1155017920 Bayesian coding exam 2019

#the function of R\_hat to judge the convergence  
Rhat <- function(data){  
 n <- dim(data)[1]  
 r <- dim(data)[2]  
 pj <- apply(data,2,mean)  
 sj <- apply(data,2,var)  
 pjk <- mean(pj)  
 B <- sum((pj-pjk)^2)\*n/(r-1)  
 W <- mean(sj)  
 R <- ((1-1/n)\*W+B/n)/W  
 return(R)  
}  
  
r.TruncatedNorm <- function(mu, sigma,lower,upper){  
 temp <- rnorm(1,mu,sigma)  
 while((temp>=upper)|(temp<=lower)){temp <- rnorm(1,mu,sigma)}  
 return(temp)  
}

#### 1

set.seed(0)  
  
g <- function(x, y) abs(cos(x)+sin(y)) \* (1/sqrt(2\*base::pi)) \* exp(-(y-1)^2) \* exp(-x)  
N <- 100000 #the sample size of each estimate  
  
estimate <- replicate(10,  
 {  
 X <- rexp(N)  
 Y <- rnorm(N)  
   
 mean(g(X, Y) / dexp(X) / dnorm(Y))  
   
 })  
  
mean(estimate)

## [1] 0.8610447

sd(estimate)

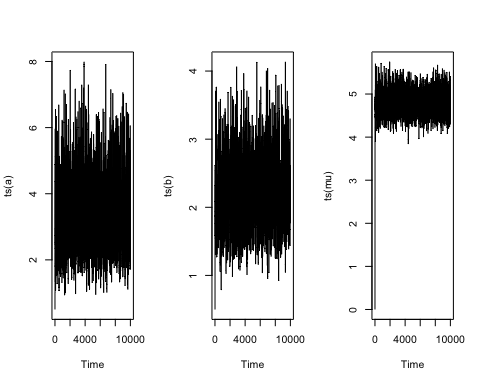
## [1] 0.003743114

#### 2

##### a

x=c(0.99, 0.22, 0.23, 0.21, 0.87, 0.12, 0.03, 0.33, 0.10, 0.26)  
y=c(1.64, 0.96, 1.05, 1.02, 0.73, 1.60, 1.70, 0.34, 0.60, 0.11)  
z=c(3.81, 4.86, 7.26, 4.24, 4.58, 4.62, 4.22, 5.87, 5.51, 4.55)  
  
d.target <- function(a, b, mu)  
{  
 prod(a \* b^2 \* y \* sqrt(x\*y + 1) \* exp(-.5 \* (x\*y+1) \* (z-mu)^2 - (b\*y) - (a\*x)))   
}

set.seed(0)  
  
N <- 10000 #sample size  
M <- 3 #number of chains  
A <- c()  
B <- c()  
Mu <- c()  
  
for(m in 1:M)  
{  
 a <- rep(NA,N);a[1] <- 0.5  
 b <- rep(NA,N);b[1] <- 0.5  
 mu <- rep(NA,N);mu[1] <- 0  
 sigma <- 1 #the sigma of the proposal norm distribution  
   
 for(i in 2:N)  
 {  
 a.try <- rlnorm(1,log(a[i-1])-sigma^2/2,sigma)  
 r.a <- d.target(a.try, b[i-1], mu[i-1]) / d.target(a[i-1], b[i-1], mu[i-1]) \* dlnorm(a[i-1],log(a.try)-sigma^2/2,sigma)/dlnorm(a.try,log(a[i-1])-sigma^2/2,sigma)  
 a[i] <- ifelse(runif(1) < r.a, a.try, a[i-1])  
   
   
 b.try <- rlnorm(1,log(b[i-1])-sigma^2/2,sigma)  
 r.b <- d.target(a[i], b.try, mu[i-1]) / d.target(a[i], b[i-1], mu[i-1]) \* dlnorm(b[i-1],log(b.try)-sigma^2/2,sigma)/dlnorm(b.try,log(b[i-1])-sigma^2/2,sigma)  
 b[i] <- ifelse(runif(1) < r.b, b.try, b[i-1])  
   
   
 mu.try <- rnorm(1,mu[i-1],sigma)  
 r.mu <- d.target(a[i], b[i], mu.try) / d.target(a[i], b[i], mu[i-1])  
 mu[i] <- ifelse(runif(1) < r.mu, mu.try, mu[i-1])  
 }  
   
 A <- cbind(A,a)  
 B <- cbind(B, b)  
 Mu <- cbind(Mu,mu)  
}  
  
par(mfrow=c(1,3))  
plot(ts(a))  
plot(ts(b))  
plot(ts(mu))



Rhat(A[-(1:(N/2)),])

## [1] 1.000224

Rhat(B[-(1:(N/2)),])

## [1] 1.000004

Rhat(Mu[-(1:(N/2)),])

## [1] 1.000475

Prior distributions for parameters are: p(a) 1, p(b) 1, p(mu) 1

#### c

mean(a[-(1:(N/2))])

## [1] 3.255274

var(a[-(1:(N/2))])

## [1] 0.9822938

mean(b[-(1:(N/2))])

## [1] 2.14964

var(b[-(1:(N/2))])

## [1] 0.2211324

mean(mu[-(1:(N/2))])

## [1] 4.823625

var(mu[-(1:(N/2))])

## [1] 0.06573634

##### d

sum((a>b)[-(1:(N/2))])/(N/2)

## [1] 0.8448

##### e

mean(a[(b<mu)[-(1:(N/2))]])

## [1] 3.241911