hw5\_192STG11\_우나영

## Library setup

library(rjags); library(runjags)

## Loading required package: coda

## Warning: package 'coda' was built under R version 3.6.1

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

## 예 5.1

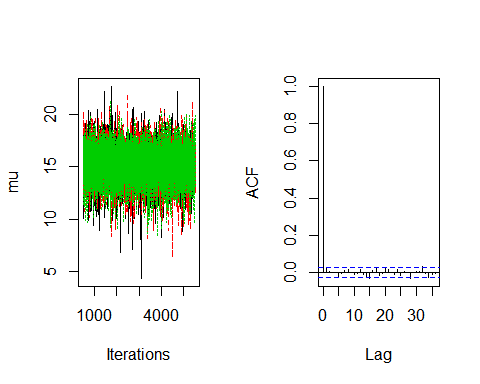
~

~ ~

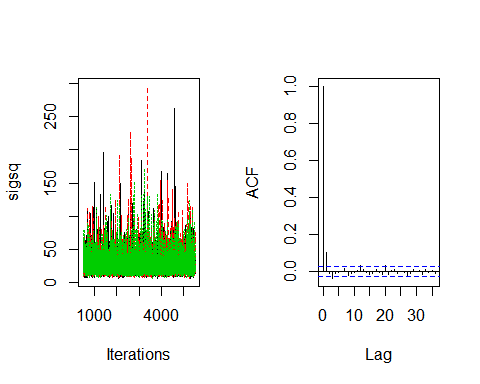
# 2. JAGS의 모형 결정(Model Specification)  
modelString = "  
model  
{  
 for(i in 1:n){  
 x[i] ~ dnorm(mu, invsigsq) # 데이터의 분포  
 }  
 mu ~ dnorm(mu0, invsigsq0) # 사전분포  
 invsigsq ~ dgamma(a, b)  
 sigsq <- 1/invsigsq # 모수의 변환  
 mu0 <- 10 # 상수값 지정  
 invsigsq0 <- 1/25  
 a <- 0.5  
 b <- 1  
}  
"  
  
writeLines(modelString, "model\_ex5\_1.txt")  
  
# 3. 리스트 형식으로 입력할 데이터를 저장  
dataList = list(n = 10, x = c(10, 13, 15, 11, 9, 18, 20, 17, 23, 21))  
  
# 4. 리스트 형식으로 랜덤 노드의 초기치 결정  
initsList = list(mu = 10, invsigsq = 1/25)  
  
# 5. rjags 패키지에 포함된 jags.model 함수를 사용하여 mcmc를 위한 준비작업을 수행한다.  
jagsModel = jags.model(file = "model\_ex5\_1.txt", data = dataList, inits = initsList, n.chains = 3, n.adapt = 500)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 10  
## Unobserved stochastic nodes: 2  
## Total graph size: 19  
##   
## Initializing model

# 6. 예비단계 생성  
update(jagsModel, n.iter = 500)  
  
# 7. MCMC 표본추출  
codaSamples = coda.samples(jagsModel, variable.names = c("mu", "sigsq"), n.iter = 5000)  
  
# 8. 수렴진단  
#### Fig 5.4 & 5.5 표본의 경로그림과 자기상관계수 #####  
par(mfrow = c(1,2))  
coda::traceplot(codaSamples[,"mu"], main = "", ylab = "mu")  
acf(codaSamples[,"mu"][[1]], plot = T, main = "")



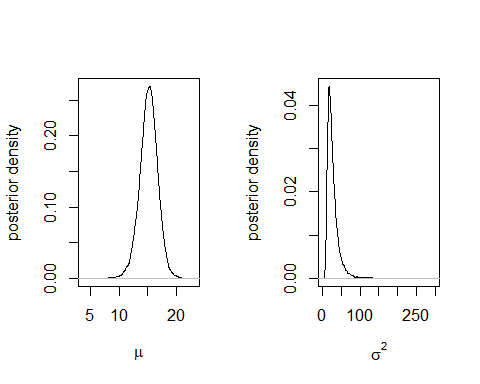
coda::traceplot(codaSamples[,"sigsq"], main = "", ylab = "sigsq")  
acf(codaSamples[,"sigsq"][[1]], plot = T, main = "")



gelman = gelman.diag(codaSamples)  
gelman.1 = as.matrix(gelman$psrf)  
if(max(gelman.1)>1.1) cat("Warning : Gelman Shrink Factor > 1.1", "\n")  
gelman.2 = gelman$mpsrf  
if(gelman.2>1.1) cat("Warning: Gelman Multivariate Shrink Factor > 1.1", "\n")  
  
#--- Check MCMC efficiency ---#  
mcmcSamples.combined = mcmc(codaSamples[[1]])  
mcmcSamples.combined = rbind(mcmcSamples.combined, mcmc(codaSamples[[2]]))  
mcmcSamples.combined = rbind(mcmcSamples.combined, mcmc(codaSamples[[3]]))  
  
ESS = effectiveSize(mcmcSamples.combined)  
cat("Effective Sample size = ", ESS)

## Effective Sample size = 14029.38 12470.43

# 9. 사후추론  
MuSamples = as.matrix(codaSamples[, "mu"])  
SigSamples = as.matrix(codaSamples[, "sigsq"])  
  
#### Fig 5.6 주변 사후밀도함수 ####  
par(mfrow=c(1,2))  
plot(density(MuSamples), main = "", xlab = bquote(mu), ylab = "posterior density")  
plot(density(SigSamples), main = "", xlab = bquote(sigma^2), ylab = "posterior density")



AcceptRate = 1 - rejectionRate(codaSamples); AcceptRate

## mu sigsq   
## 1 1

# => Gibbs

## 예 5.2 : 회귀모형의 베이지안 분석

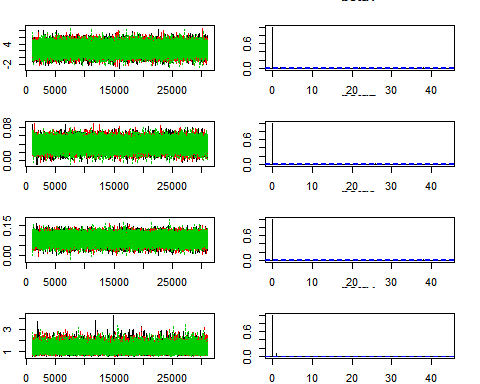
~

~

dat = read.csv("immigrants.csv")  
y = dat$wage  
n = length(y)  
X = cbind(rep(1, n), dat$sp, dat$lit)  
p = ncol(X)  
  
a = 1; b = 1  
XtX = t(X)%\*%X  
XtX.inv = solve(XtX)  
Xty = t(X) %\*% y  
beta.hat = as.vector(XtX.inv %\*% Xty)  
sigsq.hat = sum((y-X%\*%beta.hat)^2)/(n-p)  
beta0 = beta.hat  
Sig0 = diag(diag(XtX.inv))\*sigsq.hat\*100  
Sig0.inv = solve(Sig0)  
  
modelString = "  
model  
{  
 for(i in 1:length(y)){  
 y[i] ~ dnorm(inprod(X[i,], beta[]), invsigsq)  
 }  
  
  
beta[1:length(beta0)] ~ dmnorm(beta0[], Sig0.inv[,])  
  
invsigsq ~ dgamma(a,b)  
sigsq = 1/invsigsq  
  
}  
"  
  
writeLines(modelString, "model\_reg.txt")  
  
dataList = list(X=X, y = y, a = a, b=b, beta0=beta0, Sig0.inv = Sig0.inv)  
initsList = list(beta=beta.hat, invsigsq=1/sigsq.hat)  
nChains = 3  
jagsModel = jags.model(file = "model\_reg.txt", data = dataList, inits = initsList, n.chains = nChains, n.adapt = 500)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 35  
## Unobserved stochastic nodes: 2  
## Total graph size: 230  
##   
## Initializing model

update(jagsModel, n.iter = 1000)  
codaSamples= coda.samples(jagsModel, variable.names = c("beta", "sigsq"), n.iter = 30000)  
para.names <- c("beta1", "beta2", "beta3", "beta4")  
  
#### Fig 5.7: 경로그림과 자기상관 ####  
par(mfrow=c(4,2))  
par(mar = rep(2, 4))  
for(i in 1:4){  
 coda::traceplot(codaSamples[,i], main = "", ylab = para.names[i])  
 acf(codaSamples[,i][[1]], plot = T, main = para.names[i])  
}



MCMCSamples = as.matrix(codaSamples)  
(HPD=round(apply(MCMCSamples, 2, quantile, probs = c(0.025, 0.975)), 4))

## beta[1] beta[2] beta[3] sigsq  
## 2.5% -0.0240 0.0173 0.0404 0.7055  
## 97.5% 5.1335 0.0594 0.1186 1.8539

#### Fig 5.8: 사후밀도함수와 95% 사후구간 ####  
par(mfrow=c(2,2))  
for(i in 1:4){  
 plot(density(MCMCSamples[,i]), main = "", xlab = para.names[i], col = "blue")  
 abline(v=HPD[,i], col = 2)  
}

