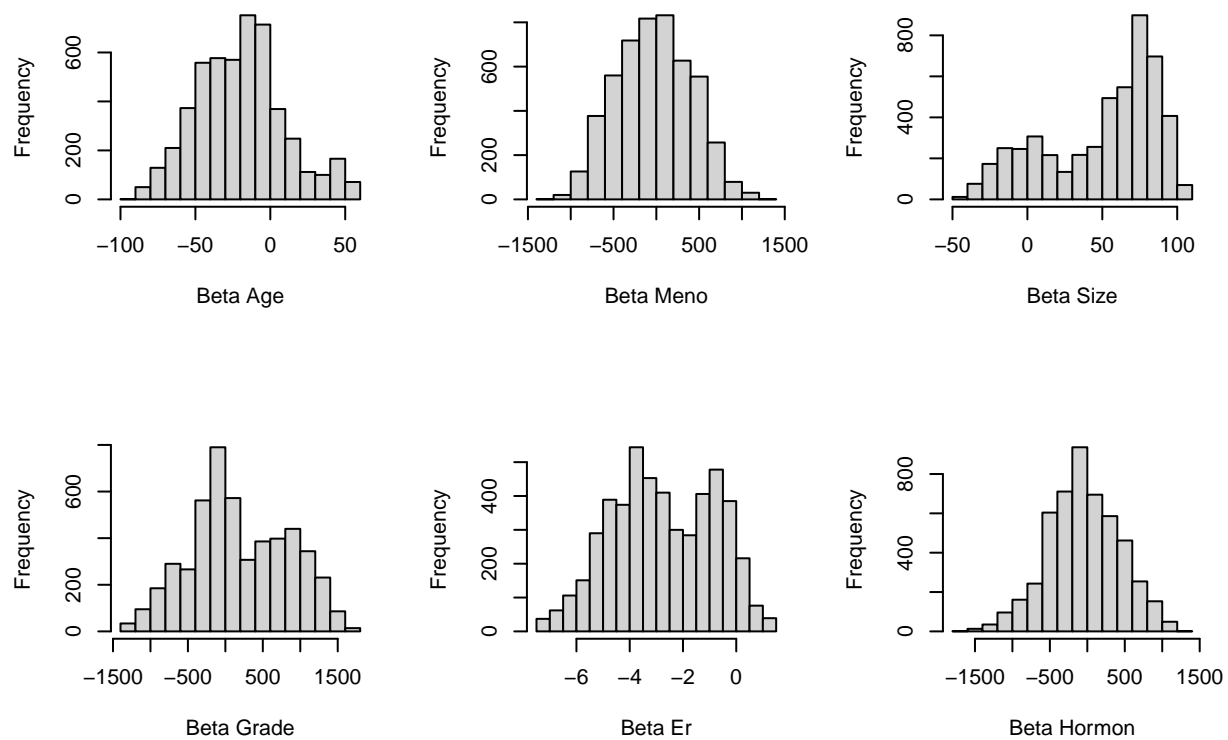
```
par(mfrow=c(2,3))
hist(result3$BETA[,2],xlab = "Beta Age",main = "")
hist(result3$BETA[,3],xlab = "Beta Meno",main = "")
hist(result3$BETA[,4],xlab = "Beta Size",main = "")
hist(result3$BETA[,5],xlab = "Beta Grade",main = "")
hist(result3$BETA[,6],xlab = "Beta Er",main = "")
hist(result3$BETA[,7],xlab = "Beta Hormon",main = "")
```



### Simulation 3

```
# Adjust the prior variance for
sigma20_3 = rep(500,dim(A)[2])
result3 = MH_Sampling(Y,delta,tau,A,beta0,sigma20_3,var.prop,m)
```

```
par(mfrow=c(2,3))
hist(result3$BETA[,2],xlab = "Beta Age",main = "")
hist(result3$BETA[,3],xlab = "Beta Meno",main = "")
hist(result3$BETA[,4],xlab = "Beta Size",main = "")
hist(result3$BETA[,5],xlab = "Beta Grade",main = "")
hist(result3$BETA[,6],xlab = "Beta Er",main = "")
hist(result3$BETA[,7],xlab = "Beta Hormon",main = "")
```



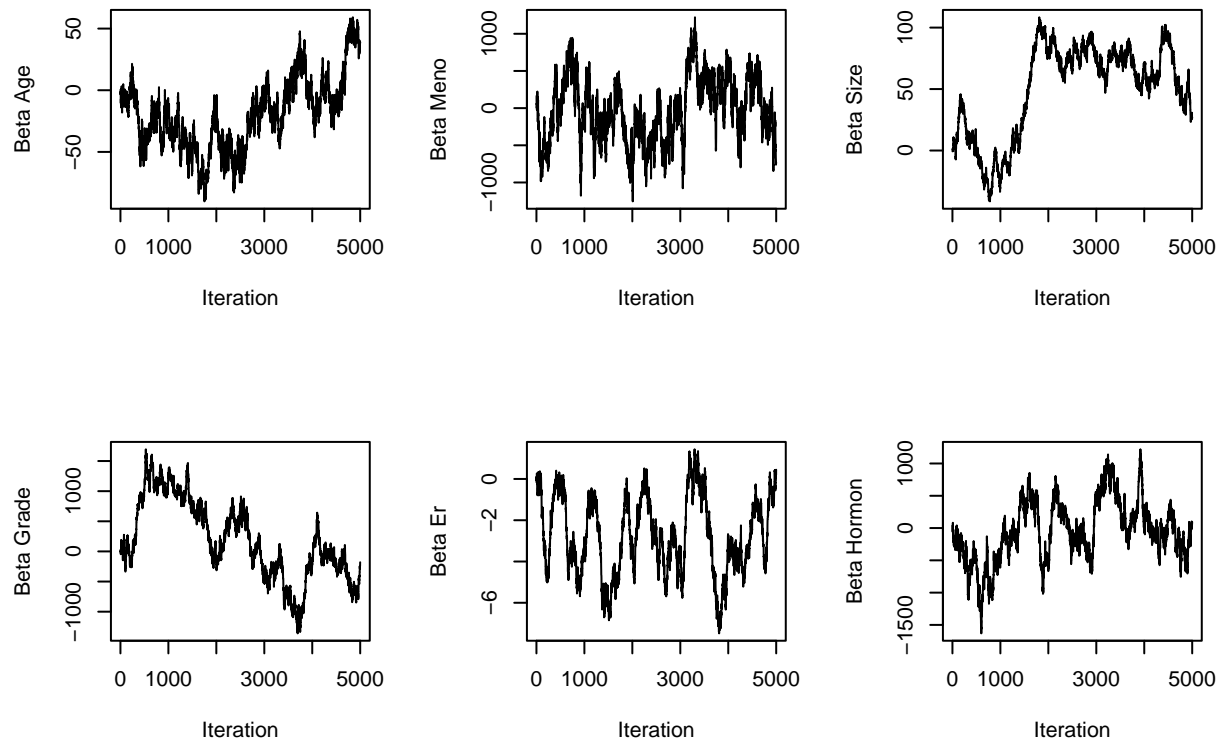
This is the simulation result with acceptance rate:

```
print(result3$accept_rate)
```

```
## [1] 0.8464
```

```
par(mfrow=c(2,3))
plot(1:5000,result3$BETA[,2],xlab = "Iteration",
     ylab = "Beta Age",type = "l")
plot(1:5000,result3$BETA[,3],xlab = "Iteration",
     ylab = "Beta Meno",type = "l")
plot(1:5000,result3$BETA[,4],xlab = "Iteration",
     ylab = "Beta Size",type = "l")
plot(1:5000,result3$BETA[,5],xlab = "Iteration",
     ylab = "Beta Grade",type = "l")
plot(1:5000,result3$BETA[,6],xlab = "Iteration",
     ylab = "Beta Er",type = "l")
plot(1:5000,result3$BETA[,7],xlab = "Iteration",
     ylab = "Beta Hormon",type = "l")
```





## Findings

1. Simulation is very time consuming. (It takes a long time even if we only choose  $m = 1000$ )

## Others

(I also try prior variance 10, but the result is strange)