Take home midterm exam - Bayesian Econometrics

February 15, 2020

1 Take home midterm exam

2 Igor Ferreira Batista Martins

2.1 Item A

We need to show that $p(\epsilon_i|\sigma^2) = \int_0^\infty p(\epsilon_i|\lambda_i, \sigma^2)p(\lambda_i)d\lambda_i = \frac{1}{2\sigma}exp\{\frac{-|\epsilon_i|}{\sigma}\}$.

$$p(\epsilon_i|\sigma^2) = \int_0^\infty p(\epsilon_i|\lambda_i, \sigma^2)p(\lambda_i)d\lambda_i$$

Since $p(\epsilon_i|\lambda_i, \sigma^2) \sim N(0, \lambda_i\sigma^2)$ and $\lambda_1, ..., \lambda_n \sim Exponential(1/2)$ then

$$\int_0^\infty (2\pi\lambda_i\sigma^2)^{-1/2} exp\Big\{\frac{-1}{2}\frac{\epsilon_i^2}{\lambda_i\sigma^2}\Big\}\frac{1}{2}exp\Big\{\frac{-\lambda_i}{2}\Big\}d\lambda_i$$

Rewriting $(2\pi\lambda_i\sigma^2)^{-1/2}$ as $(2\pi\sigma^2)^{-1/2}\lambda_i^{-1/2}$

$$\int_0^\infty (2\pi\sigma^2)^{-1/2}\lambda_i^{-1/2} exp\Big\{\frac{-1}{2}\frac{\epsilon_i^2}{\lambda_i\sigma^2}\Big\}\frac{1}{2}exp\Big\{\frac{-\lambda_i}{2}\Big\}d\lambda_i$$

Since $\frac{(2\pi\sigma^2)^{-1/2}}{2}$ does not depend on λ_i, we can write

$$\frac{(2\pi\sigma^2)^{-1/2}}{2}\int_0^\infty \lambda_i^{-1/2} exp\Big\{\frac{-1}{2}\frac{\epsilon_i^2}{\lambda_i\sigma^2}\Big\}exp\Big\{\frac{-\lambda_i}{2}\Big\}d\lambda_i$$

Using the exponential properties

$$\frac{(2\pi\sigma^2)^{-1/2}}{2}\int_0^\infty \lambda_i^{-1/2} exp\Big\{\frac{-1}{2}\Big(\frac{\epsilon_i^2}{\lambda_i\sigma^2} + \lambda_i\Big)\Big\}d\lambda_i$$

Let $\psi_i = \lambda_i^{1/2}$. Note that the integration limits does not change and that

$$d\lambda_i = 2\psi_i d\psi_i$$

Then,

$$\frac{(2\pi\sigma^2)^{-1/2}}{2}\int_0^\infty \psi_i^{-1} exp\Big\{\frac{-1}{2}\Big(\frac{\epsilon_i^2}{\psi_i^2\sigma^2}+\psi_i^2\Big)\Big\}2\psi_i d\psi_i$$

Rearranging

$$(2\pi\sigma^2)^{-1/2}\int_0^\infty exp\Big\{\frac{-1}{2}\Big(\psi_i^2+\frac{\epsilon_i^2\psi_i^{-2}}{\sigma^2}\Big)\Big\}d\psi_i$$

Consider $a^2 = 1$ and $b^2 = \frac{\epsilon^2}{\sigma^2}$, then

$$(2\pi\sigma^2)^{-1/2}\int_0^\infty exp\Big\{\frac{-1}{2}\Big(a^2\psi^2+b^2\psi_i^-2\Big)\Big\}d\psi_i$$

Using hint 1, we obtain

$$(2\pi\sigma^2)^{-1/2}\frac{\pi}{2a^2}^{1/2} exp\{-|ab|\}$$

Substituting a and b

$$(2\pi\sigma^2)^{-1/2}\frac{\pi}{2}^{1/2} exp\Big\{-|\frac{\epsilon_i}{\sigma}|\Big\}$$

Therefore,

$$\frac{1}{2\sigma}exp\Big\{\frac{-|\epsilon_i|}{\sigma}\Big\}$$

Thus, we can conclude that
$p(\epsilon_i|\sigma^2) = \int_0^\infty p(\epsilon_i|\lambda_i,\sigma^2)p(\lambda_i)d\lambda_i = \frac{1}{2\sigma}exp\{\frac{-|\epsilon_i|}{\sigma}\}$.

2.2 Item B

The joint posterior distribution, $p(\beta, \sigma^2, \{\lambda_i\}|y, x)$, can be represented by using Bayes' rule as

$$p(\beta,\sigma^2,\{\lambda_i\}|y,x) \propto \Big(\prod_{i=1}^n \phi(y_i; x_i^T\beta, \lambda_i\sigma^2)p(\lambda_i)\Big)p(\beta)p(\sigma^2)$$

One posterior simulator capable of fitting the regression is based on two Gibbs sampling steps, one for the full conditional of β and the other one for σ^2 ,both full conditionals are obtained later in this item, and one Metropolis Hastings (MH) step to sample from λ_i given β, σ^2 and $\mathcal{D}$.

We know that $\beta \sim N(\beta_0, V_0)$ and that $\sigma^2 \sim IG(\nu_0/2, \nu_0\sigma_0^2/2)$. For initial values $\beta_0, V_0, \nu_0/2$ and $\nu_0\sigma_0^2/2$, we can sample from the normal and inverse gamma to obtain $\beta^{(s)}$ and $(\sigma^2)^{(s)}$. Since we have $\beta^{(s)}$ and $(\sigma^2)^{(s)}$, we can use a MH step to obtain $\lambda_i^{(s)}$. We can use $(\sigma^2)^{(s)}$ and $\lambda_i^{(s)}$ and, knowing that the full conditional of β is also normal, we can to obtain $\beta^{(s+1)}$. Then, we can use $\beta^{(s+1)}$, $\lambda_i^{(s)}$ and, since the full of conditional of σ^2 is inverse gamma, sample $(\sigma^2)^{(s+1)}$. Since we have sampled $\beta^{(s+1)}$ and $(\sigma^2)^{(s+1)}$, we can use a MH step to obtain $\lambda_i^{(s+1)}$. By repeating this process, we will obtain a distribution that will approximate

$$p(\beta, \sigma^2, \lambda_1, ..., \lambda_n | \mathcal{D})$$

2.3 Full conditional distribution for β

2.4 Prior

$$\begin{aligned}
p(\beta) &= (2\pi)^{-p/2} det V_0^{-1/2} exp\Big\{\frac{-1}{2}(\beta - \beta_0)^T (V_0^{-1})(\beta - \beta_0)\Big\} \\
&\propto exp\Big\{\frac{-1}{2}(\beta - \beta_0)^T (V_0^{-1})(\beta - \beta_0)\Big\} \\
&\propto exp\Big\{\frac{-1}{2}(\beta^T V_0^{-1}\beta - 2\beta^T V_0^{-1}\beta_0 + \beta_0^T V_0^{-1}\beta_0)\Big\} \\
&\propto exp\Big\{\frac{-1}{2}(\beta^T V_0^{-1}\beta - 2\beta^T V_0^{-1}\beta_0)\Big\}
\end{aligned}$$

2.5 Likelihood

$$\begin{aligned}
L(\beta | y, x, \Lambda, \sigma^2) &= (2\pi)^{-p/2} det(\sigma^2\Lambda)^{-1/2} exp\Big\{\frac{-1}{2}(y - x\beta)^T (\sigma^2\Lambda)^{-1}(y - x\beta)\Big\} \\
&\propto exp\Big\{\frac{-1}{2}(y - x\beta)^T (\sigma^2\Lambda)^{-1}(y - x\beta)\Big\} \\
&\propto exp\Big\{\frac{-1}{2\sigma^2}(y - x\beta)^T \Lambda^{-1}(y - x\beta)\Big\} \\
&\propto exp\Big\{\frac{-1}{2\sigma^2}(y^T\Lambda^{-1}y - 2\beta^T x^T\Lambda^{-1}y + \beta^T x^T \Lambda^{-1}x\beta)\Big\} \\
&\propto exp\Big\{\frac{-1}{2\sigma^2}(-2\beta^T x^T\Lambda^{-1}y + \beta^T x^T \Lambda^{-1}x\beta)\Big\}
\end{aligned}$$

2.6 Posterior

$$\begin{aligned}
P(\beta|y,x,\sigma^2,\Lambda) &\propto p(\beta)\times L(\beta|y,x,\Lambda,\sigma^2)\\
&\propto exp\Big\{\frac{-1}{2}(\beta^T V_0^{-1}\beta - 2\beta^T V_0^{-1}\beta_0)\Big\}\times exp\Big\{\frac{-1}{2\sigma^2}(-2\beta^T x^T\Lambda^{-1}y + \beta^T x^T\Lambda^{-1}x\beta)\Big\}\\
&\propto exp\Big\{\frac{-1}{2}(\beta^T V_0^{-1}\beta - 2\beta^T V_0^{-1}\beta_0) + \frac{-1}{2\sigma^2}(-2\beta^T x^T\Lambda^{-1}y) + \beta^T x^T\Lambda^{-1}x\beta)\Big\}\\
&\propto exp\Big\{\frac{-\beta^T V_0^{-1}\beta}{2} + \beta^T V_0^{-1}\beta_0 + \frac{\beta^T x^T\Lambda^{-1}y}{\sigma^2} - \frac{\beta^T x^T\Lambda^{-1}x\beta}{2\sigma^2}\Big\}\\
&\propto exp\Big\{\beta^T V_0^{-1}\beta_0 + \frac{\beta^T x^T\Lambda^{-1}y}{\sigma^2} - \frac{\beta^T V_0^{-1}\beta}{2} - \frac{\beta^T x^T\Lambda^{-1}x\beta}{2\sigma^2}\Big\}\\
&\propto exp\Big\{\beta^T V_0^{-1}\beta_0 + \frac{\beta^T x^T\Lambda^{-1}y}{\sigma^2} - \frac{1}{2}\Big(\beta^T V_0^{-1}\beta + \frac{\beta^T x^T\Lambda^{-1}x\beta}{\sigma^2}\Big)\Big\}\\
&\propto exp\Big\{\beta^T\Big(V_0^{-1}\beta_0 + \frac{x^T\Lambda^{-1}y}{\sigma^2}\Big) - \frac{1}{2}\Big(\beta^T\Big(V_0^{-1} + \frac{x^T\Lambda^{-1}x}{\sigma^2}\Big)\beta\Big)\Big\}
\end{aligned}$$

Let V_1 and β_1 be such that

$$V_1 = \Big(\frac{x^T\Lambda^{-1}x}{\sigma^2} + V_0^{-1}\Big)^{-1}$$

$$\beta_1 = V_1\Big(\frac{x^T\Lambda^{-1}y}{\sigma^2} + V_0^{-1}\beta_0\Big)$$

.

Then, we can conclude that $\{\beta|y,x,\sigma^2,\Lambda\}\sim N(\beta_1,V_1)$.

2.7 Full conditional distribution for σ^2

2.8 Prior

$$\begin{aligned}
p(\sigma^2) &= \frac{(\nu_0\sigma_0^2/2)^{(\nu_0/2)}}{\Gamma(\nu_0/2)}\frac{1}{(\sigma^2)^{\frac{\nu_0}{2}+1}}exp\Big\{\frac{-\nu_0\sigma_0^2/2}{\sigma^2}\Big\}\\
&\propto \frac{1}{(\sigma^2)^{\frac{\nu_0}{2}+1}}exp\Big\{\frac{-\nu_0\sigma_0^2/2}{\sigma^2}\Big\}
\end{aligned}$$

2.9 Likelihood

$$\begin{aligned}
L(\sigma^2|y,x,\Lambda,\beta) &= (2\pi)^{-p/2}det(\sigma^2\Lambda)^{-1/2}exp\Big\{\frac{-1}{2}(y-x\beta)^T(\sigma^2\Lambda)^{-1}(y-x\beta)\Big\}\\
&\propto det(\sigma^2\Lambda)^{-1/2}exp\Big\{\frac{-1}{2}(y-x\beta)^T(\sigma^2\Lambda)^{-1}(y-x\beta)\Big\}\\
&\propto (\sigma^2)^{-n/2}exp\Big\{\frac{-1}{2\sigma^2}(y-x\beta)^T\Lambda^{-1}(y-x\beta)\Big\}
\end{aligned}$$

2.10 Posterior

$$\begin{aligned}
p(\sigma^2|x,y,\beta,\Lambda) &\propto p(\sigma^2)\times L(\sigma^2|y,x,\Lambda,\beta)\\
&\propto \frac{1}{(\sigma^2)^{(\nu_0/2+1)}}exp\Big\{\frac{-\nu_0\sigma_0^2/2}{\sigma^2}\Big\}\times(\sigma^2)^{-n/2}exp\Big\{\frac{-1}{2\sigma^2}(y-x\beta)^T\Lambda^{-1}(y-x\beta)\Big\}\\
&\propto \frac{1}{(\sigma^2)^{\frac{(\nu_0+n)}{2}+1}}exp\Big\{\frac{-1}{\sigma^2}\Big(\frac{\nu_0\sigma_0^2+(y-x\beta)^T\Lambda^{-1}(y-x\beta)}{2}\Big)\Big\}
\end{aligned}$$

Consider $\nu_1=\nu_0+n$ and $\nu_1\sigma_1^2=\nu_0\sigma_0^2+(y-x\beta)^T\Lambda^{-1}(y-x\beta)$. Then, we can conclude that $\{\sigma^2|y,x,\beta,\Lambda\}\sim IG(\nu_1/2,\nu_1\sigma_1^2/2)$

2.11 Scale mixing variable λ_i

2.12 Prior

$$\begin{aligned}
p(\lambda_i) &= \frac{1}{2}exp\Big\{\frac{-\lambda_i}{2}\Big\}\\
&\propto exp\Big\{\frac{-\lambda_i}{2}\Big\}
\end{aligned}$$

2.13 Likelihood

$$\begin{aligned}
L(\lambda_i|y_i,x_i,\beta,\sigma^2) &= \frac{1}{\sqrt{(2\pi\lambda_i\sigma^2)}}exp\Big\{\frac{-1}{2\lambda_i\sigma^2}(y_i-x_i^T\beta)^2\Big\}\\
&\propto \frac{1}{\lambda_i^{1/2}}exp\Big\{\frac{-1}{2}\Big(\frac{y_i-x_i^T\beta}{\sigma}\Big)^2\lambda_i^{-1}\Big\}
\end{aligned}$$

2.14 Posterior

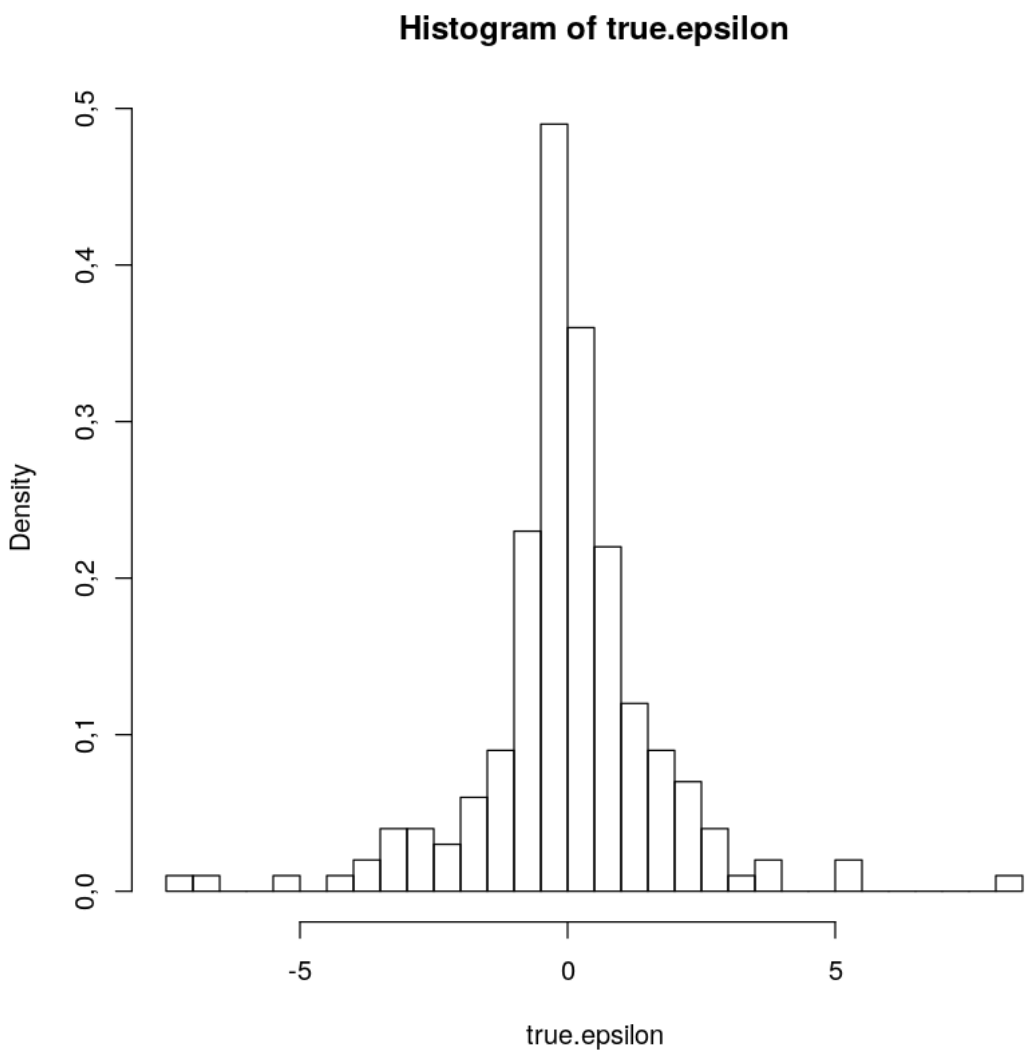
$$\begin{aligned}
p(\lambda_i|y_i,x_i,\beta,\sigma^2) &\propto p(\lambda_i)\times L(\lambda_i|y_i,x_i,\sigma^2,\beta)\\
&\propto exp\Big\{\frac{-\lambda_i}{2}\Big\}\times\frac{1}{\lambda_i^{1/2}}exp\Big\{\frac{-1}{2}\frac{(y_i-x_i^T\beta)^2}{\sigma^2}\lambda_i^{-1}\Big\}\\
&\propto \frac{1}{\lambda^{1/2}}exp\Big\{\frac{-1}{2}\Big(\lambda_i+\Big(\frac{y_i-x_i^T\beta}{\sigma}\Big)^2\lambda_i^{-1}\Big)\Big\}
\end{aligned}$$

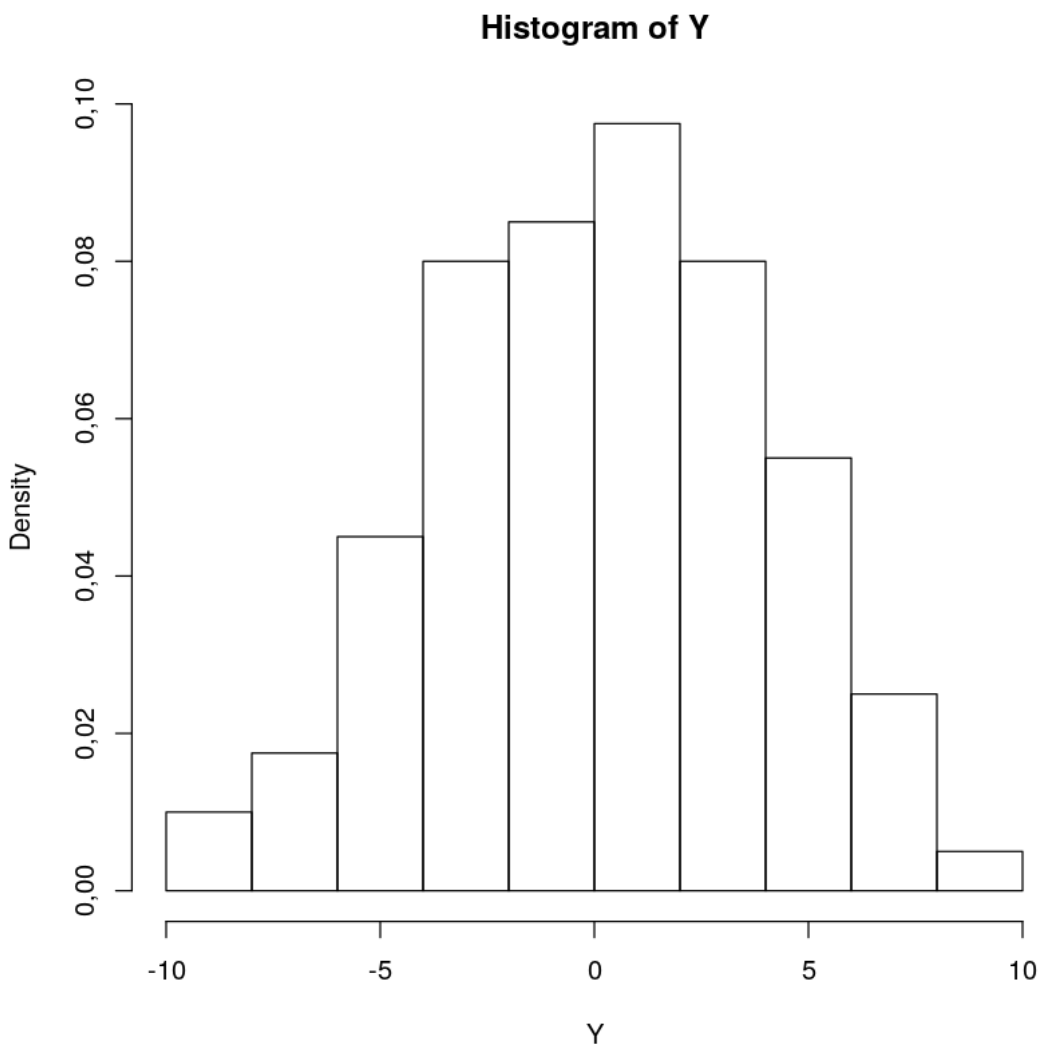
Then, we can conclude that

$$p(\lambda_i|y_i,x_i,\beta,\sigma^2)\propto\lambda^{-1/2}exp\Big\{\frac{-1}{2}\Big(\lambda_i+\Big(\frac{y_i-x_i^T\beta}{\sigma}\Big)^2\lambda_i^{-1}\Big)\Big\}$$

3 Item C

[2]:
```r
# Item c
######### Simulated data ############

n=200
p=4

true.beta=c(0,1,2,3)
true.sigma2=1

#The first collumn is a vector of 1, while the xij \sim N(0,1)
X=matrix(data=NA,nrow=n,ncol=p) #Each row of X is a xi vector
X[,1] =rep(1,n)
set.seed(123)
X[,2]= rnorm(n)
X[,3]= rnorm(n)
X[,4]= rnorm(n)

true.clambda=matrix(data=NA,nrow=n,ncol=1)
true.epsilon=matrix(data=NA,nrow=n,ncol=1)

#Generating epsilons with double exponential distribution
for(i in 1:n) {
  true.clambda[i] = rexp(1,0.5)
  true.epsilon[i] = rnorm(1,0,sqrt(true.sigma2*true.clambda[i]))
}

hist(true.epsilon, prob=TRUE, breaks=30)

Y = X%*%true.beta + true.epsilon

hist(Y, prob=TRUE, breaks=10)
```


Histogram of true.epsilon
Density
0,0
0,1
0,2
0,3
0,4
0,5
-5
0
5
true.epsilon

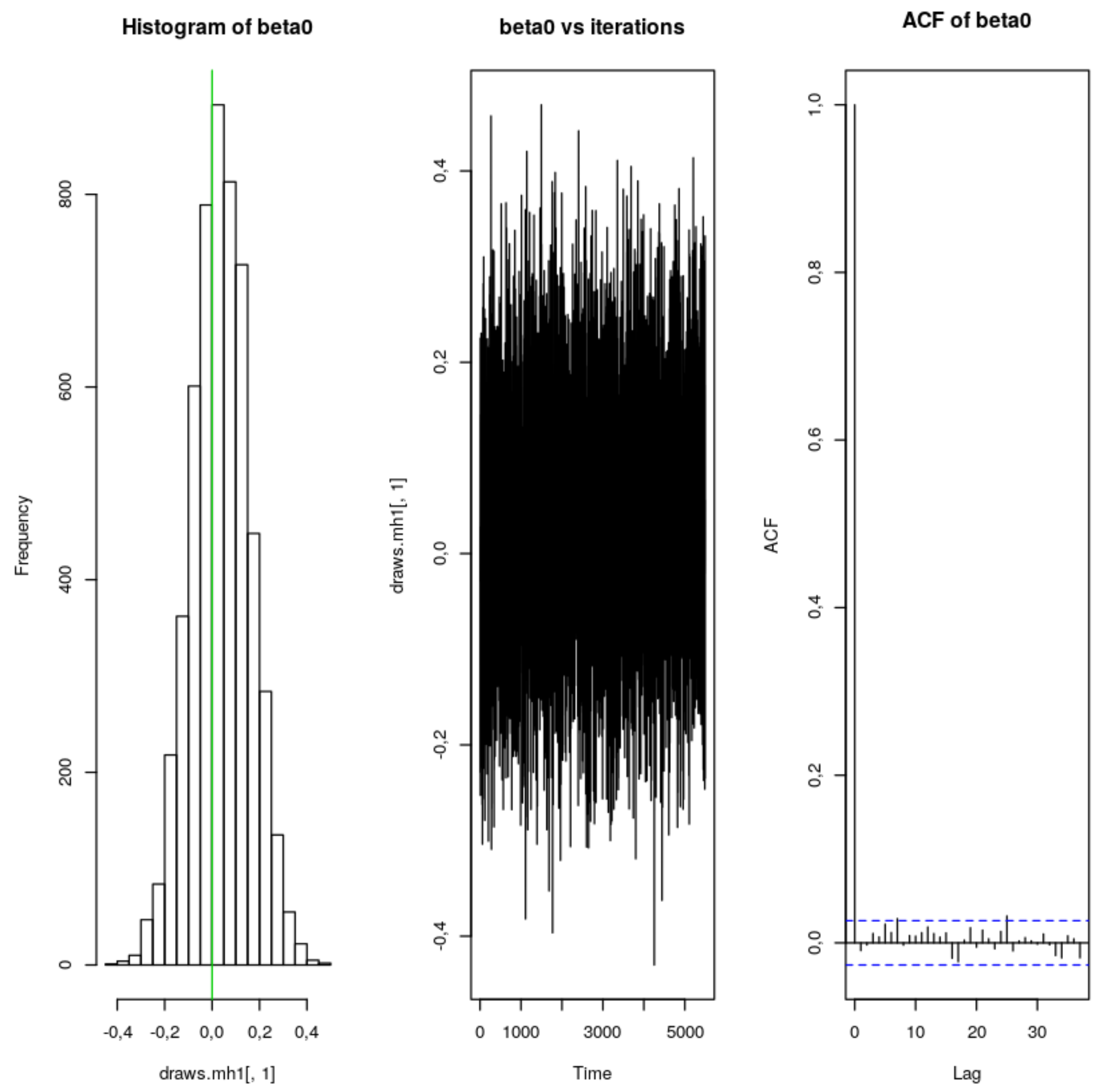


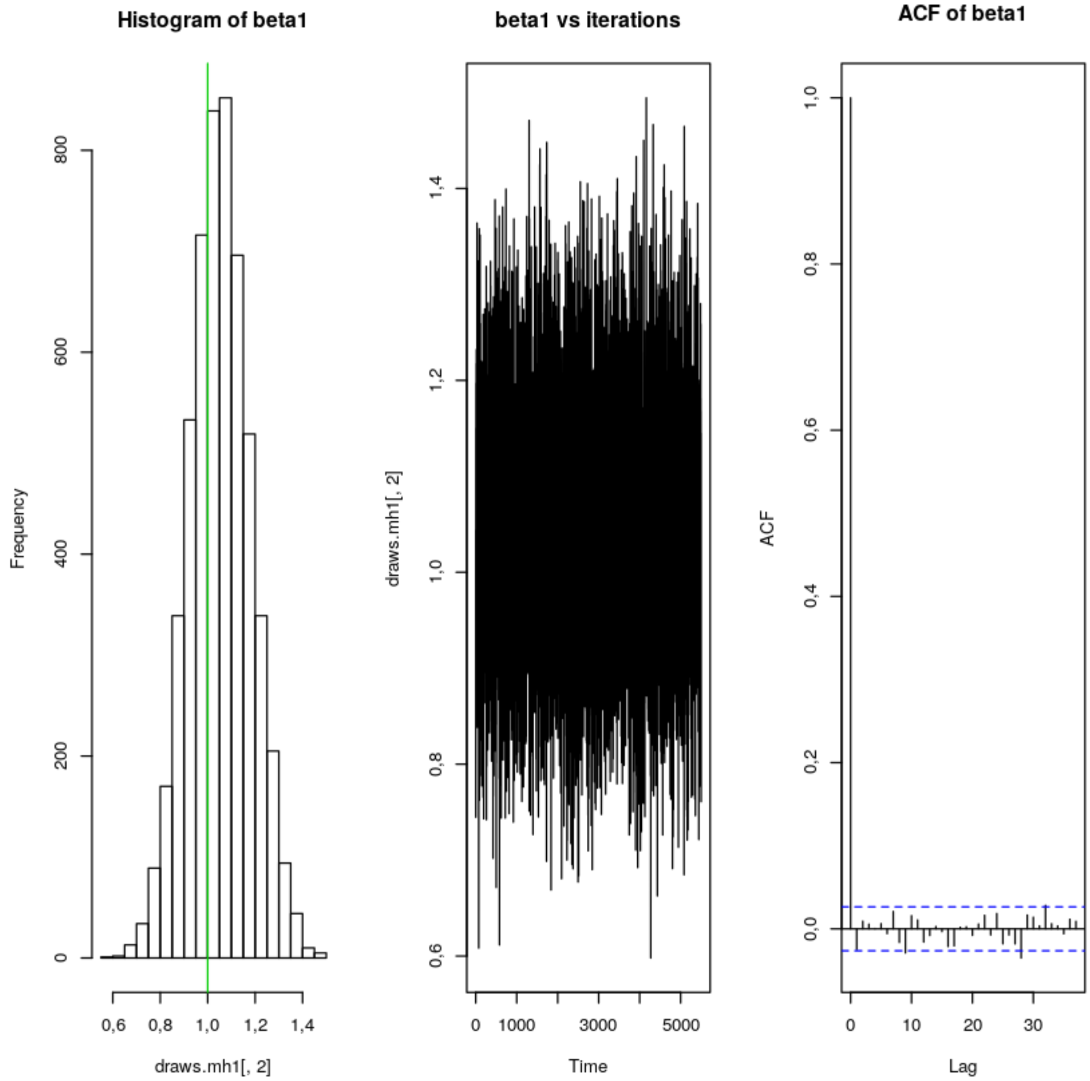
[3]:
```
########## Setting the burn-in and the number of iterations in the MCMC schemes␣
↪#########
M=5000
burnin=0.1*M

niter=burnin+M

######### Part 2: 1rst MCMC scheme - Gibbs + MH ########
logpost.lambdai = function(lambdai,yi,xi,sigma2,beta){
  -0.5*log(lambdai) -0.5*(lambdai + ( ((yi - t(xi)%*%beta)/
↪sqrt(sigma2))^2)*lambdai^(-1) )
}
```

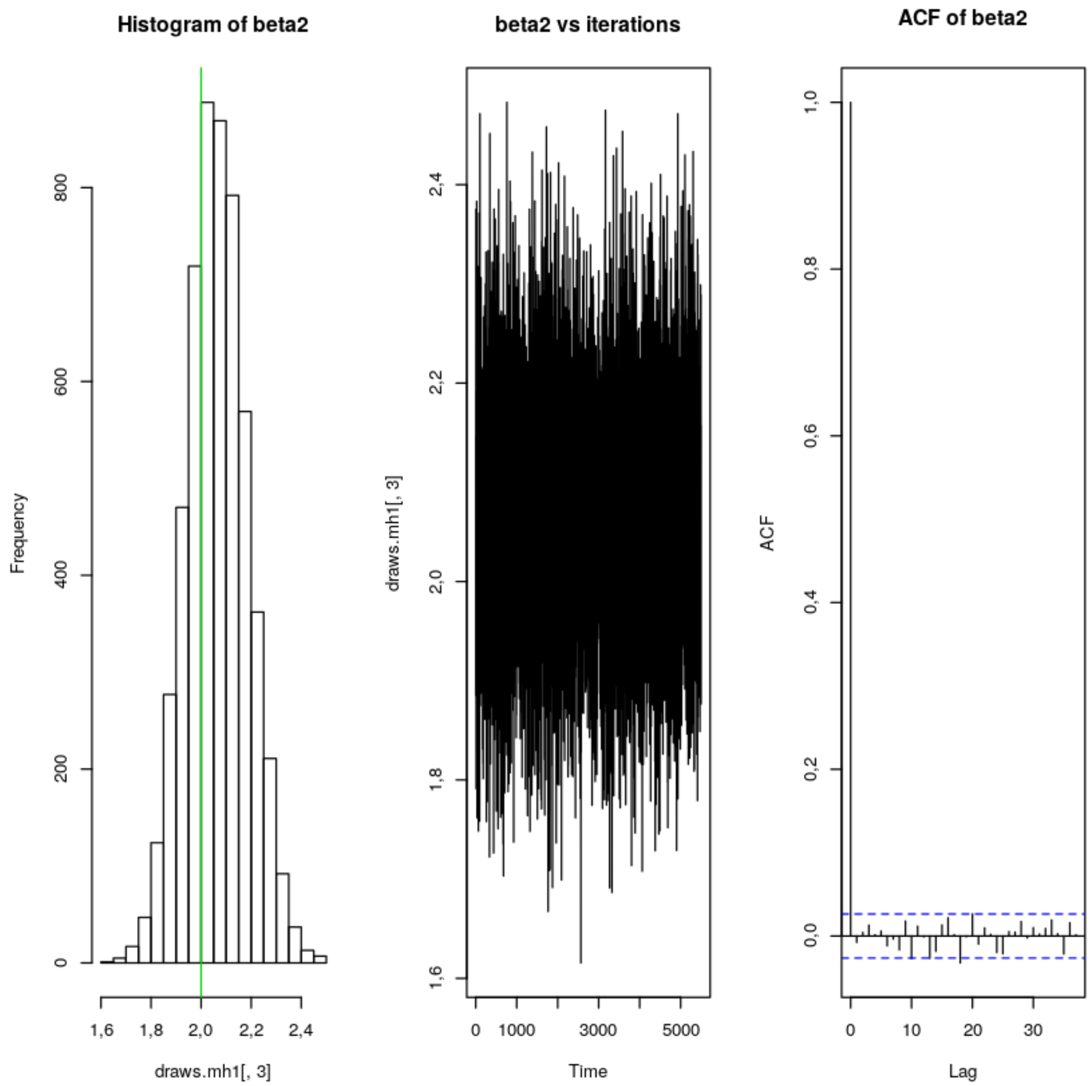
```r
#Setting up initial values
sigma2= 0.9
V0= diag(1,4)
B0=c(1,1,1,1)
invV0=solve(V0)

nu0=3
nu0sigma20=5
nu1<-(nu0+n)/2

c.lambda<-rep(2,n)
sd.clambda=0.1
invlambda=solve(diag(c.lambda))

draws.mh1<-matrix(data=NA,nrow=niter,ncol=5)

#Starting the first MCMC
mh.time=system.time(
  for(i in 1:niter){
    #Gibbs: Beta
    if(i==1){
      sigma2.mh1<-sigma2
    }
    V1<-solve((t(X)%*%invlambda%*%X)/sigma2.mh1 + invV0)
    B1<-V1%*%( (t(X)%*%invlambda%*%Y)/sigma2.mh1 + (invV0%*%B0) )
    beta.mh1<- B1 + t(chol(V1))%*%rnorm(4)

    #Gibbs: Sigma2
    nu1sigma21<-(nu0sigma20+ t(Y-X%*%beta.mh1)%*%invlambda%*%(Y-X%*%beta.mh1))/2
    sigma2.mh1<-1/rgamma(1,nu1,nu1sigma21)

    #MH: lambda
    for(j in 1:n){
      lambda_prop<-rlnorm(1,c.lambda[j],sd.clambda)
      num = logpost.lambdai(lambda_prop,Y[j],X[j,],sigma2.mh1,beta.
↪mh1)-dlnorm(lambda_prop,c.lambda[j],sd.clambda,log=TRUE)
      den= logpost.lambdai(c.lambda[j],Y[j],X[j,],sigma2.mh1,beta.mh1)-dlnorm(c.
↪lambda[j],lambda_prop,sd.clambda,log=TRUE)
      log.alpha = min(0,num-den)
      if (log(runif(1))<log.alpha){
        c.lambda[j] = lambda_prop
      }
    }
    invlambda=solve(diag(c.lambda))

    draws.mh1[i,1:4]<-beta.mh1
    draws.mh1[i,5]<-sigma2.mh1
```

```
  })

mh.time = as.numeric(mh.time[3])

par(mfrow=c(1,3))
hist(draws.mh1[,1],main="Histogram of beta0")
abline(v=true.beta[1],col=3)
ts.plot(draws.mh1[,1], main="beta0 vs iterations")
acf(draws.mh1[,1], main="ACF of beta0")

par(mfrow=c(1,3))
hist(draws.mh1[,2],main="Histogram of beta1")
abline(v=true.beta[2],col=3)
ts.plot(draws.mh1[,2], main="beta1 vs iterations")
acf(draws.mh1[,2], main="ACF of beta1")

par(mfrow=c(1,3))
hist(draws.mh1[,3],main="Histogram of beta2")
abline(v=true.beta[3],col=3)
ts.plot(draws.mh1[,3], main="beta2 vs iterations")
acf(draws.mh1[,3], main="ACF of beta2")

par(mfrow=c(1,3))
hist(draws.mh1[,4],main="Histogram of beta3")
abline(v=true.beta[4],col=3)
ts.plot(draws.mh1[,4], main="beta3 vs iterations")
acf(draws.mh1[,4], main="ACF of beta3")

par(mfrow=c(1,3))
hist(draws.mh1[,5],main="Histogram of sigma2")
abline(v=true.sigma2,col=3)
ts.plot(draws.mh1[,5], main="sigma2 vs iterations")
acf(draws.mh1[,5], main="ACF of sigma2")

#Effective sample size
#Beta 0
beta0.ess.mh1 = round(M/(1+2*sum(acf(draws.mh1[,1],lag.max=10,plot=FALSE)$acf[2:
 ↪11])))
beta0.essps.mh1 = beta0.ess.mh1/mh.time
#Beta 1
beta1.ess.mh1 = round(M/(1+2*sum(acf(draws.mh1[,2],lag.max=10,plot=FALSE)$acf[2:
 ↪11])))
beta1.essps.mh1 = beta1.ess.mh1/mh.time
#Beta 2
beta2.ess.mh1 = round(M/(1+2*sum(acf(draws.mh1[,3],lag.max=10,plot=FALSE)$acf[2:
 ↪11])))
beta2.essps.mh1 = beta2.ess.mh1/mh.time
```

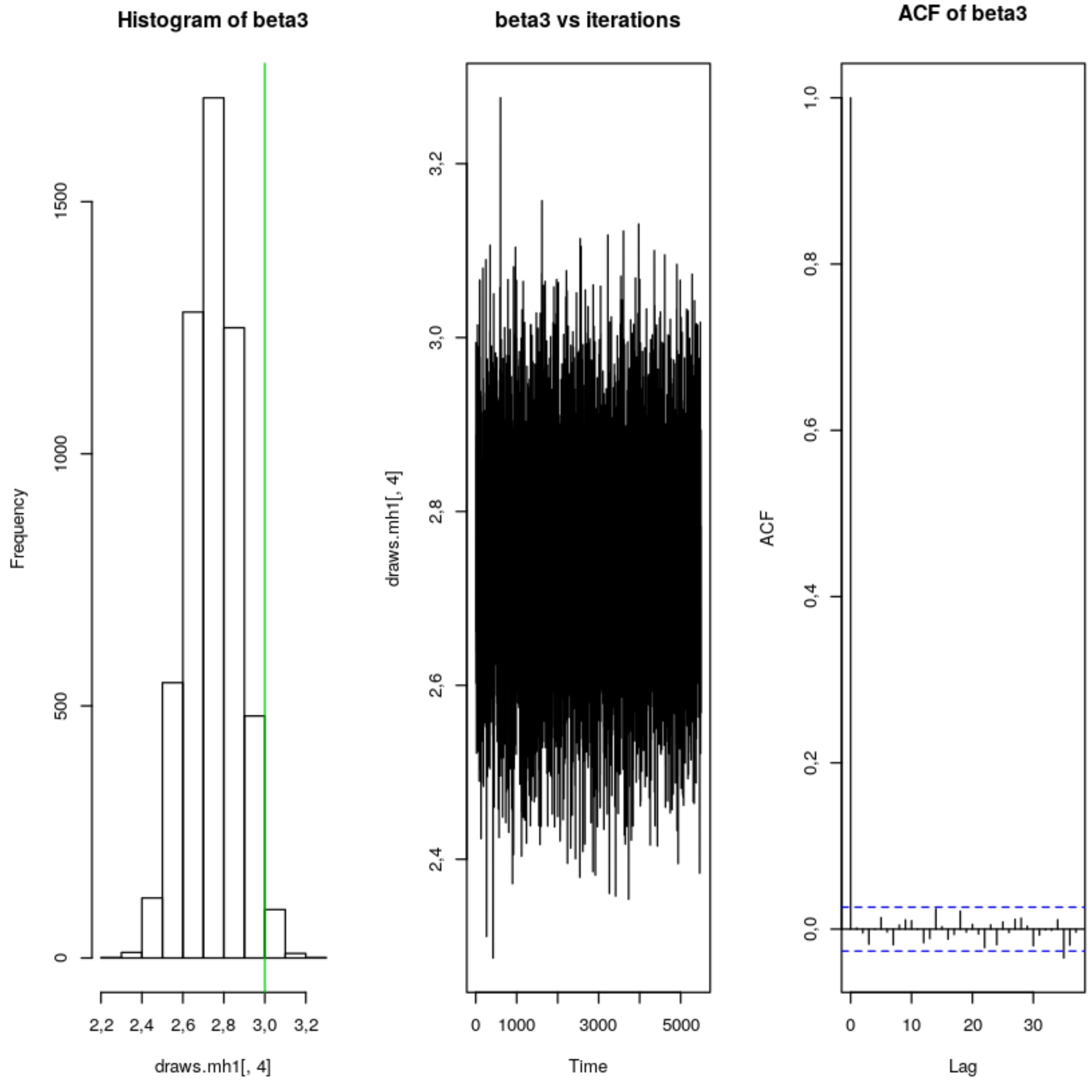
```r
#Beta 3
beta3.ess.mh1 = round(M/(1+2*sum(acf(draws.mh1[,4],lag.max=10,plot=FALSE)$acf[2:
↪11])))
beta3.essps.mh1 = beta3.ess.mh1/mh.time
#Sigma 2
sig2.ess.mh1 = round(M/(1+2*sum(acf(draws.mh1[,5],lag.max=10,plot=FALSE)$acf[2:
↪11])))
sig2.essps.mh1 = sig2.ess.mh1/mh.time
```

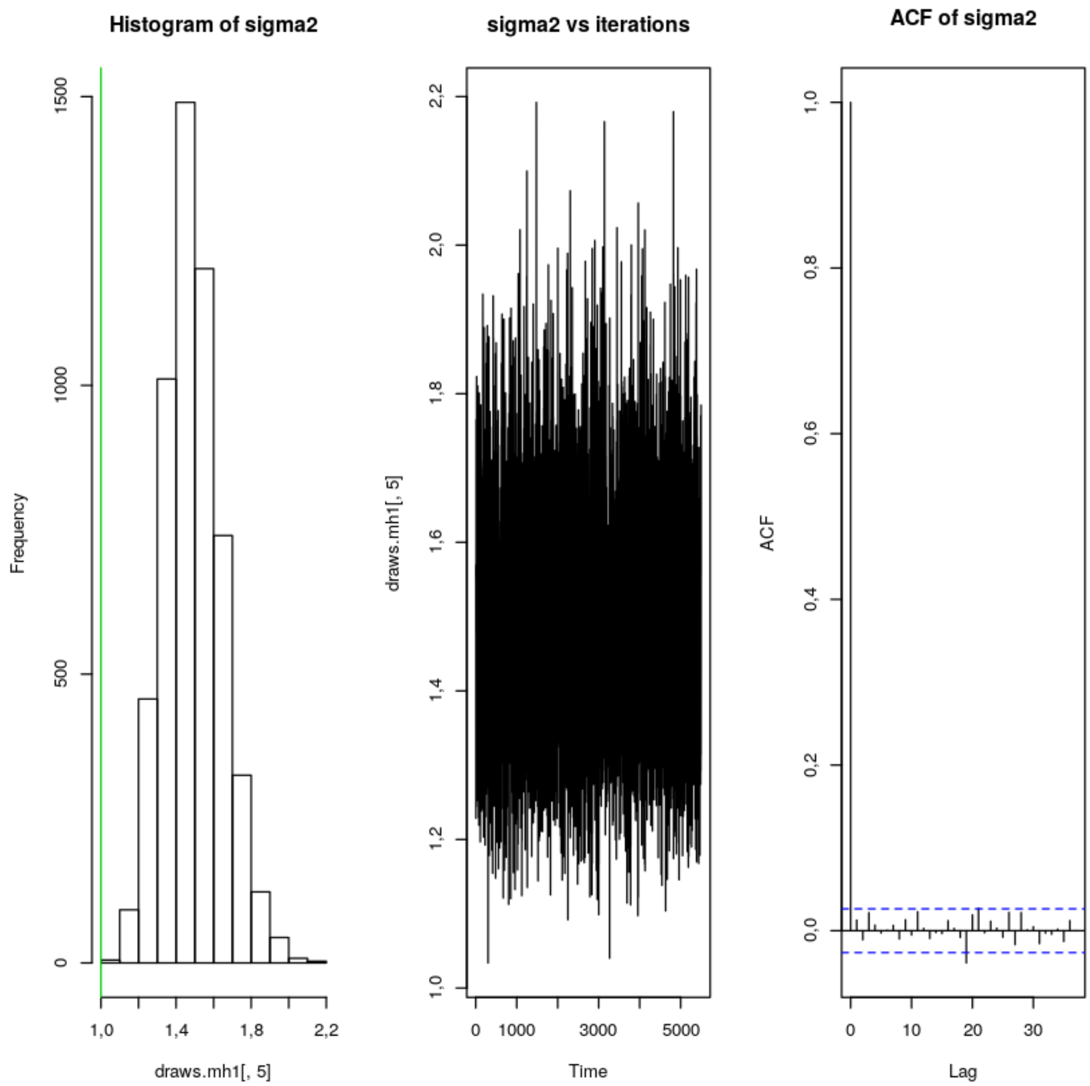



Histogram of beta1
Frequency
draws.mh1[, 2]
beta1 vs iterations
draws.mh1[, 2]
Time
ACF of beta1
ACF
Lag



Histogram of beta2
Frequency
draws.mh1[, 3]
beta2 vs iterations
draws.mh1[, 3]
Time
ACF of beta2
ACF
Lag


Histogram of beta3
Frequency
draws.mh1[, 4]
beta3 vs iterations
draws.mh1[, 4]
Time
ACF of beta3
ACF
Lag

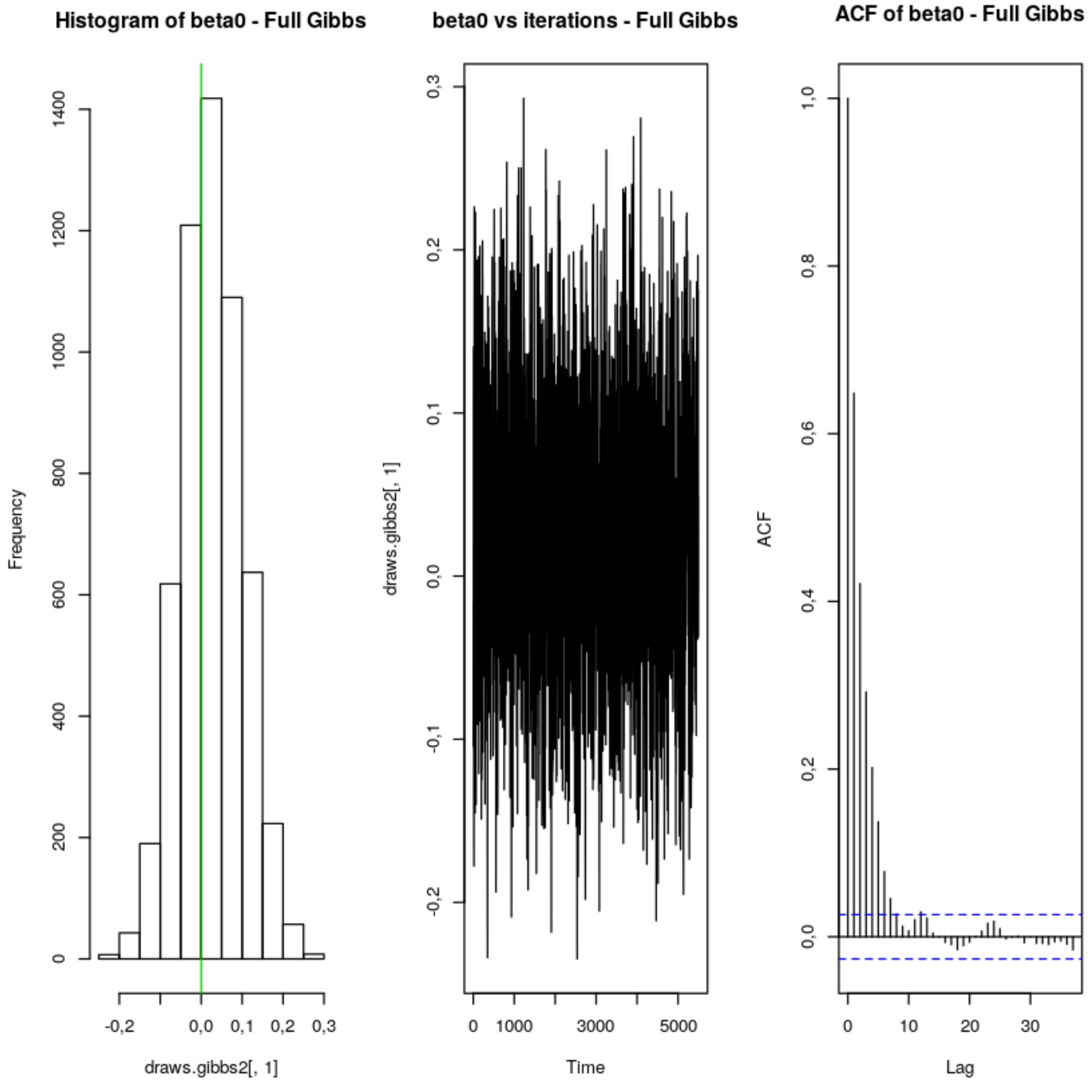


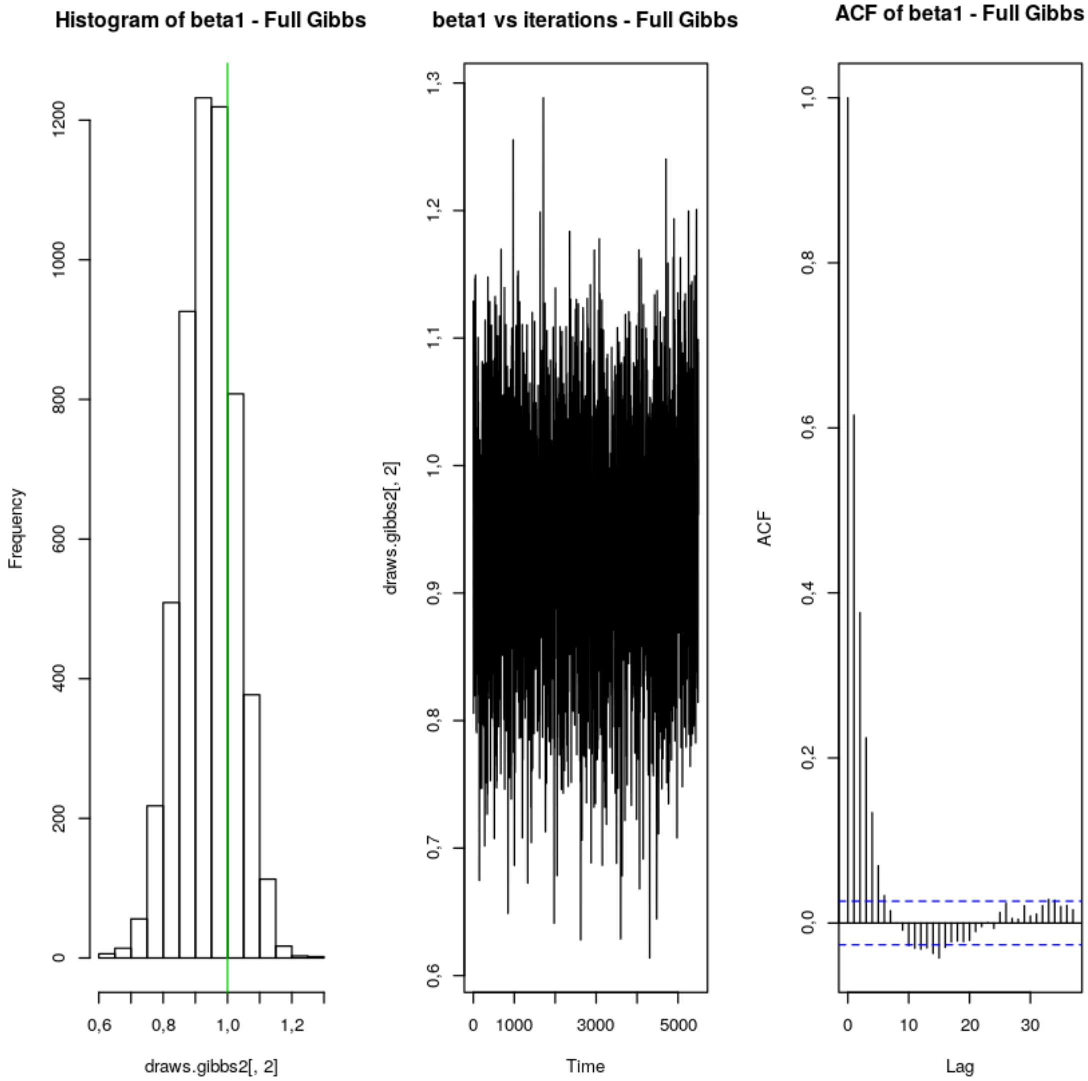
[4]:
```
######### Part 3: 2nd MCMC scheme - Full Gibbs ########

#Setting up the initial values
sigma2= 0.9
V0= diag(1,4)
B0=c(1,1,1,1)
invV0=solve(V0)

nu0=3
nu0sigma20=5

c.lambda<-rep(2,n)
sd.clambda=0.1
```

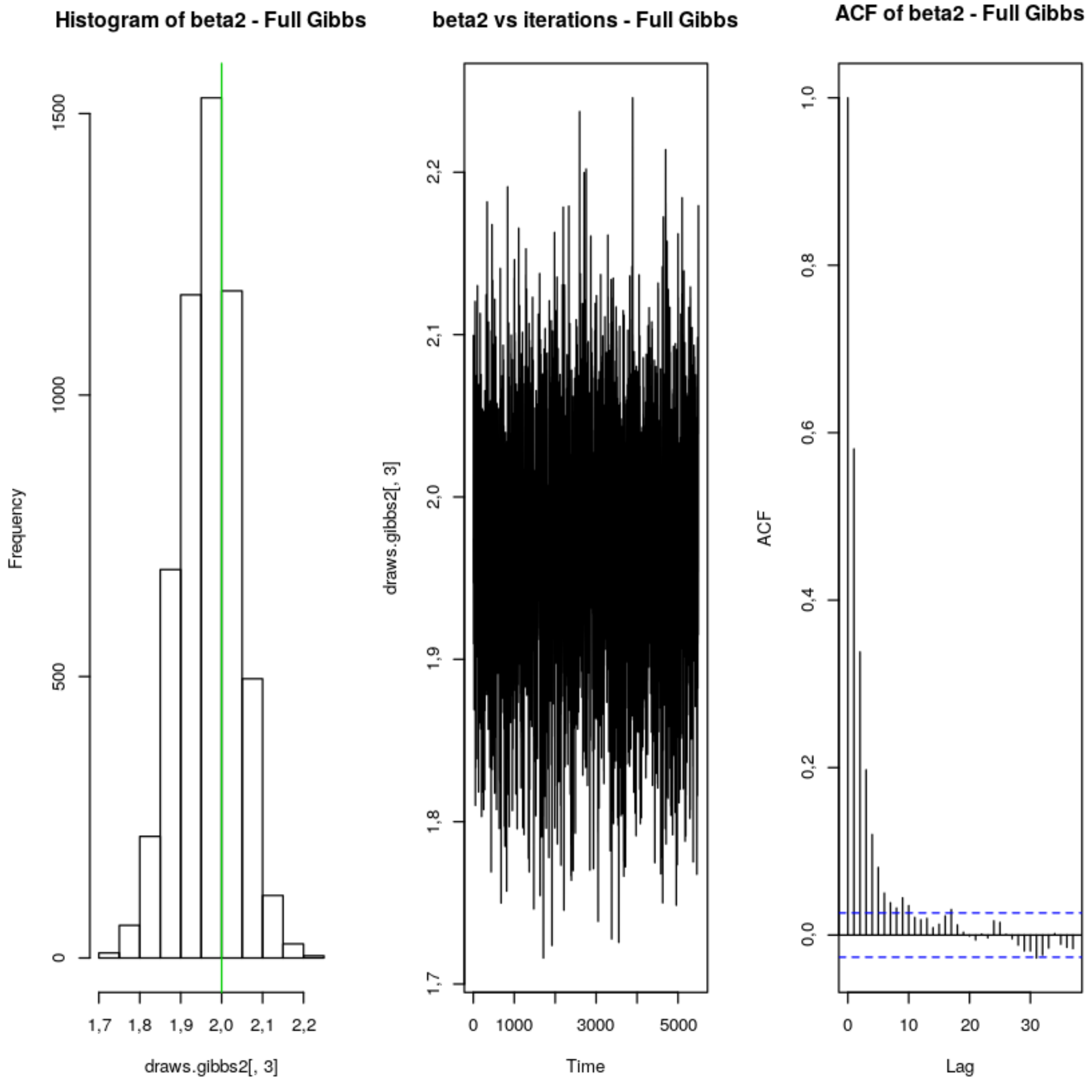
```r
invlambda=solve(diag(c.lambda))

nu1<-(nu0+n)/2

draws.gibbs2<-matrix(data=NA,nrow=niter,ncol=5)

install.packages("GIGrvg")
library(GIGrvg)

gibbs.time = system.time(
  for(i in 1:niter){
    #Gibbs: Beta
    if(i==1){
      sigma2.gibbs2<-sigma2
    }
    V1<-solve((t(X)%*%invlambda%*%X)/sigma2.gibbs2 + invV0)
    B1<-V1%*%( (t(X)%*%invlambda%*%Y)/sigma2.gibbs2 + (invV0%*%B0) )
    beta.gibbs2<- B1 + t(chol(V1))%*%rnorm(4)

    #Gibbs: Sigma2
    nu1sigma21<-(nu0sigma20+ t(Y-X%*%beta.gibbs2)%*%invlambda%*%(Y-X%*%beta.
↪gibbs2))/2
    sigma2.gibbs2<-1/rgamma(1,nu1,nu1sigma21)

    #Gibbs: lambda
    #This is the key different between the two MCMC Schemes
    #install.packages("GIGrvg")
    #library(GIGrvg)

    for(j in 1:n){
      chi= (Y[j] - X[j,]%*%beta.gibbs2)^2/sigma2.gibbs2
      c.lambda[j] =rgig(1, lambda = 0.5, chi = chi, psi= 1)
    }
    invlambda=solve(diag(c.lambda))

    #Saving beta.gibbs2 and sigma2.gibbs2
    draws.gibbs2[i,1:4]<-beta.gibbs2
    draws.gibbs2[i,5]<-sigma2.gibbs2
  })

gibbs.time = as.numeric(gibbs.time[3])

par(mfrow=c(1,3))
hist(draws.gibbs2[,1],main="Histogram of beta0 - Full Gibbs")
abline(v=true.beta[1],col=3)
ts.plot(draws.gibbs2[,1], main="beta0 vs iterations - Full Gibbs")
acf(draws.gibbs2[,1], main="ACF of beta0 - Full Gibbs")
```

```r
par(mfrow=c(1,3))
hist(draws.gibbs2[,2],main="Histogram of beta1 - Full Gibbs")
abline(v=true.beta[2],col=3)
ts.plot(draws.gibbs2[,2], main="beta1 vs iterations - Full Gibbs")
acf(draws.gibbs2[,2], main="ACF of beta1 - Full Gibbs")

par(mfrow=c(1,3))
hist(draws.gibbs2[,3],main="Histogram of beta2 - Full Gibbs")
abline(v=true.beta[3],col=3)
ts.plot(draws.gibbs2[,3], main="beta2 vs iterations - Full Gibbs")
acf(draws.gibbs2[,3], main="ACF of beta2 - Full Gibbs")

par(mfrow=c(1,3))
hist(draws.gibbs2[,4],main="Histogram of beta3 - Full Gibbs")
abline(v=true.beta[4],col=3)
ts.plot(draws.gibbs2[,4], main="beta3 vs iterations - Full Gibbs")
acf(draws.gibbs2[,4], main="ACF of beta3 - Full Gibbs")

par(mfrow=c(1,3))
hist(draws.gibbs2[,5],main="Histogram of sigma2 - Full Gibbs")
abline(v=true.sigma2,col=3)
ts.plot(draws.gibbs2[,5], main="sigma2 vs iterations - Full Gibbs")
acf(draws.gibbs2[,5], main="ACF of sigma2 - Full Gibbs")

#Effective sample size
#Beta 0
beta0.ess.gibbs2 = round(M/(1+2*sum(acf(draws.gibbs2[,1],lag.
↪max=10,plot=FALSE)$acf[2:11])))
beta0.essps.gibbs2 = beta0.ess.gibbs2/gibbs.time
#Beta 1
beta1.ess.gibbs2 = round(M/(1+2*sum(acf(draws.gibbs2[,2],lag.
↪max=10,plot=FALSE)$acf[2:11])))
beta1.essps.gibbs2 = beta1.ess.gibbs2/gibbs.time
#Beta 2
beta2.ess.gibbs2 = round(M/(1+2*sum(acf(draws.gibbs2[,3],lag.
↪max=10,plot=FALSE)$acf[2:11])))
beta2.essps.gibbs2 = beta2.ess.gibbs2/gibbs.time
#Beta 3
beta3.ess.gibbs2 = round(M/(1+2*sum(acf(draws.gibbs2[,4],lag.
↪max=10,plot=FALSE)$acf[2:11])))
beta3.essps.gibbs2 = beta3.ess.gibbs2/gibbs.time
#Sigma 2
sig2.ess.gibbs2 = round(M/(1+2*sum(acf(draws.gibbs2[,5],lag.
↪max=10,plot=FALSE)$acf[2:11])))
sig2.essps.gibbs2 = sig2.ess.gibbs2/gibbs.time
```

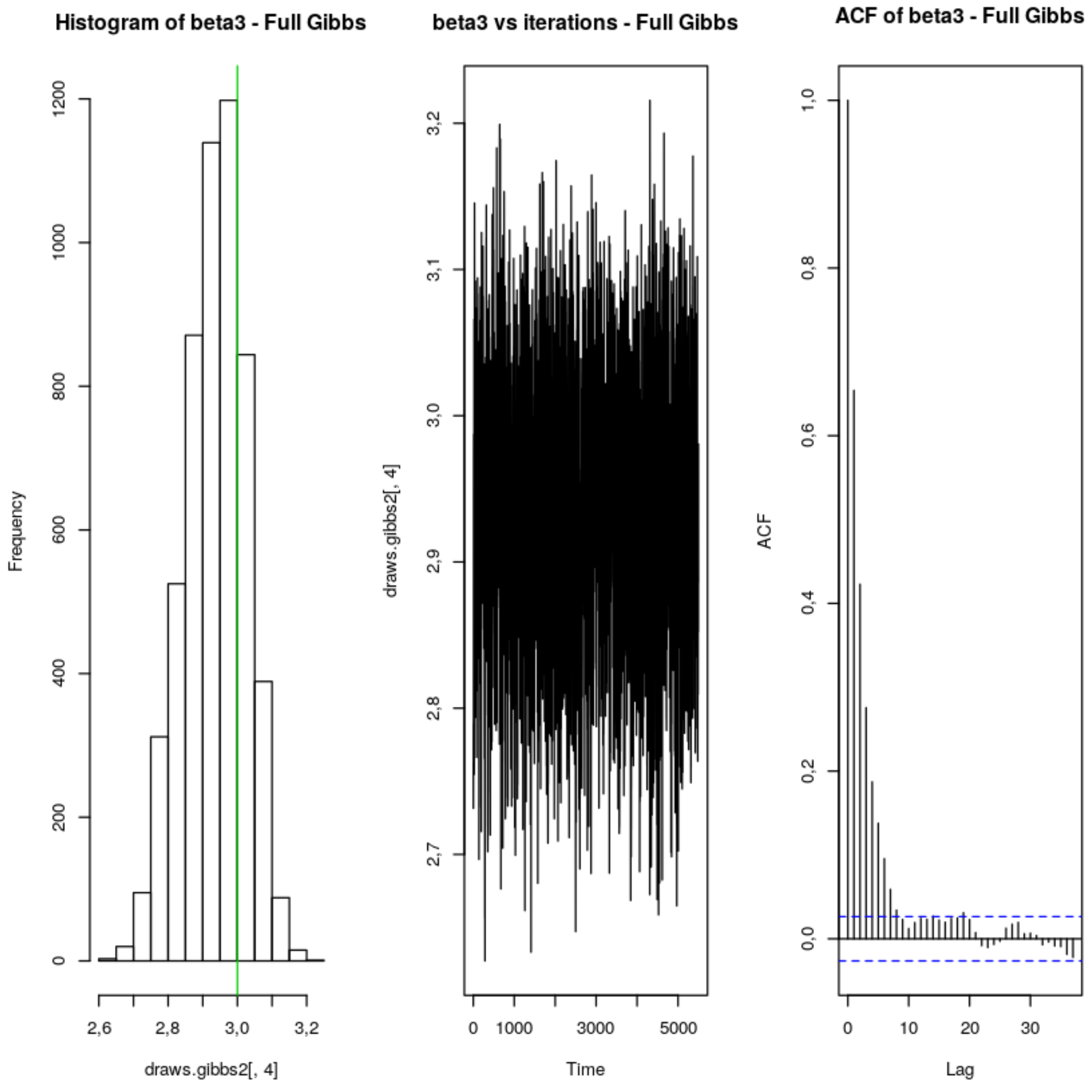
```
Updating HTML index of packages in '.Library'
Making 'packages.html' … done
```



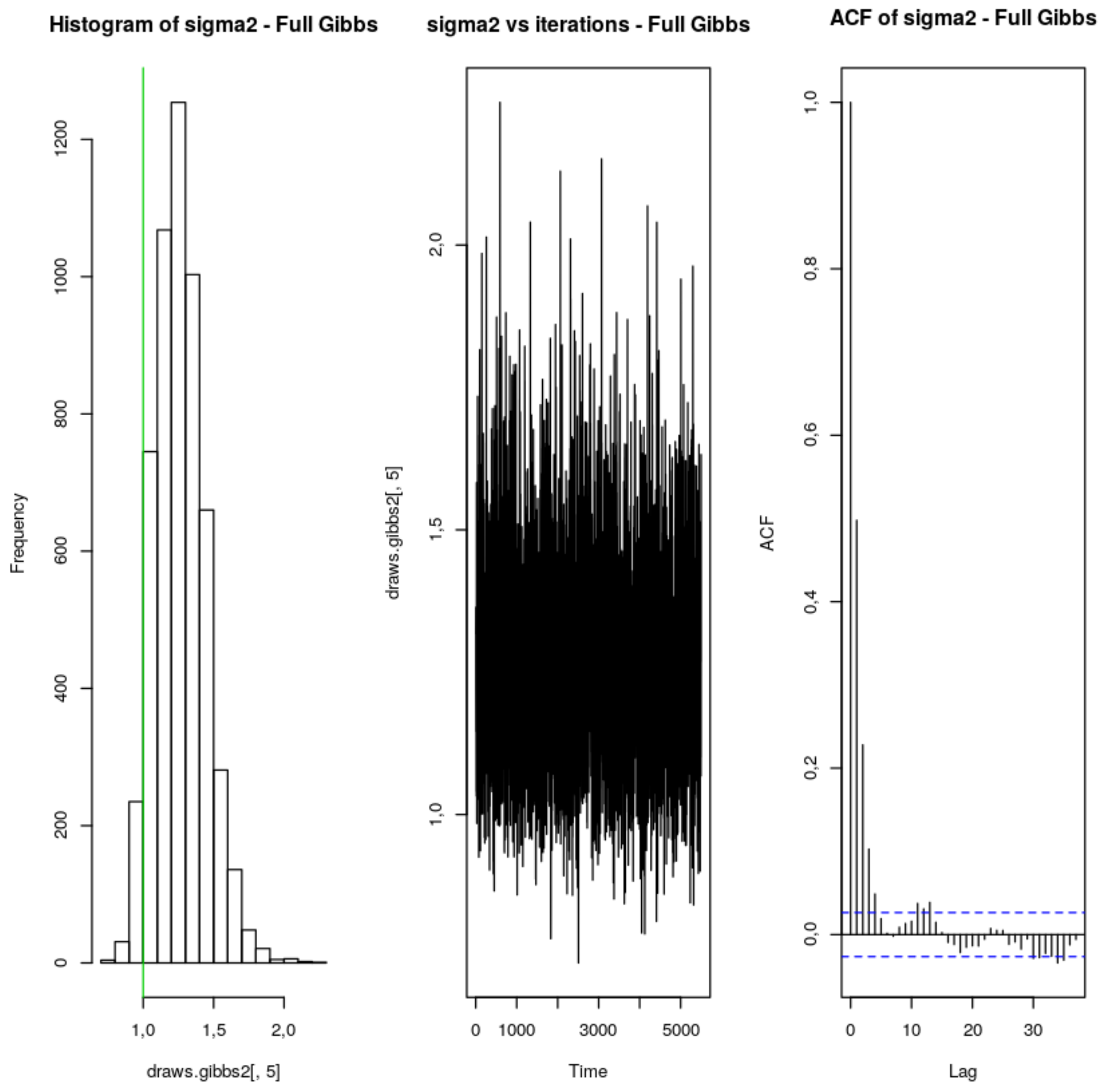

Histogram of beta1 - Full Gibbs
Frequency
draws.gibbs2[, 2]
beta1 vs iterations - Full Gibbs
draws.gibbs2[, 2]
Time
ACF of beta1 - Full Gibbs
ACF
Lag



Histogram of beta2 - Full Gibbs
Frequency
0
500
1000
1500
1,7
1,8
1,9
2,0
2,1
2,2
draws.gibbs2[, 3]
beta2 vs iterations - Full Gibbs
draws.gibbs2[, 3]
1,7
1,8
1,9
2,0
2,1
2,2
0
1000
3000
5000
Time
ACF of beta2 - Full Gibbs
ACF
0,0
0,2
0,4
0,6
0,8
1,0
0
10
20
30
Lag



Histogram of beta3 - Full Gibbs
Frequency
draws.gibbs2[, 4]
beta3 vs iterations - Full Gibbs
draws.gibbs2[, 4]
Time
ACF of beta3 - Full Gibbs
ACF
Lag



[5]:

```
#Comparison between effective sample size.
#Beta
table1=cbind(c(mh.time,beta0.ess.mh1,beta0.essps.mh1),c(gibbs.time,beta0.ess.
 ↪gibbs2,beta0.essps.gibbs2))
rownames(table1) = c("Time (sec)","ESS","ESS/sec")
colnames(table1)=c("MH","Full-Gibbs")
table1

table2=cbind(c(mh.time,beta1.ess.mh1,beta1.essps.mh1),c(gibbs.time,beta1.ess.
 ↪gibbs2,beta1.essps.gibbs2))
rownames(table2) = c("Time (sec)","ESS","ESS/sec")
colnames(table2)=c("MH","Full-Gibbs")
table2
```

```r
table3=cbind(c(mh.time,beta2.ess.mh1,beta2.essps.mh1),c(gibbs.time,beta2.ess.
 ↪gibbs2,beta2.essps.gibbs2))
rownames(table3) = c("Time (sec)","ESS","ESS/sec")
colnames(table3)=c("MH","Full-Gibbs")
table3

table4=cbind(c(mh.time,beta3.ess.mh1,beta3.essps.mh1),c(gibbs.time,beta3.ess.
 ↪gibbs2,beta3.essps.gibbs2))
rownames(table4) = c("Time (sec)","ESS","ESS/sec")
colnames(table4)=c("MH","Full-Gibbs")
table4

#Sigma2
table5=cbind(c(mh.time,sig2.ess.mh1,sig2.essps.mh1),c(gibbs.time,sig2.ess.
 ↪gibbs2,sig2.essps.gibbs2))
rownames(table5) = c("Time (sec)","ESS","ESS/sec")
colnames(table5)=c("MH","Full-Gibbs")
table5
```

	MH	Full-Gibbs
Time (sec)	86,54800	39,14300
ESS	4283,00000	1055,00000
ESS/sec	49,48699	26,95246

	MH	Full-Gibbs
Time (sec)	86,54800	39,14300
ESS	5192,00000	1295,00000
ESS/sec	59,98983	33,08382

	MH	Full-Gibbs
Time (sec)	86,54800	39,14300
ESS	5253,00000	1239,00000
ESS/sec	60,69464	31,65317

	MH	Full-Gibbs
Time (sec)	86,54800	39,14300
ESS	5041,00000	1041,00000
ESS/sec	58,24514	26,59479

	MH	Full-Gibbs
Time (sec)	86,54800	39,14300
ESS	4692,00000	1743,00000
ESS/sec	54,21269	44,52903

The simple MH algorithm is reasonable. It produces adequate estimates for the vector of β and σ^2. However, for β_4 and σ^2 the simple MH does not obtain mean and median values as close to the true values as the full-fledge Gibbs scheme. In terms of running time, the full-fledge Gibbs scheme is faster than the MH version, but has lower effective sample size. I would choose the Full-Gibbs since it leads to posterior densities with mean and median values closer to the true values.