```
#install.packages("coda")
library(coda)
#Problem 1

#a

#reading data and formatting
school1<-read.table("school1.dat")
school2<-read.table("school2.dat")
school3<-read.table("school3.dat")
school4<-read.table("school4.dat")
school5<-read.table("school5.dat")
school6<-read.table("school6.dat")
school7<-read.table("school7.dat")
school8<-read.table("school8.dat")
school<-list(school1,school2,school3,school4,school5,school6,school7,school8)
for(i in 1:length(school)){
  school[[i]]<-school[[i]][,1]
}

#The code below uses the algorithms provided in Hoff
#prior parameters
mu0<-7 ; g20<-5
t20<-10; eta0<-2
s20<-15; nu0<-2
#

#starting values
m<-length(school)
n<-sv<-ybar<-rep(NA,m)
for(j in 1:m){
  ybar[j]<-mean(school[[j]])
  sv[j]<-var(school[[j]])
  n[j]<-length(school[[j]])
}
theta<-ybar    ; sigma2<-mean(sv)
mu<-mean(theta); tau2<-var(theta)
#

#setup MCMC
set.seed(1)
S<-5000
THETA<-SMT<-NULL
#

#MCMC algorithm
for(s in 1:S){

  #sample new value of the thetas
  for(j in 1:m){
    vtheta<-1/(n[j]/sigma2+1/tau2)
    etheta<-vtheta*(ybar[j]*n[j]/sigma2+mu/tau2)
    theta[j]<-rnorm(1,etheta,sqrt(vtheta))
  }

  #sample a new value of sigma2
  nun<-nu0+sum(n)
  ss<-nu0*s20
  for(j in 1:m){ss<-sigma2+sum((school[[j]]-theta[j])^2)}
  sigma2<-1/rgamma(1,nun/2,ss/2)

  #sample a new value of mu
```

```
  vmu<-1/(m/tau2+1/g20)
  emu<-vmu*(m*mean(theta)/tau2+mu0/g20)
  mu<-rnorm(1,emu,sqrt(vmu))

  #sample a new value of tau2
  etam<-eta0+m
  ss<-eta0*t20+sum((theta-mu)^2)
  tau2<-1/rgamma(1,etam/2,ss/2)

  #store results
  SMT<-rbind(SMT,c(sigma2,mu,tau2))
  THETA<-rbind(THETA,theta)
}
colnames(SMT)<-c("sigma2","mu","tau2")

#check effective sample size
for(i in 1:m){cat(effectiveSize(THETA[,i]),"\n")}
for(i in 1:3){cat(effectiveSize(SMT[,i]),"\n")}

#b
#posterior mean and 95% CI
apply(SMT,2,mean)
apply(SMT,2,function(x) quantile(x,prob=c(0.025,0.975)))

#prior 95% CI
#install.packages("pscl")
library(pscl)
sigma2.quantile.prior<-c(qigamma(0.025,nu0/2,nu0*s20/2),
                         qigamma(0.975,nu0/2,nu0*s20/2))
mu.quantile.prior<-qnorm(c(0.025,0.975),mu0,g20)
tau2.quantile.prior<-c(qigamma(0.025,eta0/2,eta0*t20/2),
                       qigamma(0.975,eta0/2,eta0*t20/2))

#c
R.posterior<-R.prior<-vector(mode = "numeric",length = nrow(SMT))
for(i in 1:length(R.posterior)){

  #Calculate R for a MCMC draws
  R.posterior[i]<-SMT[i,3]/(SMT[i,3]+SMT[i,1])

  #Use Monte Carlo simulation to approximate R
  tau2<-1/rgamma(1,eta0/2,eta0*t20/2)
  sigma2<-1/rgamma(1,nu0/2,nu0*s20/2)
  R.prior[i]<-tau2/(tau2+sigma2)
}

#plot the approximate prior and posterior density of R to compare
plot(density(R.prior),xlab = "R",ylim=c(0,6),main = "Prior and posterior
densities of R")
lines(density(R.posterior),col=90)
legend("topleft",c("prior","posterior"),lty=1,col=c("black",90))
hist(R.prior)
hist(R.posterior)

#d
#The posterior probability that theta7 is smaller than theta theta6
mean(THETA[,7]<THETA[,6])

#The posterior probability that theta7 is the smallest of all theta's
mean(apply(THETA,1,min)==THETA[,7])

#e
```

```r
#sample average
as.numeric(lapply(school,mean))

#posterior expectation of theta
apply(THETA,2,mean)

plot(apply(THETA,2,mean),as.numeric(lapply(school,mean)),xlab=expression(E(the
ta)),ylab = expression(bar(y)))
lines(x=c(5,12),y=c(5,12))

#sample mean of all observations
sum(as.numeric(lapply(school,sum)))/sum(as.numeric(lapply(school,length)))
#posterior expectation of mu
mean(SMT[,2])

#Problem 2
#a
sparrow<-read.table("msparrownest.dat")
logistic<-function(x){1/(1+exp(-x))}
logit<-function(x){log(x/(1-x))}

#c
n<-nrow(sparrow)
y<-sparrow[,1]
x<-cbind(rep(1,n),sparrow[,2])

#proposal distribution
beta<-c(logit(0.9)-(logit(0.9)-logit(0.1))*3,(logit(0.9)-logit(0.1))/5)
var.prop<-6*solve(t(x)%*%x)

#prior parameters
pmn.beta<-c(0,0)
psd.beta<-c(10,10)

S<-10000
acs<-0
BETA<-matrix(0,nrow = S,ncol = 2)
set.seed(1)

for(s in 1:S){
  beta.p<-t(rmvnorm(1,beta,var.prop))

  lhr<-sum(dbinom(y,1,logistic(x%*%beta.p),log = T))+
    sum(dnorm(beta.p,pmn.beta,psd.beta,log = T))-
    sum(dbinom(y,1,logistic(x%*%beta),log = T))-
    sum(dnorm(beta,pmn.beta,psd.beta,log = T))

  if(log(runif(1))<lhr){beta<-beta.p; acs<-acs+1}
  BETA[s,]<-beta
}
effectiveSize(BETA)
acs/S

#d
x<-seq(-30,30,length.out = 601)
#plot alpha
plot(density(BETA[,1]),xlim=c(-30,30),col=90,xlab = expression(alpha),main =
expression(paste("Prior and posterior density of ",alpha)))
lines(x,dnorm(x,0,10))
legend("topright",c("prior","posterior"),lty=1,col=c("black",90))
#plot beta
x<-seq(-20,20,length.out = 601)
```

```
plot(density(BETA[,2]),xlim=c(-20,20),col=90,xlab = expression(alpha),main =
expression(paste("Prior and posterior density of ",beta)))
lines(x,dnorm(x,0,10))
legend("topright",c("prior","posterior"),lty=1,col=c("black",90))

#e
x.seq<-seq(10,15,length.out = 201)
BAND<-NULL
for(x in x.seq){

  BAND<-rbind(BAND,quantile(exp(BETA[,1]+x*BETA[,2])/(1+exp(BETA[,1]+x*BETA[,2
  ])), prob=c(0.025, 0.5, 0.975)))
}
plot(x.seq,BAND[,2],type="l",ylim=c(0,1),xlab="x",ylab="f")
lines(x.seq,BAND[,1],lty=2)
lines(x.seq,BAND[,3],lty=2)

#Problem 4
tplant<-read.table("tplant.dat")
colnames(tplant)<-c("height","time","ph")
attach(tplant)
library(mvtnorm)

#a
summary(lm(height~time*ph))#interaction terem not significant
tplant.lm<-lm(height~time+ph)
coefficients(tplant.lm)
x<-cbind(rep(1,length(height)),time,ph)
solve(t(x)%*%x)%*%t(x)%*%height #ols

#b
plot(residuals(tplant.lm),ylab = "residuals")
for(i in 1:9){lines(x=c(2*i+0.5,2*i+0.5),y=c(-2,2),col=91)}

#c

#prior parameters
n<-length(height)
lmfit<-lm(height~-1+x)
nu0<-1
s20<-1
T0<-diag(1/1000,nrow=3)

#The function to make a new C_{\rho}
updateCor<-function(r){
  C.rho<-matrix(data = rep(0,400),ncol = 20)
  odd<-T
  for(i in 1:20){
    C.rho[i,i]<-1;
    if(odd){
      C.rho[i,i+1]<-r
      odd<-F
    }else{
        C.rho[i,i-1]<-r
        odd<-T}
  }
  C.rho
}

#MCMC
set.seed(1)
#starting values
```

```
beta<-lmfit$coef
s2<-summary(lmfit)$sigma^2
rho<-0.8
S<-30000 ; odens<-S/1000 #thinning
OUT<-NULL ; ac<-0
for(s in 1:S)
{
  #update beta
  Cor<-updateCor(rho)  ; iCor<-solve(Cor)
  V.beta<- solve(t(x)%*%iCor%*%x/s2 + T0)
  E.beta<- V.beta%*%(t(x)%*%iCor%*%height/s2)
  beta<-t(rmvnorm(1,E.beta,V.beta))

  #update simga

  s2<-1/rgamma(1,(nu0+n)/2,(nu0*s20+t(height-x%*%beta)%*%iCor%*%(height-x%*%be
  ta))/2)

  #update rho using Metropolis algorithm
  rho.p<-runif(1,rho-0.25,rho+0.25)
  if(rho.p<0){rho.p<-abs(rho.p)}else{ if(rho.p>1){rho.p<-2-rho.p}}
  Cor.p<-updateCor(rho.p)
  #log accepting ratio(assuming uniform prior for rho)
  lr<-dmvnorm(height,mean = x%*%beta,sigma=s2*Cor.p,log = T)-
    dmvnorm(height,mean = x%*%beta,sigma=s2*Cor,log = T)

  if(log(runif(1)) < lr) { rho<-rho.p ; ac<-ac+1 }

  if(s%%odens==0){
    OUT<-rbind(OUT,c(beta,s2,rho))
  }
}
library(coda)
apply(OUT,2,effectiveSize)

#d

#compare the mean of estimated parameters
apply(OUT,2,mean)
coef(lmfit)
summary(lmfit)$sigma^2

#compare standard error of coefficient
apply(OUT,2,sd)[1:3]
summary(lmfit)$coefficient[,"Std. Error"]

mean(OUT[,5])
quantile(OUT[,5],c(0.25,0.95))
```