

# Bayesian\_hw5

Ching-Tsung\_Deron\_Tsai

2022/10/31

```
library(tidyverse)
library(MASS)
library(coda)
library(MCMCpack)
library(mvtnorm)
sample.theta.j <- function(n.j,sigma2,mu,ybar.j,tau2){
  post.var <- 1/((n.j/sigma2)+(1/tau2))
  post.mean <- post.var*((n.j/sigma2)*ybar.j)+(mu/tau2)
  new.theta.j <- rnorm(1,post.mean,sqrt(post.var))
  return(new.theta.j)
}

sample.sigma2 <- function(n.j,y,theta.j,nu.0,sigma2.0,m,n){
  theta.j.expanded <- NULL
  for(i in 1:m){
    theta.j.expanded <- c(theta.j.expanded,rep(theta.j[i],n.j[i]))
  }
  nu.n <- nu.0+sum(n.j)
  sigma2.n <- (1/nu.n)*((nu.0*sigma2.0)+sum((y-theta.j.expanded)^2))
  new.sigma2 <- 1/rgamma(1,(nu.n/2),((nu.n*sigma2.n)/2))
  return(new.sigma2)
}

sample.mu <- function(theta.j,m,tau2,mu.0,gamma2.0){
  post.var <- 1/((m/tau2)+(1/gamma2.0))
  theta.bar <- mean(theta.j)
  post.mean <- post.var*((m/tau2)*theta.bar)+(mu.0/gamma2.0)
  new.mu <- rnorm(1,post.mean,sqrt(post.var))
  return(new.mu)
}

sample.tau2 <- function(theta.j,m,mu,eta.0,tau2.0){
  eta.n <- m+eta.0
  tau2.n <- (1/eta.n)*((eta.0*tau2.0)+sum((theta.j-mu)^2))
  new.tau2 <- 1/rgamma(1,(eta.n/2),((eta.n*tau2.n)/2))
  return(new.tau2)
}

sample.beta.metrop <- function(beta.k,var.proposal,y,X,beta0,Sigma){
  accept <- 0
  beta.star <- mvrnorm(1,beta.k,var.proposal)
  # change to dbinom for logistic regression instead of Poisson regression in lecture code:
```

```

log.num <- sum(dbinom(y,1,exp(X%*%beta.star)/(1+exp(X%*%beta.star)),log=TRUE))
               +dmvnorm(beta.star,beta0,Sigma,log=TRUE)
log.den <- sum(dbinom(y,1,exp(X%*%beta.k)/(1+exp(X%*%beta.k)),log=TRUE))
               +dmvnorm(beta.k,beta0,Sigma,log=TRUE)
r <- exp(log.num-log.den)
u <- runif(1,0,1)
if(r >1){
  new.beta <- beta.star
}
if(r <=1){
  if(u <r){
    new.beta <- beta.star
  }
  if(u >= r){
    new.beta <- beta.k
  }
}
if(sum(new.beta==beta.star)==length(beta.star)){
  accept <- 1
}
out <- list(new.beta=new.beta,accept=accept)
return(out)
}

```

## 8.3

a

```

nms = paste0("../HW_Q/school",1:8,".dat")
y = lapply(nms, function(x) unlist(read.table(x)))
# names(y) <- 1:8
# data.frame(gp=apply(names(y),function(i) rep(i,length(y[[i]]))) %>% unlist(), y=unlist(y))

# params:
mu.0 = 7 ; gamma2.0 = 5 ; tau2.0 = 10
eta.0 = nu.0 = 2 ; sigma2.0 = 15
m = length(y)
n.j = map_dbl(y,length)
# initial values:
init.theta.j = map_dbl(y,mean)
init.mu = mean(init.theta.j)
init.sigma2 = map_dbl(y,var) %>% mean()
init.tau2 = var(init.theta.j)

# group specific parameters:
S=1e4
theta.MCMC <- matrix(0,nrow=S,ncol=m)
# other parameters, mu, sigma2, tau2:
other.pars.MCMC <- matrix(0,S,ncol=3)
new.theta.j <- rep(0,m)

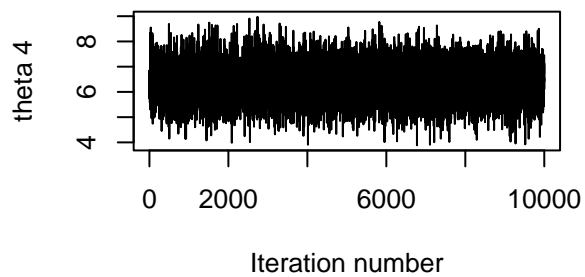
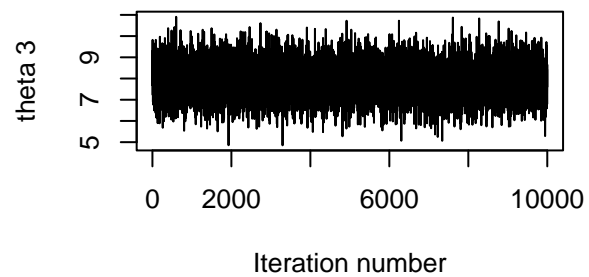
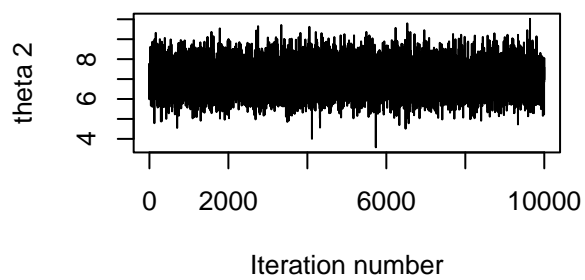
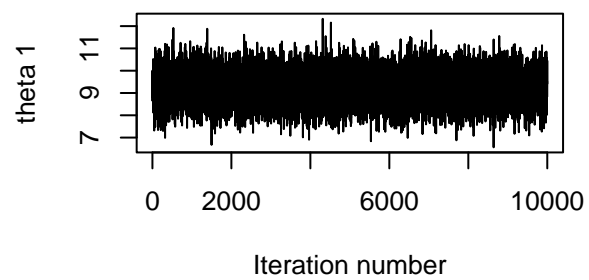
# MCMC:

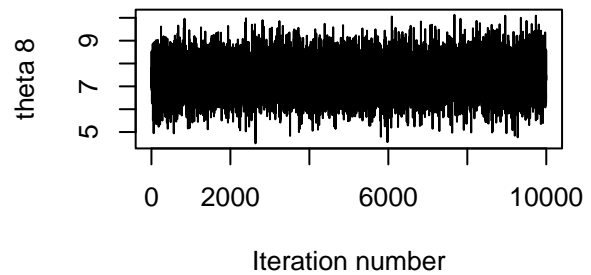
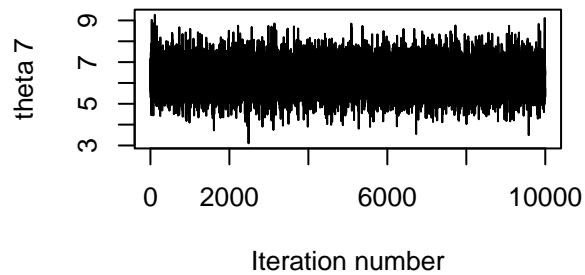
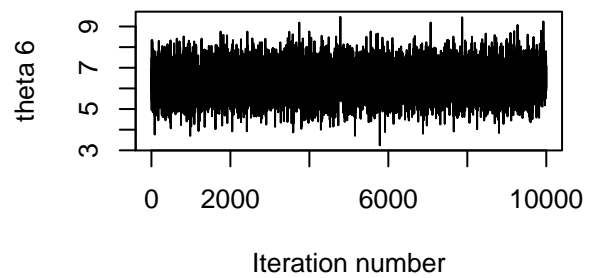
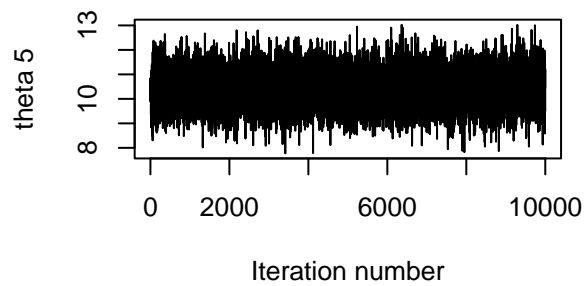
```

```

set.seed(0)
for(k in 1:S){
  if(k==1){
    theta.j <- init.theta.j
    mu <- init.mu
    sigma2 <- init.sigma2
    tau2 <- init.tau2
  }
  new.mu <- sample.mu(theta.j,m,tau2,mu.0,gamma2.0)
  new.tau2 <- sample.tau2(theta.j,m,new.mu,eta.0,tau2.0)
  new.sigma2 <- sample.sigma2(n.j,unlist(y),theta.j,nu.0,sigma2.0,m,n)
  for(l in 1:m){
    new.theta.j[l] <- sample.theta.j(n.j[l],new.sigma2,new.mu,init.theta.j[l],new.tau2)
  }
  mu <- new.mu
  tau2 <- new.tau2
  sigma2 <- new.sigma2
  theta.j <- new.theta.j
  theta.MCMC[k,] <- theta.j
  other.pars.MCMC[k,] <- c(mu,sigma2,tau2)
}
colnames(other.pars.MCMC) <- c("mu","sigma2","tau2")
# diagnosis:
# trace plot
par(mfrow=c(2,2))
for (i in 1:8){
  lb = paste("theta",i)
  plot(theta.MCMC[,i],xlab="Iteration number",ylab=lb,type="l")
}

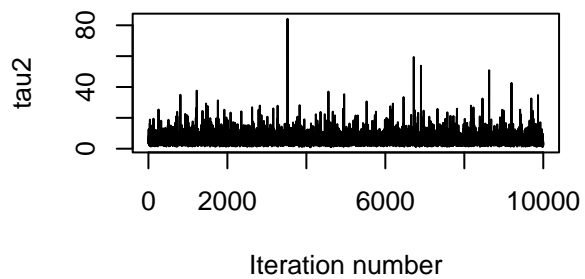
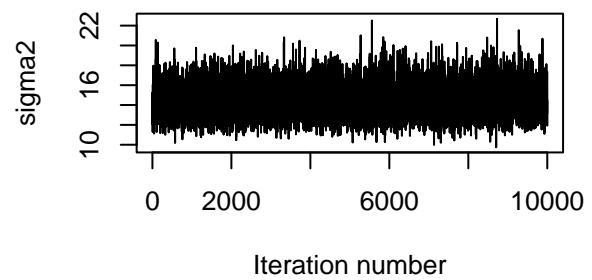
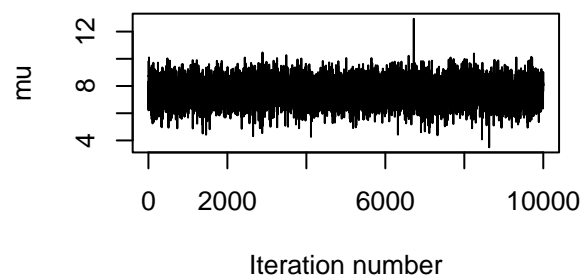
```



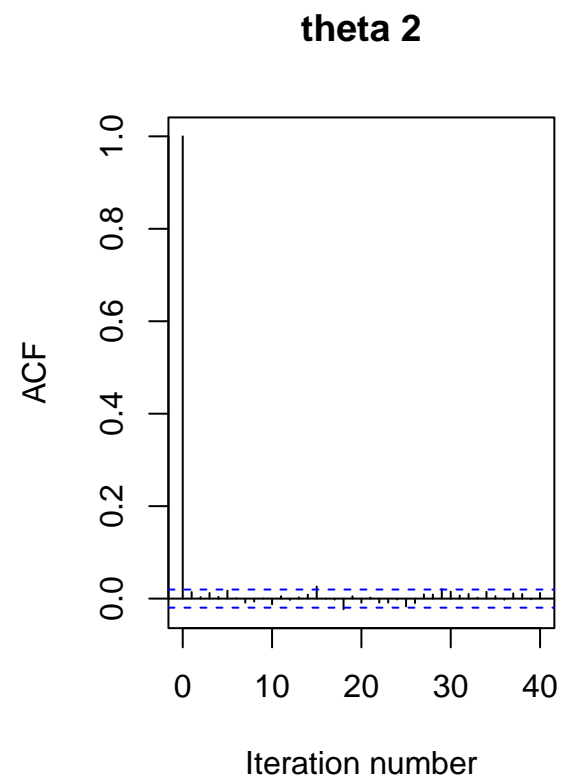
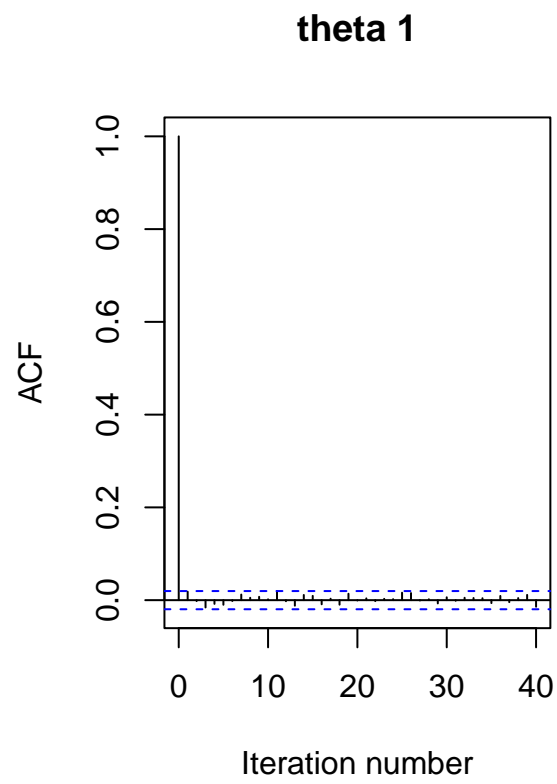


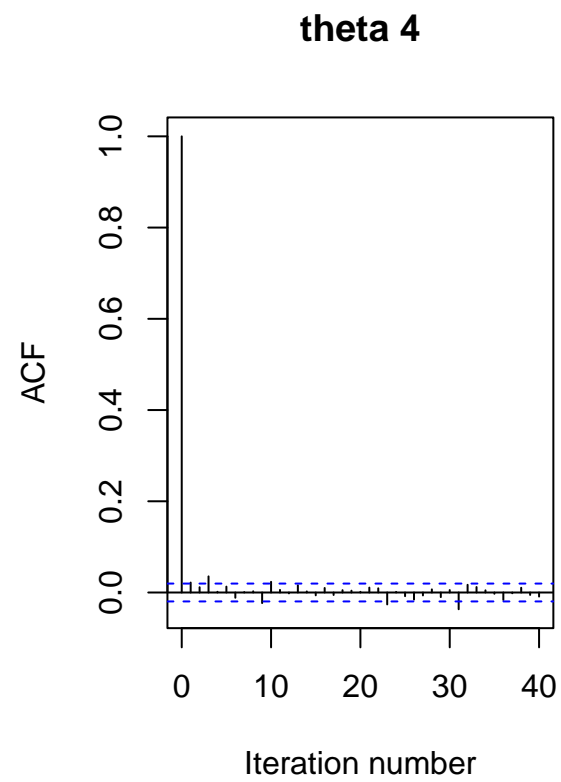
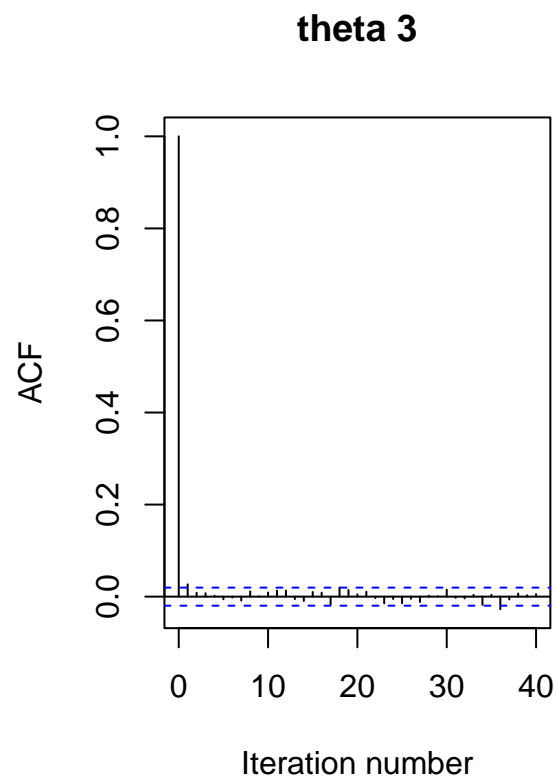
```
for (i in 1:3){
  lb = colnames(other.pars.MCMC)[i]
  plot(other.pars.MCMC[,i],xlab="Iteration number",ylab=lb,type="l")
}
# -> didn't observe clear pattern, may have some convergence from 1-500/1000 for some thetas
#   use 1000 as burn-in period for all

# auto correlation:
par(mfrow=c(1,2))
```

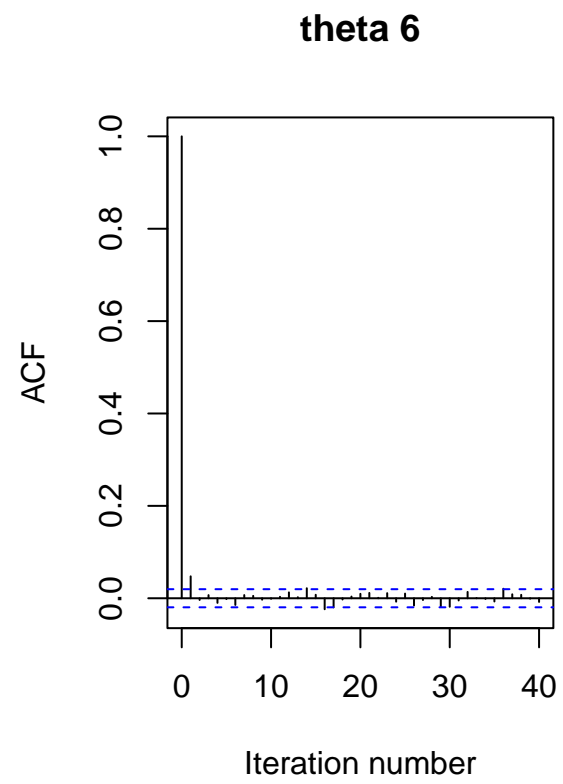
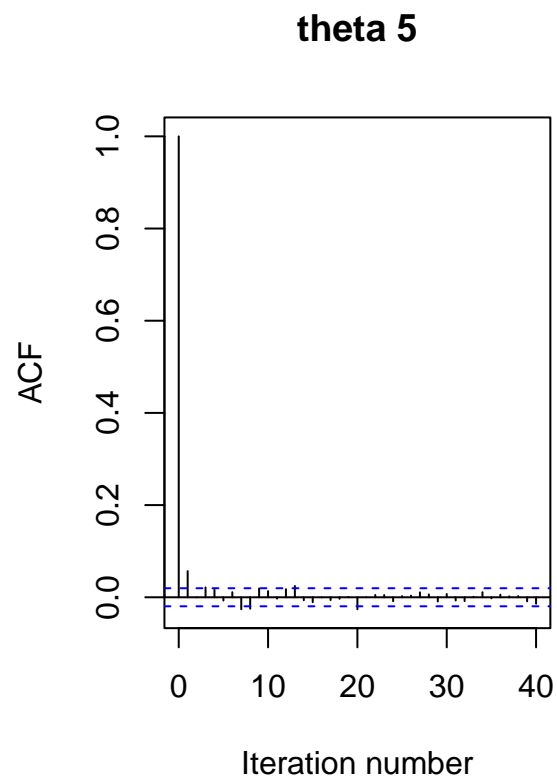


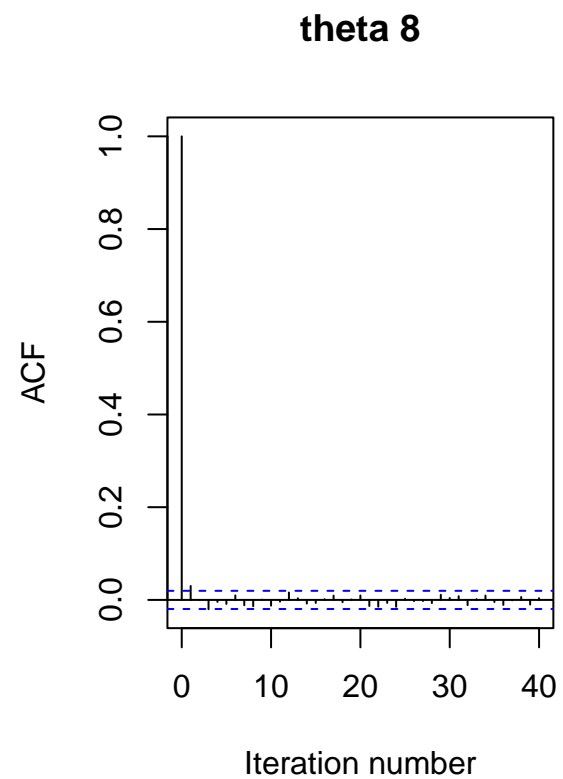
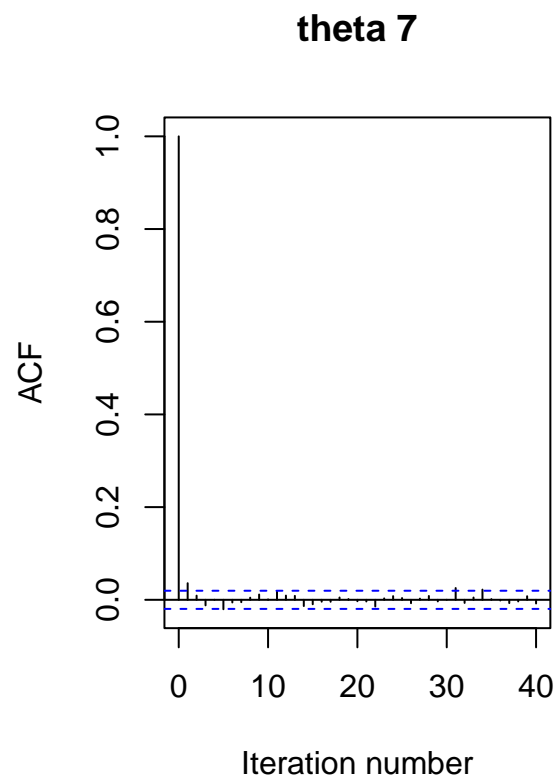
```
for (i in 1:8){
  lb = paste("theta",i)
  acf(theta.MCMC[,i],xlab="Iteration number",main=lb)
}
```



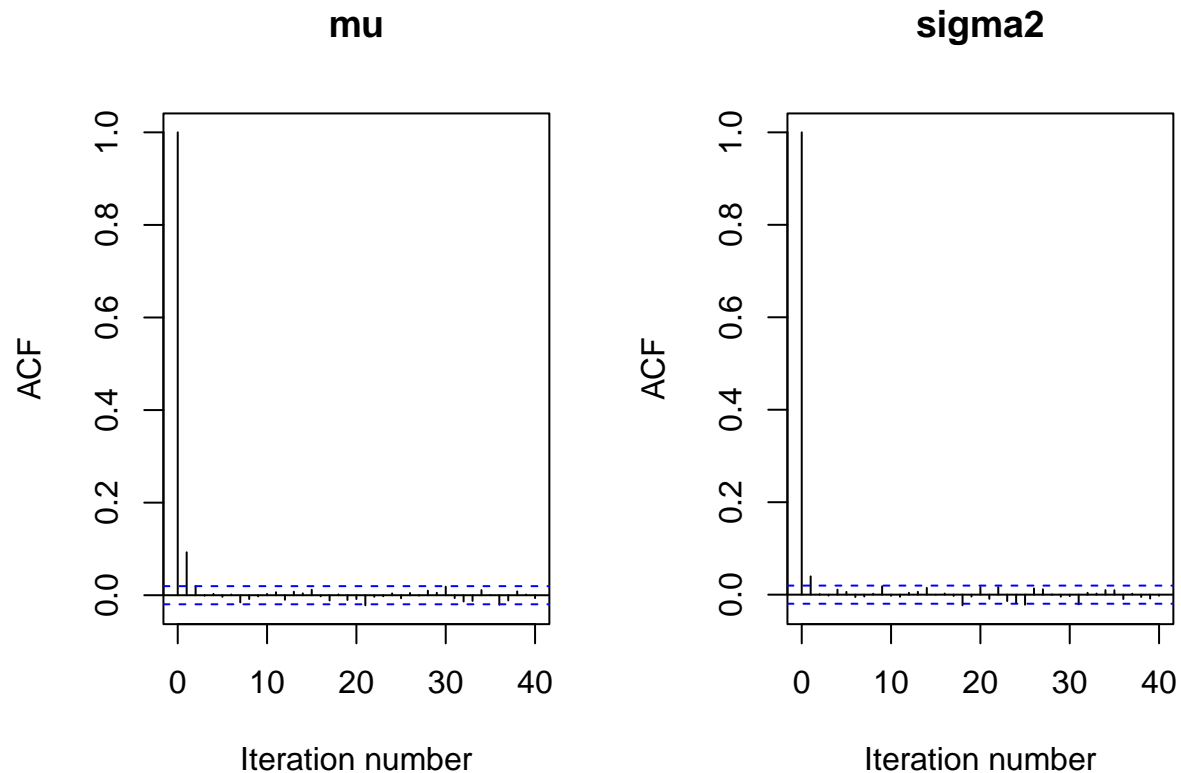








```
# -> uncorrelated after 1-3, use 4 for all theta for convenience
for (i in 1:3){
  lb = colnames(other.pars.MCMC)[i]
  acf(other.pars.MCMC[,i],xlab="Iteration number",main=lb)
}
```



```
# -> uncorrelated after 1-2, use 3 for all other params for convenience
```

```
# thinning & remove burn-in; make all the same for convenience
eff.theta.MCMC <- theta.MCMC[-c(1:1000),][seq(1,9000,4),]
eff.other.pars.MCMC <- other.pars.MCMC[-c(1:1000),][seq(1,9000,3),]
nrow(eff.theta.MCMC)           # 2250, effective sample size for theta 1-8
```

```
## [1] 2250
```

```
nrow(eff.other.pars.MCMC)    # 3000, effective sample size for mu,sigma2,tau2
```

```
## [1] 3000
```

```
# -> all more than 1000
effectiveSize(theta.MCMC)
```

```
##      var1      var2      var3      var4      var5      var6      var7      var8
## 9640.543 9718.172 9479.787 8660.692 7988.488 9092.115 9307.081 9821.197
```

```
effectiveSize(eff.theta.MCMC)
```

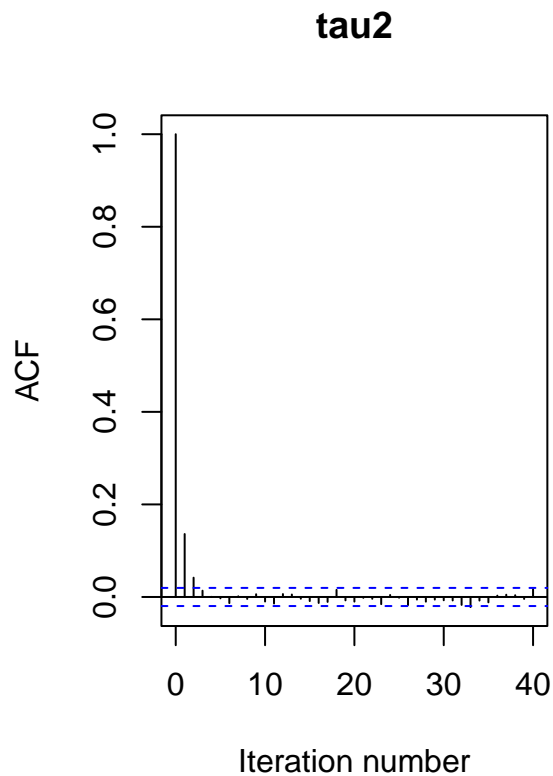
```
##      var1      var2      var3      var4      var5      var6      var7      var8
## 2402.333 2250.000 2250.000 2250.000 2637.777 2250.000 2250.000 2472.529
```

```
effectiveSize(other.pars.MCMC)
```

```
##      mu    sigma2    tau2  
## 8301.723 9239.191 7252.075
```

```
effectiveSize(eff.other.pars.MCMC)
```

```
##      mu    sigma2    tau2  
## 3000.00 3000.00 2800.58
```



**b**

Ans. Prior estimate for Sigma2 was the worst while Mu and tau2 have a pretty close peak comparing to the posterior. The priors were all quite spread out, yet the posterior estimation of all three parameters were have a sharp peak. A much stronger belief was obtained.

```
# Confidence Interval:
```

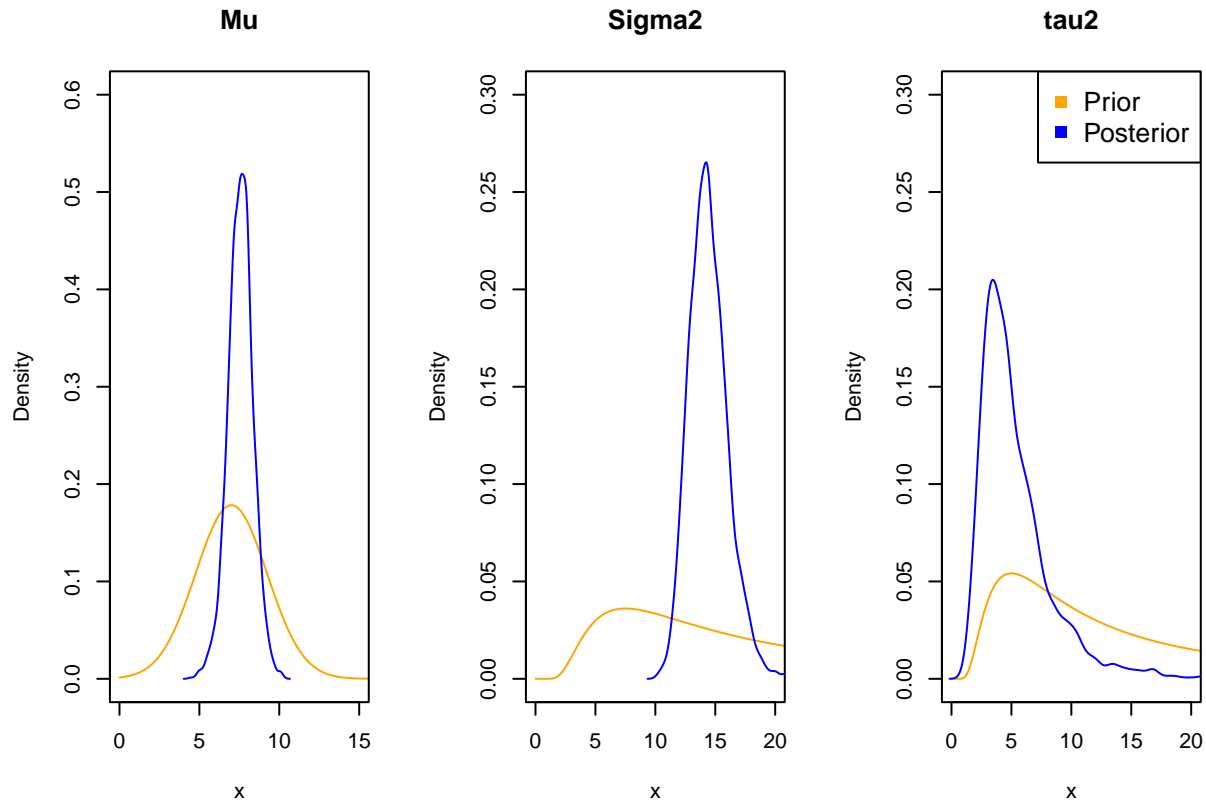
```
apply(eff.other.pars.MCMC,2,function(x) quantile(x,c(0.025,0.5,0.975))) %>% round(3) %>% t()
```

```
##      2.5%    50%   97.5%  
## mu      5.946   7.584   9.121  
## sigma2 11.716 14.312 17.948  
## tau2    1.847   4.550 14.371
```

```

par(mfrow=c(1,3))
# mu:
x = seq(0,20,0.01)
plot(x,dnorm(x,mu.0,sqrt(gamma2.0)), type="l", col="orange", xlim=c(0,15), ylim=c(0,0.6),ylab = "Density",
lines(density(eff.other.pars.MCMC[,1]), col="blue")
# sigma2:
x = seq(0,40,0.01)
plot(x,dinvgamma(x,nu.0/2, nu.0*sigma2.0/2), type="l", col="orange",
      ylim=c(0,0.3), xlim=c(0,20),ylab = "Density",main = "Sigma2")
lines(density(eff.other.pars.MCMC[,2]), col="blue")
# tau2:
x = seq(0,40,0.01)
plot(x,dinvgamma(x,eta.0/2, eta.0*tau2.0/2), type="l", col="orange",
      ylim=c(0,0.3), xlim=c(0,20),ylab = "Density",main = "tau2")
lines(density(eff.other.pars.MCMC[,3]), col="blue")
legend("topright",col=c("orange","blue"), legend = c("Prior", "Posterior"),pch=15, cex=1.2)

```



c

Ans. According to the formula, we know that  $R$  represents the proportion of between-school variance from total variance. The prior belief has a pretty spread guess ( $R = 0.4(0.02 - 0.94)$ ) and didn't provide clear estimation, while the posterior has a clear peak around 0.2 ( $R = 0.24(0.11 - 0.50)$ )

```

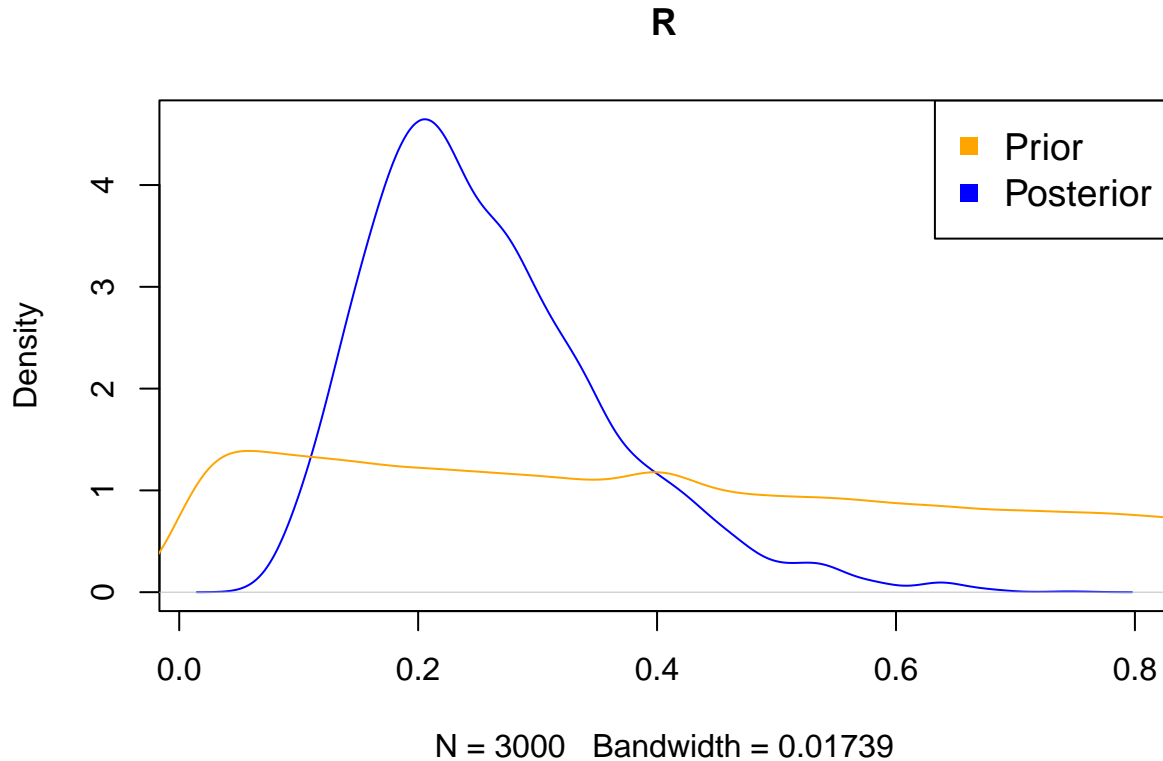
par(mfrow=c(1,1))
set.seed(0)
prior_sigma2 <- rinvgamma(1e5,nu.0/2, nu.0*sigma2.0/2) # sigma2

```

```

set.seed(1)
prior_tau2 <- rinvgamma(1e5,eta.0/2, eta.0*tau2.0/2) # tau2
prior_R <- prior_tau2/(prior_tau2+prior_sigma2)
post_R <- eff.other.pars.MCMC[,3]/(eff.other.pars.MCMC[,3] + eff.other.pars.MCMC[,2])
plot(density(post_R), col="blue", main="R")
lines(density(prior_R), col="orange")
legend("topright",col=c("orange","blue"), legend = c("Prior", "Posterior"),pch=15, cex=1.2)

```



```

# Confidence Interval:
sapply(list(prior=prior_R, posterior=post_R),
        function(x) quantile(x,c(0.025,0.5,0.975))) %>% t()

```

```

##           2.5%      50%      97.5%
## prior      0.0163803 0.4000000 0.9636807
## posterior  0.1114526 0.2397888 0.5034957

```

**d**

The posteriors are shown in the below table

```

ans = rbind(`theta 7 smaller than theta 6`=(eff.theta.MCMC[,7] < eff.theta.MCMC[,6]) %>% mean(),
            `theta 7 is the smallest`=apply(eff.theta.MCMC,1,function(x) which.min(x)==7) %>% mean()
            )
colnames(ans)="Posterior"
ans

```

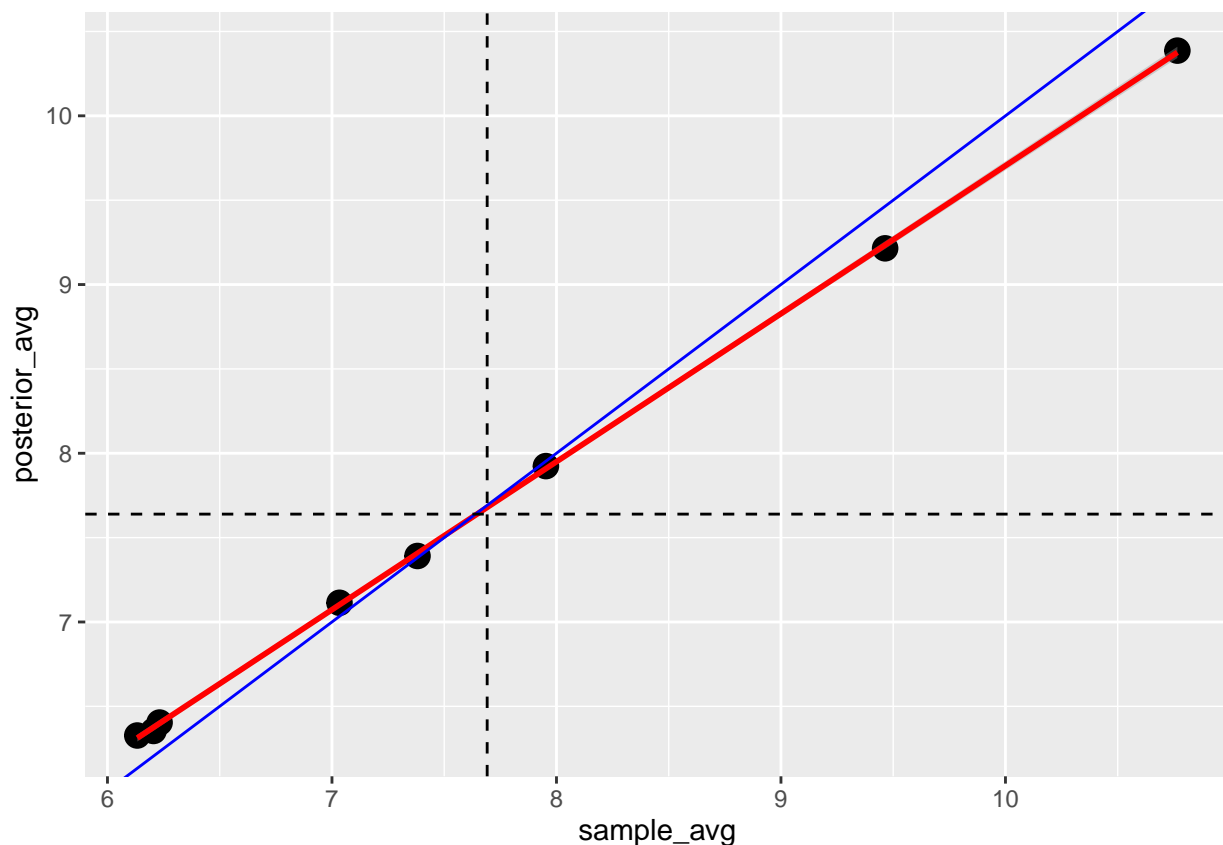
```
##                                     Posterior
## theta 7 smaller than theta 6 0.5182222
## theta 7 is the smallest      0.3244444
```

e

Ans. I fit a linear regression line and add a auxiliary line with slope=1 & intercept=0. The overlapping shows that the values of sample average and posterior average are pretty close. The further the sample mean in each subgroup, the more the posterior will be pulled towards the global mean.

```
ggplot(data.frame(sample_avg=init.theta.j,
                  posterior_avg=apply(eff.theta.MCMC,2,mean)),
       aes(x=sample_avg,y=posterior_avg)) +
  geom_point(size=4) +
  geom_smooth(method = "lm", col="red") +
  geom_abline(slope = 1, intercept = 0, col = "blue") +
  geom_hline(yintercept = mean(eff.theta.MCMC), linetype = "dashed") + # posterior mean
  geom_vline(xintercept = mean(unlist(y)), linetype = "dashed")      # sample mean
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



10.2

a

$$\begin{aligned}
\log\left(\frac{\theta_i}{1-\theta_i}\right) &= \alpha + \beta x_i \\
\frac{\theta_i}{1-\theta_i} &= \exp(\alpha + \beta x_i) \\
\rightarrow \theta_i &= \frac{\exp(\alpha + \beta x_i)}{\exp(\alpha + \beta x_i) + 1} = \Pr(Y_i = 1 | \alpha, \beta, x_i) \\
p(y_i | \alpha, \beta, x_i) &= \text{Bernoulli}\left(\frac{\exp(\alpha + \beta x_i)}{\exp(\alpha + \beta x_i) + 1}\right) \\
\prod p(y_i | \alpha, \beta, x_i) &= \left[\frac{\exp(\alpha + \beta x_i)}{\exp(\alpha + \beta x_i) + 1}\right]^{y_i} \left[1 - \frac{\exp(\alpha + \beta x_i)}{\exp(\alpha + \beta x_i) + 1}\right]^{1-y_i} \\
\text{Let } k_i &= \exp(\alpha + \beta x_i) \\
\frac{k_i^{y_i}}{(k_i + 1)^{y_i}} \left[\frac{(k_i + 1) - k_i}{k_i + 1}\right]^{1-y_i} &= \frac{k_i^{y_i}}{(k_i + 1)^{y_i + (1-y_i)}} = \frac{k_i^{y_i}}{1 + k_i} \\
\Rightarrow \prod p(y_i | \alpha, \beta, x_i) &= \prod \frac{\exp(\alpha y_i + \beta x_i y_i)}{1 + \exp(\alpha + \beta x_i)}
\end{aligned}$$

**b**

I don't have previous assumption of nesting success, and would like to set up a weak prior for the nesting success as  $0.5(0.01 - 0.99)$  in a logistic regression with expected intercept=0

$$\begin{aligned}
&\text{if } \Pr(Y = 1 | \alpha, \beta, x) = 0.5 : \\
&\rightarrow \log\left(\frac{\Pr(Y = 1 | \alpha, \beta, x)}{\Pr(Y = 0 | \alpha, \beta, x)}\right) = 0 \rightarrow \text{Let } \alpha \sim N(0, \sigma_\alpha^2) \\
&\text{if } \Pr(Y = 1 | \alpha, \beta, x) \in (0.01, 0.99) : \\
&\rightarrow \alpha \in \left(\log\left(\frac{0.99}{0.01}\right), \log\left(\frac{0.01}{0.99}\right)\right) = (-4.5951, 4.5951) \\
&\rightarrow \sigma_\alpha \approx 4.5951/1.96 \approx 2.34 \\
&\alpha \sim N(0, 5.49)
\end{aligned}$$

We may set the mean of  $\beta$  by the expected log(odds), and set up a SD to make the range of the log odds also within  $(-4.5951, 4.5951)$  because expected  $\alpha = 0$

$$\begin{aligned}
X \cdot \beta &= 0 \rightarrow \text{Let } \beta \sim (0, \sigma_\beta^2) \\
0 \pm 1.96\sigma_\beta &\in (-4.5951/X, 4.5951/X) \\
\therefore X \in (10, 15) &\rightarrow \sigma_\beta \approx (4.5951/10)/1.96 = 0.2344 \\
\beta &\sim N(0, 0.0549)
\end{aligned}$$

**c**

```

df <- read.table("../HW_Q/msparrownest.dat", col.names = c("success", "wingspan"))
# data info:
y <- df$success
n <- length(y)
p <- 2
X <- matrix(cbind(rep(1,n),df$wingspan),nrow=n,ncol=p)
# Prior
beta0 <- rep(0,p)

```



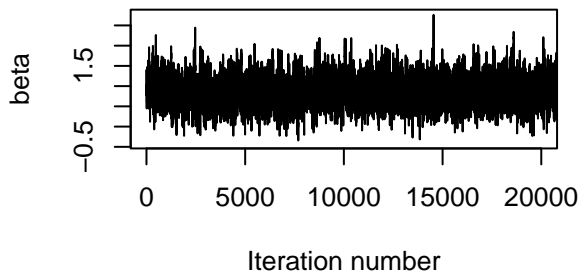
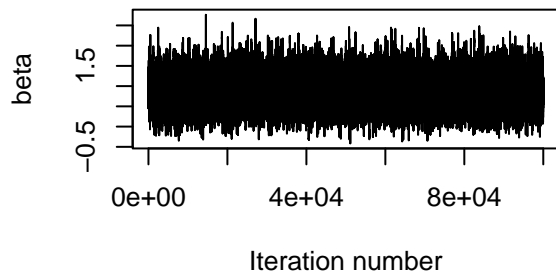
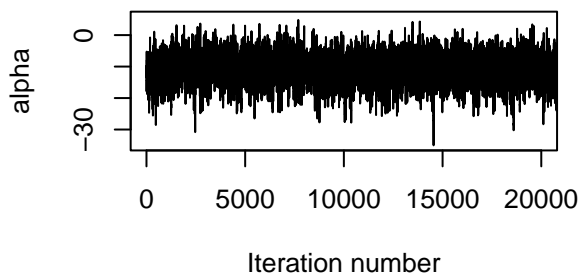
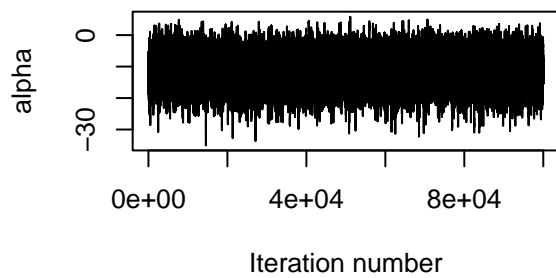
```

Sigma <- matrix(c(5.49,0,0,.0549),2,2)
# initial values:
beta.init <- as.numeric(glm(success~wingspan, family = binomial,df)$coeff)
V <- 8*solve((t(X)%*%X)) # start with V = solve((t(X)%*%X)); modified to get adequate acceptance rate
## V <- k*solve((t(X)%*%X))
# k = 1: 0.78991
# k = 3: 0.6533
# k = 5: 0.57032
# k = 7: 0.50895
# k = 8: 0.48194
# Metropolis algorithm:
S = 1e5
beta.MCMC <- matrix(0,nrow=S,ncol=2)
no.accept <- 0
set.seed(0)
for(k in 1:S){
  if(k==1){
    beta <- beta.init
  }
  out.beta <- sample.beta.metrop(beta,V,y,X,beta0,Sigma)
  new.beta <- out.beta$new.beta
  no.accept <- no.accept+out.beta$accept
  beta <- new.beta
  beta.MCMC[k,] <- beta
}
# acceptance rate:
no.accept/S # 0.48194

## [1] 0.48194

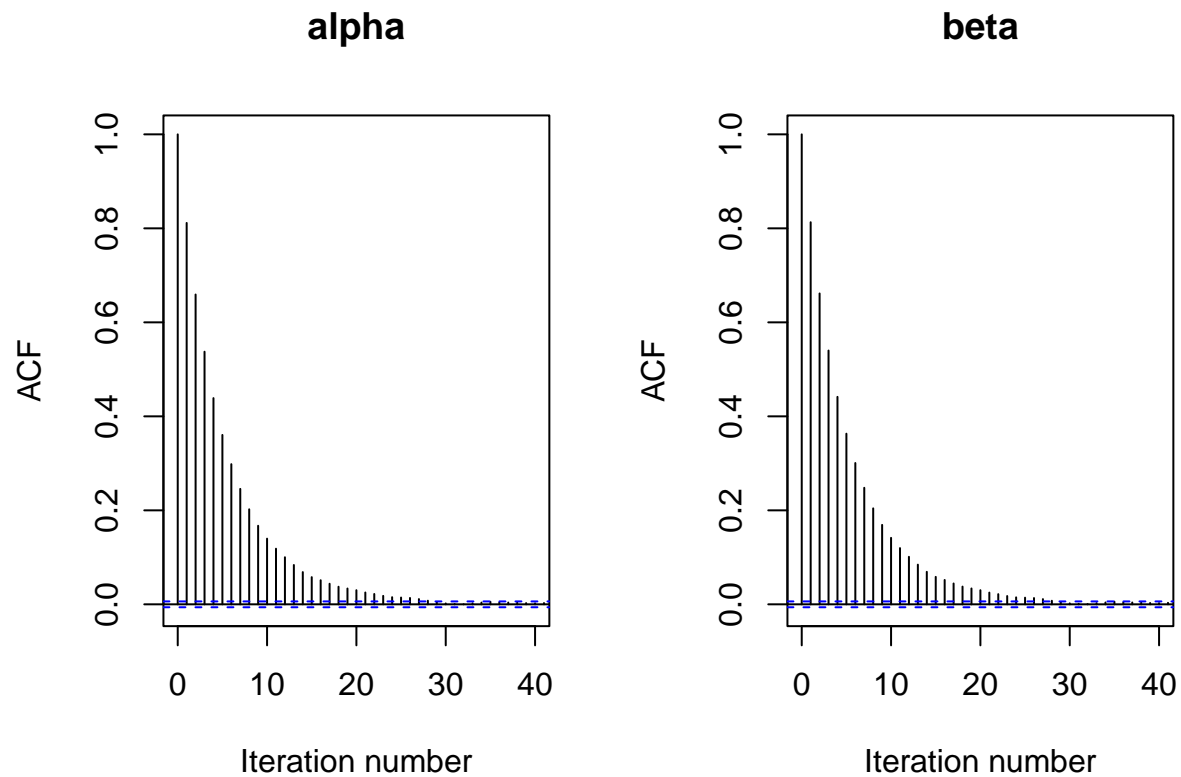
## diagnosis:
# trace plot
par(mfrow=c(2,2))
names(beta.MCMC) <- c("alpha","beta")
for (i in 1:2){
  plot(beta.MCMC[,i],xlab="Iteration number",ylab=names(beta.MCMC)[i],type="l")
  plot(beta.MCMC[,i],xlab="Iteration number",ylab=names(beta.MCMC)[i],type="l", xlim = c(0,2e4)) # car
}

```



```
# -> may have some convergence before 5000
#   use 5000 as burn-in period for all

# auto correlation:
par(mfrow=c(1,2))
for (i in 1:2){
  acf(beta.MCMC[,i],xlab="Iteration number",main=names(beta.MCMC)[i], xlim=c(0,40))
}
```



```
# -> use 30 for thinning
# effective samples:
eff.beta.MCMC <- beta.MCMC[-c(1:5000),][seq(1,95000,30),]
effectiveSize(beta.MCMC)
```

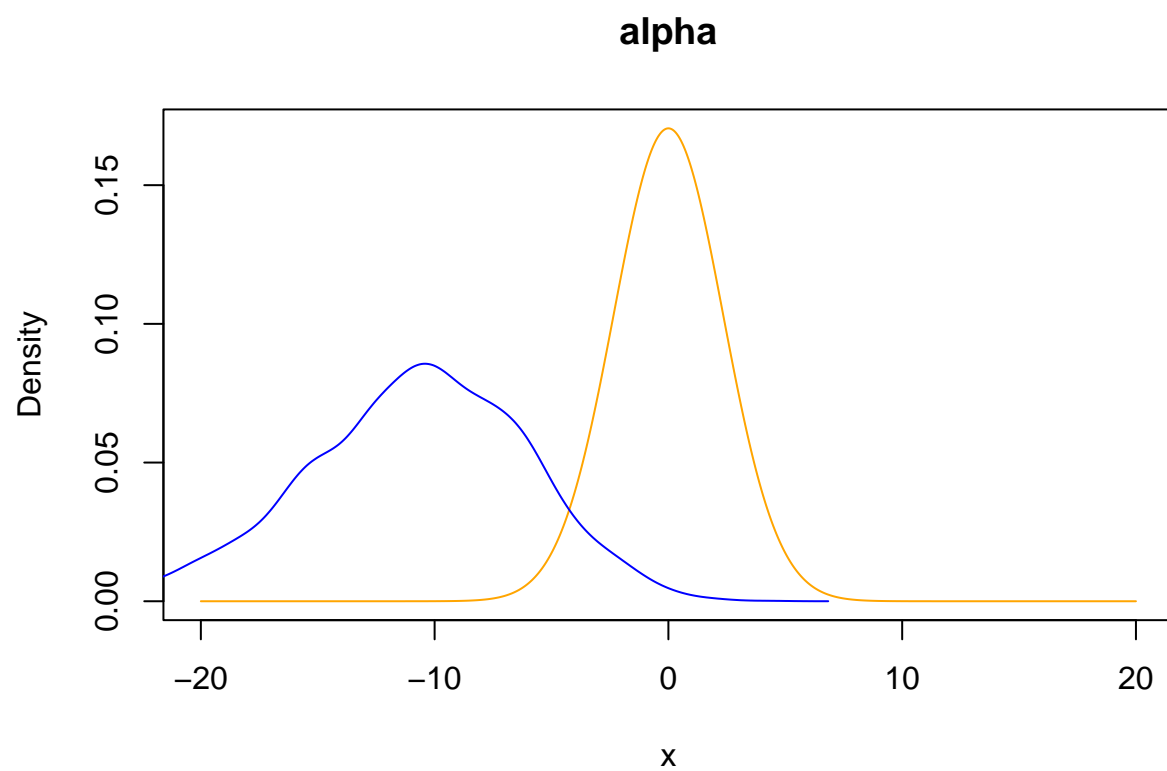
```
##      var1      var2
## 9945.650 9882.345
```

```
effectiveSize(eff.beta.MCMC)
```

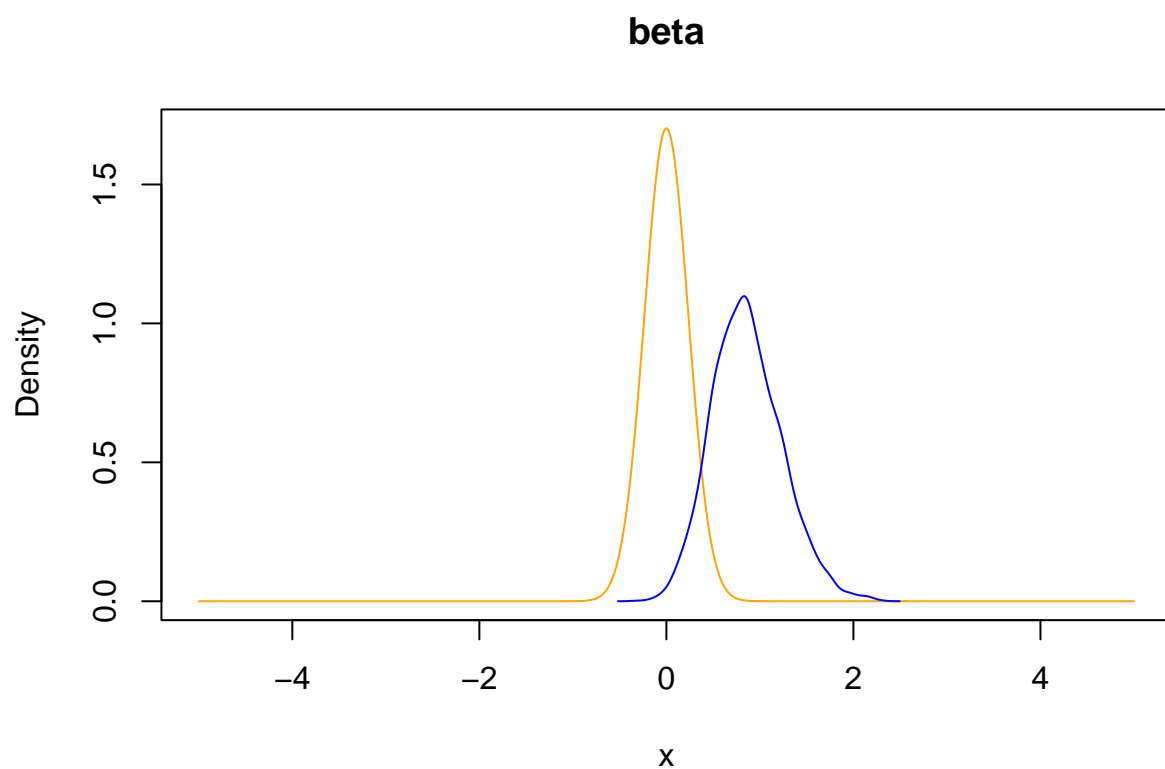
```
##      var1      var2
## 2957.493 2964.641
```

d

```
alpha <- eff.beta.MCMC[,1]
beta <- eff.beta.MCMC[,2]
# alpha:
x = seq(-20,20,0.01)
plot(x,dnorm(x,0,2.34), type="l", col="orange",ylab = "Density",main = "alpha")
lines(density(alpha), col="blue")
```



```
# beta:  
x = seq(-5,5,0.01)  
plot(x,dnorm(x,0,0.2344), type="l", col="orange",ylab = "Density",main = "beta")  
lines(density(beta), col="blue")
```



e

```
x = seq(10,15,by=0.02)
band <- NULL
for( i in 1:length(x) ) {
  band <- rbind(band,quantile( exp(alpha+x[i]*beta)/(1 + exp(alpha+x[i]*beta)), prob=c(.025,.5,.975) ))
}
plot(x,band[,2],type="l",ylim=c(0,1),xlab="wingspan",ylab="f")
lines(x,band[,1],lty=2)
lines(x,band[,3],lty=2)
```

