

Lecture 13: M-H and Gibbs R scripts

1 Example one: Binomial likelihood with Beta prior

If $\theta \sim \text{Beta}(\alpha, \beta)$ and $(y|\theta) \sim \text{Binomial}(n, \theta)$, then the joint pdf of (y, θ) is

$$\begin{aligned} p(y, \theta) &= \binom{n}{y} \theta^y (1 - \theta)^{n-y} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \\ &= \binom{n}{y} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha+y-1} (1 - \theta)^{\beta+n-y-1} \end{aligned}$$

from which we can deduce that the posterior of θ is $\text{Be}(\alpha + y, \beta + n - y)$. Even though we know the true posterior distribution for this example, we will apply the Metropolis-Hastings algorithms to simulate the posterior pdf of θ , where we use $q(\theta|y) = p(y, \theta)$ as the kernel of the posterior pdf $p(\theta|y)$.

1.1 Demonstration of Metropolis Hastings 1.

Choose $\text{Beta}(4, 16)$ as the proposed transition pdf $Q(\theta|\theta^{(t-1)})$.

The acceptance rate $\alpha(\theta^{(t-1)}, \theta^{(t)})$ is to be computed based on the log-pdf. Namely, we first calculate $\log \left(\frac{q(\theta^{(t)}|y)Q(\theta^{(t-1)}|\theta^{(t)})}{q(\theta^{(t-1)}|y)Q(\theta^{(t)}|\theta^{(t-1)})} \right)$, then we obtain the value of $\alpha(\theta^{(t-1)}, \theta^{(t)})$. We will also use the Bernoulli distribution to implement the accepting/rejecting procedure.

```
y<-rbinom(1,size=20,prob=0.2); y #Generate the y data.
```

```
## [1] 4
```

```
#Construct simulations.
a<-4; b<-16 #hyper-parameters of the proposed transition pdf.
ap<-1; bp<-1 #hyper-parameters of the prior distribution.
thta0<-0.95 #Starting point.
indi<-0 #initial value of the number of distinct generated theta's
iter =20000 #number of iterations.
thta.v<-thta0
thta.v <-rep(thta0,iter+1) #posterior samples of theta
for(i in 1:iter){
thta.t<-rbeta(1,a,b)
#working with log densities.
r_up<-dbinom(y,20,thta.t,log=TRUE) + dbeta(thta.t,ap,bp,log=TRUE) +
  dbeta(thta.v[i],a,b,log=TRUE)
r_low<-dbinom(y,20,thta.v[i],log=TRUE) + dbeta(thta.v[i],ap,bp,log=TRUE) +
  dbeta(thta.t,a,b,log=TRUE)
r<-r_up-r_low
```

```

#Draw an indicator for whether to accept/reject candidate
ind<-rbinom(1,1,exp( min(c(r,0)) ) )
thta.v[i+1]<- ind*thta.t + (1-ind)*thta.v[i]
indi<-indi+ind
}
#Example of plot of iterations.
length(thta.v)

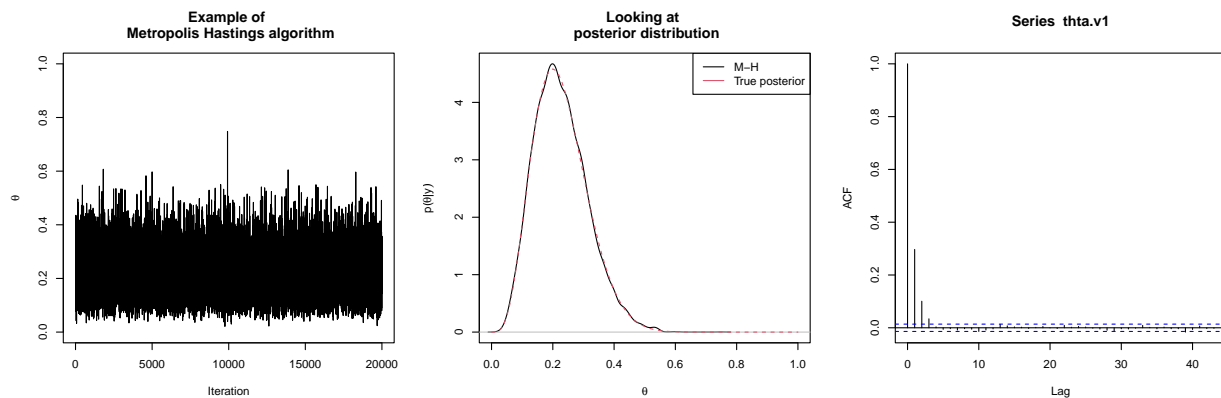
```

```
## [1] 20001
```

```

thta.v1 <- thta.v[2:(iter+1)]
par(mfrow=c(1,3))
plot(thta.v1,xlab='Iteration',ylab=expression(theta),ylim=c(0,1),
     main='Example of \n Metropolis Hastings algorithm',type='l')
#Compare to posterior.
plot(density(thta.v1), xlab=expression(theta), ylab=expression(paste(p,"(",theta,group("|",y,")")),
     xlim=c(0,1), main='Looking at \n posterior distribution')
curve(dbeta(x,y+ap,20-y+bp), add=TRUE, col=2, lty=2)
legend('topright',legend=c('M-H', 'True posterior'),lwd=1,col=1:2)
acf(thta.v1)

```



```
indi
```

```
## [1] 16393
```

```
head(thta.v1)
```

```
## [1] 0.2031706 0.3947234 0.4005966 0.2070418 0.1792938 0.1792938
```

1.2 Demonstration of Metropolis Hastings 2.

Choose truncated Normal($\hat{\theta}_{ML}, n^{-1}\hat{\theta}_{ML}(1 - \hat{\theta}_{ML})$) as the proposed transition pdf $Q(\theta|\theta^{(t-1)})$.

```
y<-rbinom(1,size=20,prob=0.2); y
```

```
## [1] 4
```

```

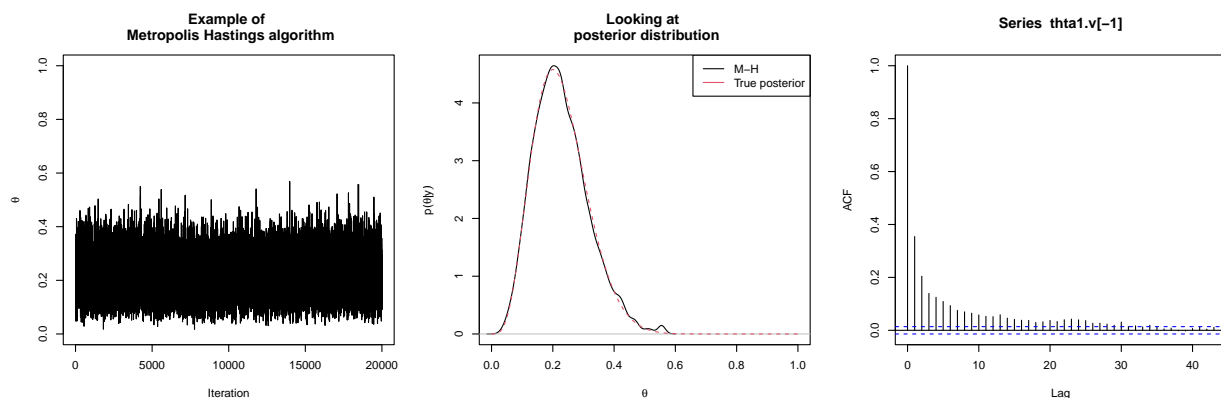
#Construct simulations.
phat<-y/20; sdphat<-sqrt(phat*(1-phat)/20)
ap<-1; bp<-1 #hyper-parameters of prior distributions.
thta1<-phat #Starting point. Compared to the true theta.
thta1.v<-rep(thta1,iter+1); indi<-0; iter =20000 #number of iterations.
rj=1 #ratio of MLE std deviation for conditional distribution

for(i in 1:iter){
#Generate truncated normal without truncated normal function
thta1.t<-rnorm(1,phat,rj*sdphat)
while(abs(thta1.t-0.5)> 0.5) {thta1.t<-rnorm(1,phat,rj*sdphat)}

#working with log densities.
r_up<-dbinom(y,20,thta1.t,log=TRUE) + dbeta(thta1.t,ap,bp,log=TRUE) +
  dnorm(thta1.v[i],phat,rj*sdphat,log=TRUE)
r_low<-dbinom(y,20,thta1.v[i],log=TRUE) + dbeta(thta1.v[i],ap,bp,log=TRUE) +
  dnorm(thta1.t,phat,rj*sdphat,log=TRUE)
r<-r_up-r_low
#Draw an indicator whether to accept/reject candidate
ind<-rbinom(1,1,exp( min(c(r,0)) ) )
thta1.v[i+1]<- ind*thta1.t + (1-ind)*thta1.v[i]
indi<-indi+ind
}

#Example of plot of iterations.
par(mfrow=c(1,3))
plot(thta1.v[-1],xlab='Iteration',ylab=expression(theta),ylim=c(0,1),
     main='Example of \n Metropolis Hastings algorithm',type='l')
#Compare to posterior.
plot(density(thta1.v[-1]),xlab=expression(theta), ylab=expression(paste(p,"(",theta,group("|",y,")"))),
     xlim=c(0,1), main='Looking at \n posterior distribution')
curve(dbeta(x,y+ap,20-y+bp),add=TRUE,col=2,lty=2)
legend('topright',legend=c('M-H','True posterior'),lwd=1,col=1:2)
acf(thta1.v[-1])

```



```
indi
```

```
## [1] 17239
```

```
head(thta1.v[-1])
```

```
## [1] 0.3053750 0.1602193 0.3018484 0.3465536 0.3123646 0.1467432
```

1.3 Gibbs sampler: Binomial likelihood with Beta prior.

Question: Does this make sense?

Gibbs sampling is specifically designed to deal with multi-parameter problems. The binomial likelihood with beta prior is an example of a single parameter problem so Gibbs sampling is not applicable. However it is an example where the posterior distribution is known analytically. Therefore, it helps understand the performance of Metropolis-Hastings algorithms, as we can compare the output from the Markov chain to the exact result.

2 Example two: Bivariate Normal

In this example, assume $\mathbf{y}_i = (y_{1i}, y_{2i})'$ are drawn from a bivariate normal distribution where each component marginally has the same variance.

$$p(\mathbf{y}|\boldsymbol{\mu}, \sigma^2, \rho) = \frac{1}{2\pi\sigma^2\sqrt{(1-\rho^2)}} e^{-\frac{(\mathbf{y}-\boldsymbol{\mu})'\mathbf{R}^{-1}(\mathbf{y}-\boldsymbol{\mu})}{2\sigma^2}}.$$

where $\boldsymbol{\mu} = (\mu_1 \ \mu_2)$, $\mathbf{R} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \Rightarrow \mathbf{R}^{-1} = \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}$.

For most demonstrations of this problem, we will assume that the parameters requiring estimation are μ_1, μ_2, σ^2 . In some cases, we will extend this and assume ρ is not fixed and also requires estimation.

Note if \mathbf{R} is known, this is just an extension of the normal likelihood considered in lab 2, so is analytically known. Therefore we will consider a flat prior for μ_1, μ_2 and assume $p(\sigma^2) \propto (\sigma^2)^{-1}$.

In this case, the joint distribution is

$$\begin{aligned} p(\boldsymbol{\mu}, \sigma^2, \mathbf{y}) &= (\sigma^2)^{-1} \prod_{i=1}^n \frac{1}{2\pi\sigma^2\sqrt{(1-\rho^2)}} e^{-\frac{(\mathbf{y}_i-\boldsymbol{\mu})'\mathbf{R}^{-1}(\mathbf{y}_i-\boldsymbol{\mu})}{2\sigma^2}} \\ &= (\sigma^2)^{-1} \frac{1}{(2\pi\sigma^2)^n(1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i-\bar{\mathbf{y}}+\bar{\mathbf{y}}-\boldsymbol{\mu})'\mathbf{R}^{-1}(\mathbf{y}_i-\bar{\mathbf{y}}+\bar{\mathbf{y}}-\boldsymbol{\mu})}{2\sigma^2}} \\ &= (\sigma^2)^{-1} \frac{1}{(2\pi\sigma^2)^n(1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i-\bar{\mathbf{y}})'\mathbf{R}^{-1}(\mathbf{y}_i-\bar{\mathbf{y}})}{2\sigma^2}} e^{-\frac{n(\bar{\mathbf{y}}-\boldsymbol{\mu})'\mathbf{R}^{-1}(\bar{\mathbf{y}}-\boldsymbol{\mu})}{2\sigma^2}} \end{aligned}$$

Marginalising out $\boldsymbol{\mu}$ from this distribution gives,

$$\begin{aligned} p(\sigma^2, \mathbf{y}) &= \int p(\boldsymbol{\mu}, \sigma^2, \mathbf{y}) d\boldsymbol{\mu} \\ &= (\sigma^2)^{-1} \frac{1}{(2\pi\sigma^2)^n(1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i-\bar{\mathbf{y}})'\mathbf{R}^{-1}(\mathbf{y}_i-\bar{\mathbf{y}})}{2\sigma^2}} \int e^{-\frac{n(\bar{\mathbf{y}}-\boldsymbol{\mu})'\mathbf{R}^{-1}(\bar{\mathbf{y}}-\boldsymbol{\mu})}{2\sigma^2}} d\boldsymbol{\mu} \\ &= (\sigma^2)^{-1} \frac{1}{(2\pi\sigma^2)^n(1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i-\bar{\mathbf{y}})'\mathbf{R}^{-1}(\mathbf{y}_i-\bar{\mathbf{y}})}{2\sigma^2}} 2\pi\sqrt{\det(\sigma^2\mathbf{R}/n)} \\ &\propto \frac{1}{(\sigma^2)^{\frac{2n-2}{2}+1}(1-\rho^2)^{n/2-1}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i-\bar{\mathbf{y}})'\mathbf{R}^{-1}(\mathbf{y}_i-\bar{\mathbf{y}})}{2\sigma^2}} \end{aligned}$$

which indicates that $\sigma^2|\mathbf{y} \sim \text{InvGa}((2n-2)/2, (2n-2)s^2/2) = \text{Inv}\chi^2(2n-2, s^2)$, where $s^2 = \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \mathbf{R}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}}) / (2n-2)$. Note that in practice it is easier to work with precisions rather than variances. For the precision $\tau = (\sigma^2)^{-1}$, the marginal posterior is Gamma.

Similarly, integrating out σ^2 from the joint distribution

$$\begin{aligned} p(\mu, \mathbf{y}) &= \int p(\mu, \sigma^2, \mathbf{y}) d\sigma^2 \\ &= \frac{1}{(2\pi)^n (1-\rho^2)^{n/2}} \int (\sigma^2)^{-(n+1)} e^{-\frac{(n-2)s^2 + n(\bar{\mathbf{y}} - \mu)' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \mu)}{2\sigma^2}} d\sigma^2 \\ &= \frac{1}{(2\pi)^n (1-\rho^2)^{n/2}} \Gamma(n) \left(\frac{(2n-2)s^2 + n(\bar{\mathbf{y}} - \mu)' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \mu)}{2} \right)^{-n} \\ &\propto \left(1 + \frac{n(\bar{\mathbf{y}} - \mu)' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \mu)}{(2n-2)s^2} \right)^{-(2n-2+2)/2} \end{aligned}$$

reveals that the marginal posterior $\mu|\mathbf{y}$ is multivariate t with location parameter \bar{y} , scale parameter $s^2 \mathbf{R}/n$ and $2n-2$ degrees of freedom. By implication, the marginal posterior of the individual components of $\mu, \mu_i; i = 1, 2$ are univariate t with location parameter \bar{y}_i , scale parameter $s^2 \mathbf{R}_{ii}/n = s^2/n$ and $\text{df} = 2n-2$.

For the purposes of data generation, assume $n = 80$, $\rho = 0.5$, $\mu_1 = 3, \mu_2 = 6$ and $\sigma^2 = 2$. In addition, estimate s^2 and $\bar{\mathbf{y}}$.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.1
```

```
n=80
rho=0.5
R = matrix(c(1,rho,rho,1),2,2)
y<-mvrnorm(n,mu=c(3,6),Sigma=2*R)
bary<-colMeans(y)
Rinv <- solve(R)
ydiff<-y -rep(1,n)%*%t(bary)
s2<- sum(diag(ydiff%*%Rinv%*%t(ydiff)))/(2*n-2)
```

2.1 Demonstration of Gibbs sampler

We now consider Gibbs sampler for generating random vectors from the posterior distribution of μ_1, μ_2 and σ^2 .

To determine the conditional posterior distributions necessary to constructing a Gibbs sampler, let's look at the joint distribution.

$$p(\mu, \sigma^2, \mathbf{y}) = \frac{1}{(2\pi)^n (\sigma^2)^{n+1} (1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \mathbf{R}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}})}{2\sigma^2}} e^{-\frac{n(\bar{\mathbf{y}} - \mu)' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \mu)}{2\sigma^2}}$$

and for each parameter of interest μ_1, μ_2, σ^2 extract the kernel.

For σ^2 , the kernel is

$$\frac{1}{(\sigma^2)^{n+1}} e^{-\frac{(2n-2)s^2 + n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})}{2\sigma^2}}$$

which indicating the conditional posterior $\sigma^2 | \boldsymbol{\mu}, \mathbf{R}, y$ is $\text{InvGa}(n, (n-1)s^2 + n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})/2)$.

For $\boldsymbol{\mu}$, the kernel is

$$e^{-\frac{n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})}{2\sigma^2}}$$

For μ_1 and μ_2 , we need to expand $(\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})$ to determine the kernel. This expansion is,

$$\begin{aligned} (\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu}) &= (\bar{y}_1 - \mu_1)^2 \mathbf{R}_{11}^{-1} + (\bar{y}_2 - \mu_2)^2 \mathbf{R}_{22}^{-1} + 2(\bar{y}_1 - \mu_1)(\bar{y}_2 - \mu_2) \mathbf{R}_{12}^{-1} \\ &\propto \begin{cases} \mu_1^2 \mathbf{R}_{11}^{-1} - 2\mu_1(\bar{y}_1 \mathbf{R}_{11}^{-1} + (\bar{y}_2 - \mu_2) \mathbf{R}_{12}^{-1}) = \frac{\mu_1^2 - 2\mu_1(\bar{y}_1 - \rho(\bar{y}_2 - \mu_2))}{1 - \rho^2} & \text{for } \mu_1 \\ \mu_2^2 \mathbf{R}_{22}^{-1} - 2\mu_2(\bar{y}_2 \mathbf{R}_{22}^{-1} + (\bar{y}_1 - \mu_1) \mathbf{R}_{12}^{-1}) = \frac{\mu_2^2 - 2\mu_2(\bar{y}_2 - \rho(\bar{y}_1 - \mu_1))}{1 - \rho^2} & \text{for } \mu_2 \end{cases} \end{aligned}$$

From this, we can deduce the conditional posterior distributions are,

$$\begin{aligned} \mu_1 | \mu_2, \sigma^2, y &\sim \mathcal{N}(\bar{y}_1 - \rho(\bar{y}_2 - \mu_2), \sigma^2(1 - \rho^2)/n) \\ \mu_2 | \mu_1, \sigma^2, y &\sim \mathcal{N}(\bar{y}_2 - \rho(\bar{y}_1 - \mu_1), \sigma^2(1 - \rho^2)/n). \end{aligned}$$

Now we are in the position to implement a Gibbs sampler. However to make things easier to code, we will work with $\tau = (\sigma^2)^{-1}$, so that the conditional posterior is Gamma.

```
#Generate initial values for mu_1, mu_2, sigma^2.
mu<-rnorm(2,2/n)
tau<-rgamma(1,n,2*(n-1))
sigma2<-1/tau
#Specify number of iterations.
iter = 5000
#specify correlation.
#Storing draws
para.est<-matrix(0,iter,3)

#Running the simulations.
for(i in 1:iter){
  mu[1] <-rnorm(1,bary[1] - rho*(bary[2]-mu[2]), sd= sqrt(sigma2*(1-rho^2)/n))
  mu[2] <-rnorm(1,bary[2] - rho*(bary[1]-mu[1]), sd= sqrt(sigma2*(1-rho^2)/n))
  SSmu <-t(bary-mu)%*%Rinv%*(bary-mu)
  tau <-rgamma(1,n,(n-1)*s2 + 0.5*n*SSmu)
  sigma2<-1/tau
  para.est[i,]<-c(mu,sigma2)
}

#Comparison of results to true posterior.
par(mfrow=c(1,3))
plot(density(para.est[-c(1:100),1]),xlab=expression(mu[1]),
     ylab=expression(paste(mu[1], '~p(', mu[1], '|y)')), main=' Posterior distribution')
mseq=seq(-5,5,by=0.001)
lines(mseq*sqrt(s2/n)+bary[1],dt(mseq,df=2*n-2)*sqrt(n/s2),
     xlim=c(min(para.est[,1], max(para.est[,1])), col=2, lty=2)
```

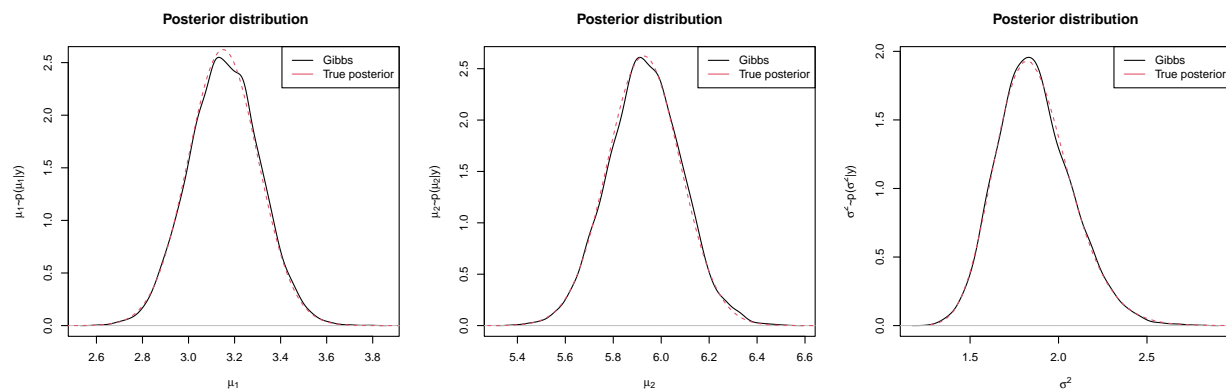
```

legend('topright',legend=c('Gibbs','True posterior'),lwd=1,col=1:2)

plot(density(para.est[-c(1:100),2]),xlab=expression(mu[2]),
     ylab=expression(paste(mu[2], '~p(', mu[2], '|y)')),main='Posterior distribution')
mse=seq(-5,5,by=0.001)
lines(mse*sqrt(s2/n)+bary[2],dt(mse,df=2*n-2)*sqrt(n/s2),
      xlim=c(min(para.est[,2], max(para.est[,2])),col=2,lty=2)
legend('topright',legend=c('Gibbs','True posterior'),lwd=1,col=1:2)

plot(density(para.est[-c(1:100),3]),xlab=expression(sigma^2),
     ylab=expression(paste(sigma^2, '~p(', sigma^2, '|y)')),main='Posterior distribution')
curve(dgamma(1/x,n-1,(n-1)*s2)/x^2,xlim=c(min(para.est[,3]),max(para.est[,3])),col=2,lty=2,add=TRUE)
legend('topright',legend=c('Gibbs','True posterior'),lwd=1,col=1:2)

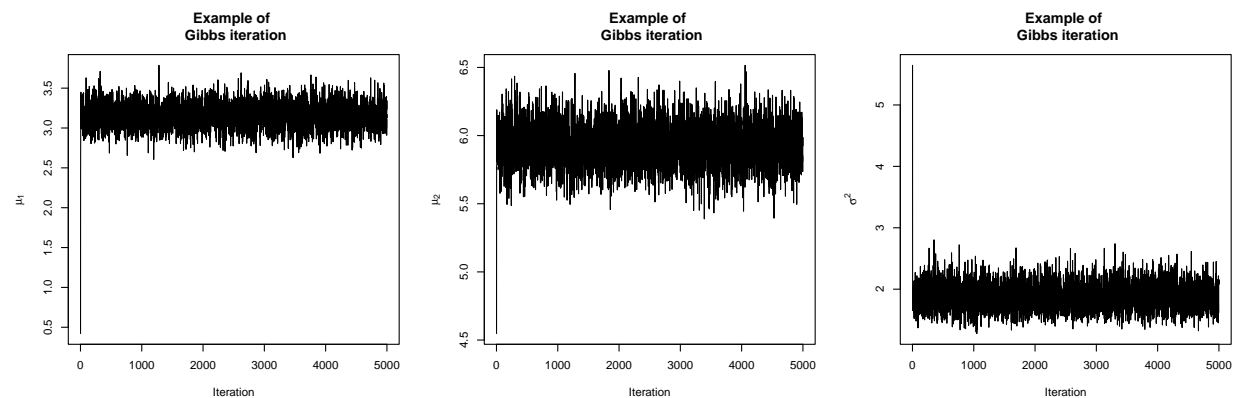
```



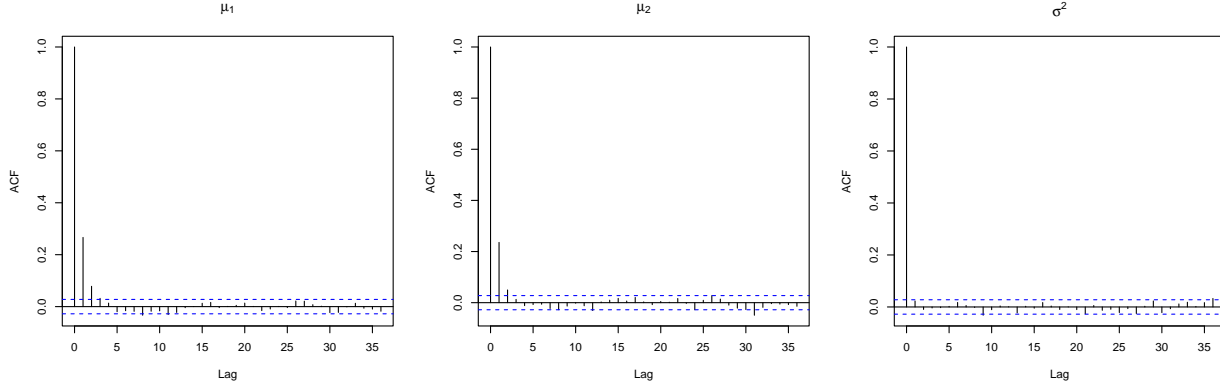
```

#Looking at the iterations.
par(mfrow=c(1,3))
plot(para.est[1:iter,1],xlab='Iteration',ylab=expression(mu[1]),
     main='Example of \n Gibbs iteration',type='l')
plot(para.est[1:iter,2],xlab='Iteration',ylab=expression(mu[2]),
     main='Example of \n Gibbs iteration',type='l')
plot(para.est[1:iter,3],xlab='Iteration',ylab=expression(sigma^2),
     main='Example of \n Gibbs iteration',type='l')

```



```
#acf plot
par(mfrow=c(1,3))
acf(para.est[,1],main=expression(mu[1]),cex.main=2)
acf(para.est[,2],main=expression(mu[2]),cex.main=2)
acf(para.est[,3],main=expression(sigma^2),cex.main=2)
```



2.2 Demonstration of M-H algorithm in Gibbs

Note in the above example of the Gibbs sampler, we did not attempt to estimate ρ , only σ^2, μ_1, μ_2 . Now we also want to estimate the posterior of ρ . Based on the joint distribution, the kernel of ρ is,

$$\begin{aligned} p(\boldsymbol{\mu}, \sigma^2, \mathbf{y}, \rho) &= \frac{1}{(2\pi)^n (\sigma^2)^{n+1} (1-\rho^2)^{n/2}} e^{-\frac{\sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})' \mathbf{R}^{-1} (\mathbf{y}_i - \bar{\mathbf{y}})}{2\sigma^2}} e^{-\frac{n(\bar{\mathbf{y}} - \boldsymbol{\mu})' \mathbf{R}^{-1} (\bar{\mathbf{y}} - \boldsymbol{\mu})}{2\sigma^2}} \\ &= \frac{e^{-\frac{\sum_{i=1}^n (y_{i1} - \bar{y}_1)^2 + \sum_{i=1}^n (y_{i2} - \bar{y}_2)^2 - 2\rho \sum_{i=1}^n (y_{i1} - \bar{y}_1)(y_{i2} - \bar{y}_2)}{2\sigma^2(1-\rho^2)}}}{(2\pi)^n (\sigma^2)^{n+1} (1-\rho^2)^{n/2}} e^{-\frac{n(\bar{y}_1 - \mu_1)^2 + n(\bar{y}_2 - \mu_2)^2 - 2n\rho(\bar{y}_1 - \mu_1)(\bar{y}_2 - \mu_2)}{2\sigma^2(1-\rho^2)}}. \end{aligned}$$

From this, we can deduce that the kernel for ρ is,

$$K(\rho) \propto \frac{e^{-\frac{\sum_{i=1}^n (y_{i1} - \bar{y}_1)^2 + \sum_{i=1}^n (y_{i2} - \bar{y}_2)^2 - 2\rho \sum_{i=1}^n (y_{i1} - \bar{y}_1)(y_{i2} - \bar{y}_2)}{2\sigma^2(1-\rho^2)}}}{(1-\rho^2)^{n/2}} e^{-\frac{n(\bar{y}_1 - \mu_1)^2 + n(\bar{y}_2 - \mu_2)^2 - 2n\rho(\bar{y}_1 - \mu_1)(\bar{y}_2 - \mu_2)}{2\sigma^2(1-\rho^2)}}$$

This is not a distribution you will readily recognise, so in order to estimate ρ , we will incorporate a Metropolis step into our Gibbs sampler. We know $\rho \in (-1, 1)$, so we can select a symmetric conditional distribution $U(-1, 1)$, making the calculation of the acceptance rate α easier. Similarly, we will use $U(-1, 1)$ as a prior for ρ , so that the ratio $p(\rho^{(t)}|y)/p(\rho^{(t-1)}|y)$ is just the ratio of bivariate normal densities evaluated at $\rho^{(t)}, \rho^{(t-1)}$. Lastly, by choosing to estimate ρ , we will need to update s^2 at each iteration when drawing from the conditional posterior of σ^2 . Moreover the marginal posterior for σ^2 determined earlier is no longer valid, as it was still conditional on knowing ρ .

```
#Generate initial values for mu_1, mu_2, sigma^2, \rho.
library(mvtnorm) #package that allows us to calculate multivariate densities.
```

```
## Warning: package 'mvtnorm' was built under R version 4.3.1
```



```

mu<-rnorm(2)
tau<-rgamma(1,1,1)
sigma2<-1/tau
rho0 <-runif(1,-1,1)

#Specify number of iterations.
iter = 10000
#specify correlation.
#Storing draws
para.est<-matrix(0,iter,4)

#Counter for acceptances.
indi<-0
#Running the simulations.
for(i in 1:iter){
  mu[1] <-rnorm(1,bary[1] - rho0*(bary[2]-mu[2]), sd= sqrt(sigma2*(1-rho0^2)/n))
  mu[2] <-rnorm(1,bary[2] - rho0*(bary[1]-mu[1]), sd= sqrt(sigma2*(1-rho0^2)/n))
  R.t <-matrix(c(1,rho0,rho0,1),2,2)
  Rinv.t<-solve(R.t)
  s2.t <-sum(diag(ydiff%*%Rinv.t%*%t(ydiff)))/(2*n-2)
  SSmu <-t(bary-mu)%*%Rinv.t%*%(bary-mu)
  tau <-rgamma(1,n,(n-1)*s2.t + 0.5*n*SSmu)
  sigma2<-1/tau

  #update rho as Metroplis.
  rhoc <-runif(1,-1,1)
  R.c <-matrix(c(1,rhoc,rhoc,1),2,2)
  #Calculate difference in log densities.
  r_up<- sum(dmvnorm(y, mean=mu, sigma = sigma2*R.c, log=TRUE))
  r_low<- sum(dmvnorm(y, mean=mu, sigma = sigma2*R.t, log=TRUE))
  r<-r_up-r_low
  #Draw an indicator whether to accept/reject candidate
  ind<-rbinom(1,1,exp( min(c(r,0)) ) )
  rho0<- ind*rhoc + (1-ind)*rho0 #new state for rho.
  indi<-indi+ind
  para.est[i,]<-c(mu,sigma2,rho0)
}

#Comparison of results to true posterior.
par(mfrow=c(2,2))
plot(density(para.est[-c(1:50),1]),xlab=expression(mu[1]),
      ylab=expression(paste(mu[1], '~p(', mu[1], '|y)' )),main=' Posterior distribution')
mseq=seq(-5,5,by=0.001)
lines(mseq*sqrt(s2/n)+bary[1],dt(mseq,df=2*n-2)*sqrt(n/s2),
      xlim=c(min(para.est[,1], max(para.est[,1]))),col=2,lty=2)
legend('topright',legend=c('Gibbs','True posterior'),lwd=1,col=1:2)

plot(density(para.est[-c(1:50),2]),xlab=expression(mu[2]),
      ylab=expression(paste(mu[2], '~p(', mu[2], '|y)' )),main='Posterior distribution')
mseq=seq(-5,5,by=0.001)
lines(mseq*sqrt(s2/n)+bary[2],dt(mseq,df=2*n-2)*sqrt(n/s2),
      xlim=c(min(para.est[,2], max(para.est[,2]))),col=2,lty=2)

```

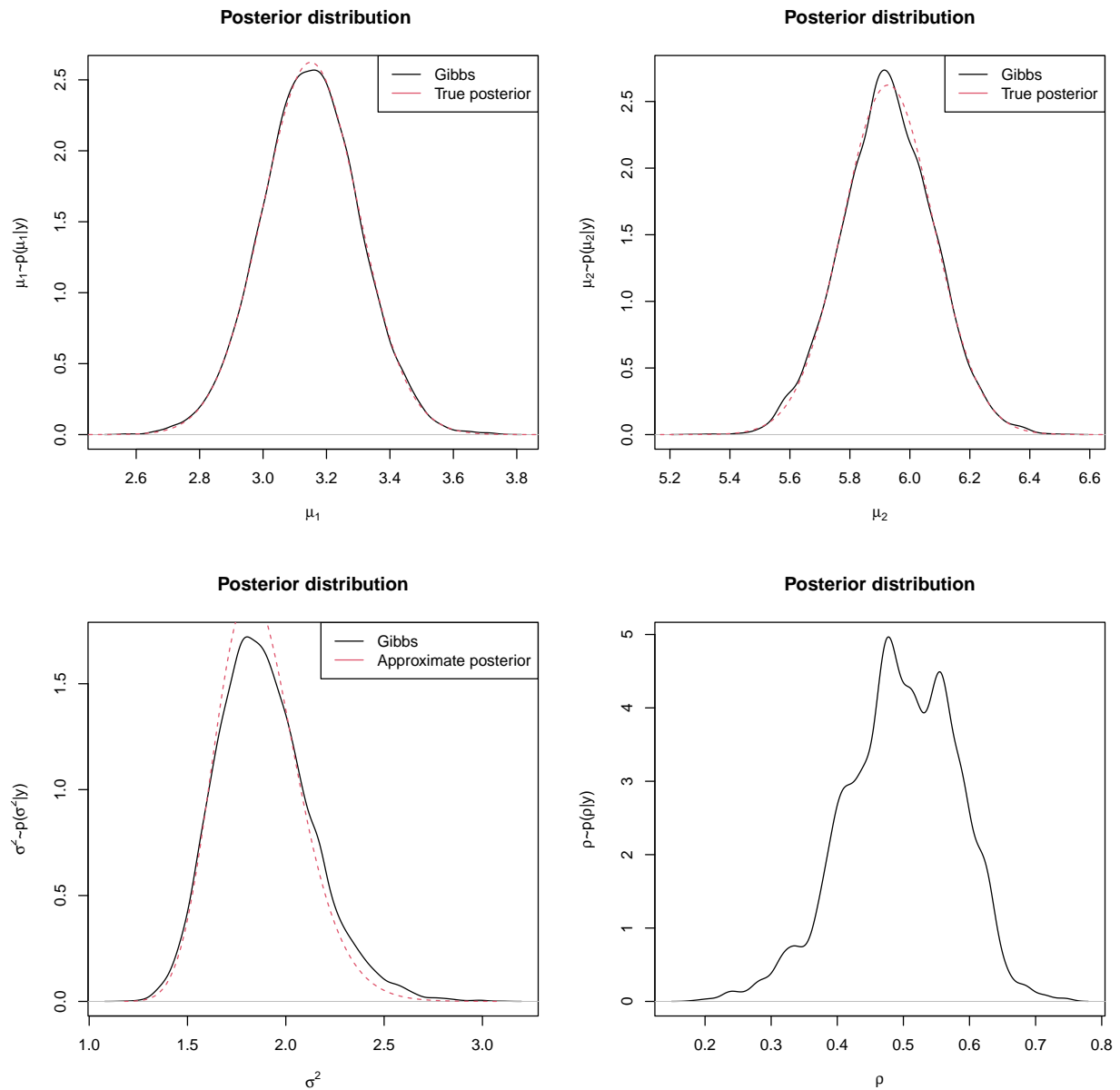
```

legend('topright',legend=c('Gibbs','True posterior'),lwd=1,col=1:2)

plot(density(para.est[-c(1:50),3]),xlab=expression(sigma^2),
      ylab=expression(paste(sigma^2,'~p(',sigma^2,'|y)')),main='Posterior distribution')
curve(dgamma(1/x,n-1,(n-1)*s2)/x^2,
      xlim=c(min(para.est[-c(1:50),3]),max(para.est[-c(1:50),3])),col=2,lty=2,add=TRUE)
legend('topright',legend=c('Gibbs','Approximate posterior'),lwd=1,col=1:2)

plot(density(para.est[-c(1:50),4]),xlab=expression(rho),
      ylab=expression(paste(rho,'~p(',rho,'|y)')),main='Posterior distribution')

```



```

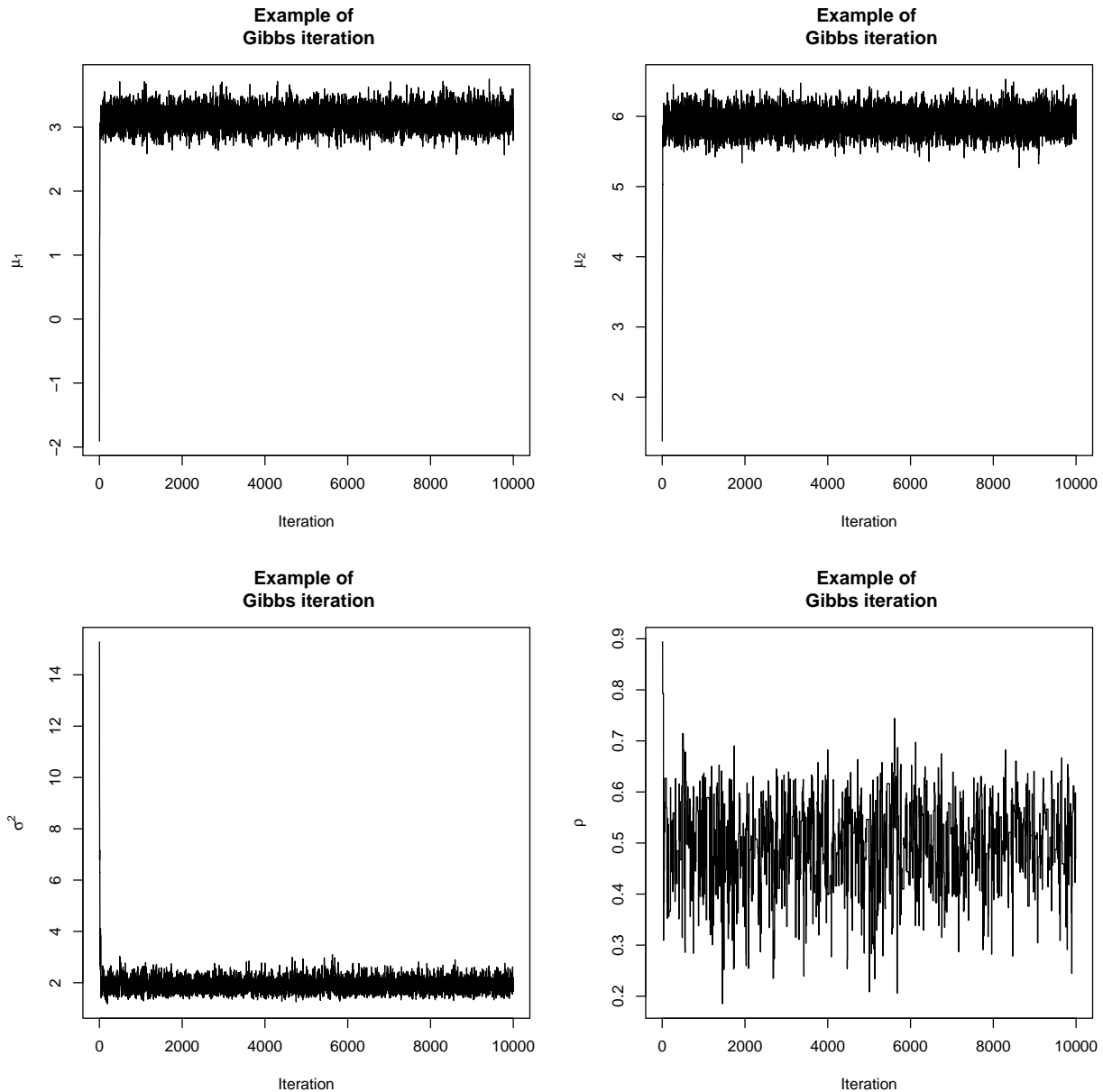
#Looking at the iterations.
par(mfrow=c(2,2))

```

```

plot(para.est[1:iter,1],xlab='Iteration',ylab=expression(mu[1]),
     main='Example of \n Gibbs iteration',type='l')
plot(para.est[1:iter,2],xlab='Iteration',ylab=expression(mu[2]),
     main='Example of \n Gibbs iteration',type='l')
plot(para.est[1:iter,3],xlab='Iteration',ylab=expression(sigma^2),
     main='Example of \n Gibbs iteration',type='l')
plot(para.est[1:iter,4],xlab='Iteration',ylab=expression(rho),
     main='Example of \n Gibbs iteration',type='l')

```



```

#Report number of iterations accepted in metropolis component
indi

```

```
## [1] 1145
```

```
#summary of correlation
summary(para.est[,4])
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1852  0.4415  0.4996  0.4973  0.5585  0.8939
```

```
cor(y[,1],y[,2]) #Compare to empirical correlation.
```

```
## [1] 0.5127177
```

3 Final comments

While we can see that Metropolis-Hastings and Gibbs sampler provide us with dependent random samples from the posterior (target) distribution, there are nevertheless some issues with just immediately taking the output of our results.

3.1 Early iterations.

In some of the examples in this lecture, we saw that the initial parameter values were drawn well outside the range of plausible parameter values suggested by the posterior. Moreover we saw that it could take a large number of iterations for the Markov chain to cease to be influenced by the initial choice for parameters. This would especially be the case for Metropolis-Hastings if the acceptance rate is low.

Keeping early iterations, before the chain has converged to the posterior distribution, is therefore not a good idea. Discarded early iterations are often referred to as *burn-in* or *warm-up*.

3.2 The use of dependent samples.

As noted in the previous lecture, samples generated from Metropolis-Hastings algorithm and Gibbs sampler are Markov chains. This means that the samples are dependent. We saw this in the auto-correlation plots produced by various examples in this lecture. While the existence of correlation between samples did not lead to a problem when attempting to empirically estimate the posterior density, there may be situations where correlated samples are not desirable. However, recent literature has shown that this kind of dependency is weak, which will not have bad impacts on learning.