STAT7016 Assignment 4

Rui Qiu

2017-10-08

Setting

```
library(LearnBayes)
library(coda)
library(MASS)
library(MCMCpack)
library(mvtnorm)

set.seed(7016)
```

Problem 1

The files `school1.dat` through `school8.dat` give weekly hours spent on homework for students sampled from eight different schools. We want to obtain posterior distributions for the true means for the eight different schools using a hierarchical normal model with the following prior parameters:

$$\mu_0 = 7, \gamma_0^2 = 5, \tau_0^2 = 10, \eta_0 = 2, \sigma_0^2 = 15, \nu_0 = 2$$

(a)

Run a Gibbs sampling algorithm to approximate the posterior distribution of $\{\theta, \sigma^2, \mu, \tau^2\}$. Assess the convergence of the Markov chain, and find the effective sample size for $\{\sigma^2, \mu, \tau^2\}$. Run the chain long enough so that the effective sample sizes are all above $1,000$.

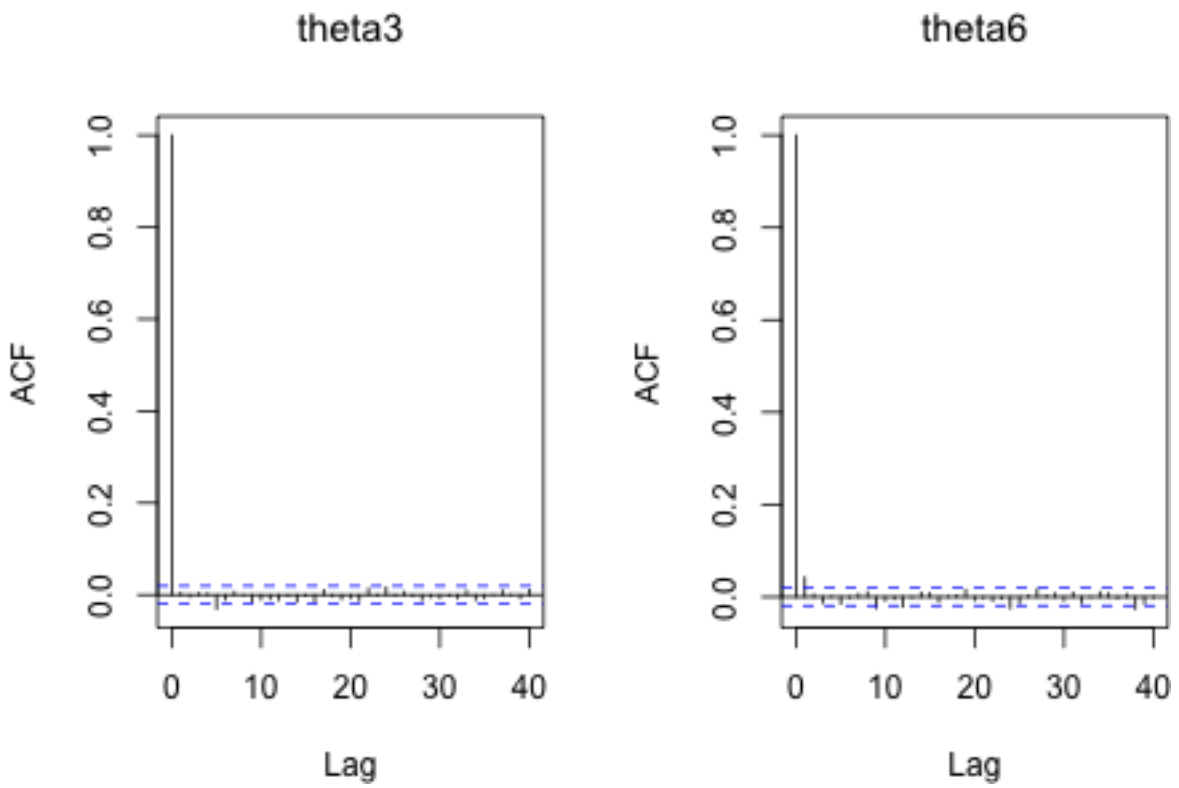
Solution:

```
hw <- NULL
for (i in 1:8) {
    aschool <- read.table(paste0('school', i, '.dat'),header=F)
    aschool <- cbind(rep(i,length(aschool)),aschool)
    hw <- rbind(hw, aschool)
}
colnames(hw) <- c('school','hours')

# priors
mu0 <- 7; gamma20 <- 5; tau20 <- 10; eta0 <- 2; sigma20 <- 15; nu0 <- 2

# initial values (use sample means and sample variances)
n <- rep(NA,8)
ybar <- rep(NA,8)
svar <- rep(NA,8)
for (i in 1:8) {
    Yi <- hw[hw[,1]==i,2]
```

```
    n[i] <- length(Yi)
    ybar[i] <- mean(Yi)
    svar[i] <- var(Yi)
}

# initial theta estimates
theta <- ybar
sigma2 <- mean(svar)
mu <- mean(theta)
tau2 <- var(theta)

# Gibbs
K <- 10000
paras <- matrix(nrow=K,ncol=8+3)
colnames(paras) <- c("theta1","theta2","theta3","theta4","theta5",
                     "theta6","theta7","theta8","sigma2","mu","tau2")
for (i in 1:K) {
    # sample thetas
    for (j in 1:8) {
        v <- 1/(n[j]/sigma2+1/tau2)
        e <- v*(ybar[j]*n[j]/sigma2+mu/tau2)
        theta[j] <- rnorm(1,e,sqrt(v))
    }

    # sample sigma2
    nun <- nu0 + sum(n)
    ss <- nu0*sigma20
    for (j in 1:8) {
        ss <- ss+sum((hw[hw[,1]==j,2]-theta[j])^2)
    }
    sigma2 <- 1/rgamma(1,nun/2,ss/2)

    # sample mu
    v <- 1/(8/tau2+1/gamma20)
    e <- v*(8*mean(theta)/tau2+mu0/gamma20)
    mu <- rnorm(1,e,sqrt(v))

    # sample tau2
    eta8 <- eta0+8
    ss <- eta0*tau20+sum((theta-mu)^2)
    tau2 <- 1/rgamma(1,eta8/2,ss/2)

    paras[i,1:8] <- theta
    paras[i,9:11] <- c(sigma2,mu,tau2)
}
post.sigma2 <- paras[,9]
post.mu <- paras[,10]
post.tau2 <- paras[,11]
```

Calculate the effective sample sizes, also plot the acf below (we only picked 2 acfs)

```
par(mfrow=c(1,2))
acf(paras[,3],main="theta3")
acf(paras[,6],main="theta6")
```




```
par(mfrow=c(1,1))
ef <- cbind(effectiveSize(post.sigma2),effectiveSize(post.mu),
            effectiveSize(post.tau2))
colnames(ef) <- c("sigma2","mu","tau2")
rownames(ef) <- "effective size"
ef
```

```
##                  sigma2       mu     tau2
## effective size 8932.689 8174.218 6876.188
```

all greater than 1000. good enough.

(b)

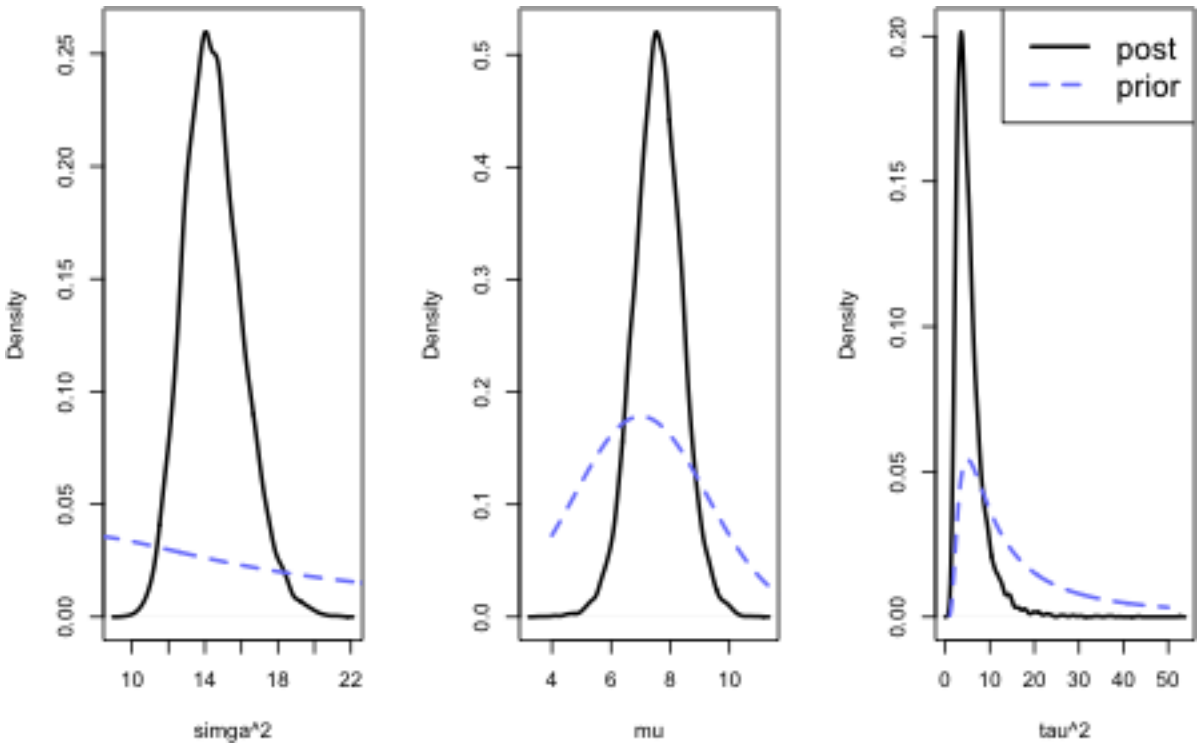
Compute posterior means and 95% confidence regions for $\{\sigma^2, \mu, \tau^2\}$. Also, compare the posterior densities to the prior densities, and discuss what was learned from the data.

```
post.paras <- rbind(quantile(post.sigma2,prob=c(0.025,0.5,0.975)),
                    quantile(post.mu,prob=c(0.025,0.5,0.975)),
                    quantile(post.tau2,prob=c(0.025,0.5,0.975)))
rownames(post.paras) <- c("sigma2","mu","tau2")
post.paras
```

```
##             2.5%       50%     97.5%
## sigma2 11.700823 14.354105 17.920654
## mu      5.958100  7.578993  9.173253
## tau2    1.918046  4.643076 14.336684
```

```
par(mfrow=c(1,3))
plot(density(post.sigma2),xlab="simga^2",main="",lwd=2) # post sigma2
```

```
lines(seq(0,25,0.1),dinvgamma(seq(0,25,0.1),nu0/2,nu0*sigma20/2),lwd=2,
      lty=2,col="#7B84FC") # prior sigma2
plot(density(post.mu),xlab="mu",main="",lwd=2) # post mu
lines(seq(4,15,0.1),dnorm(seq(4,15,0.1),mu0,sqrt(gamma20)),lwd=2,
      lty=2,col="#7B84FC") # prior mu
plot(density(post.tau2),xlab="tau^2",main="",lwd=2) # post tau2
lines(seq(1,50,0.1),dinvgamma(seq(1,50,0.1),eta0/2,eta0*tau20/2),lwd=2,
      lty=2,col="#7B84FC") # prior tau2
legend("topright",legend=c("post","prior"),col=c("black","#7B84FC"),
       lty=c(1,2),lwd=c(2,2),cex=1.5)
```




```
par(mfrow=c(1,1))
```

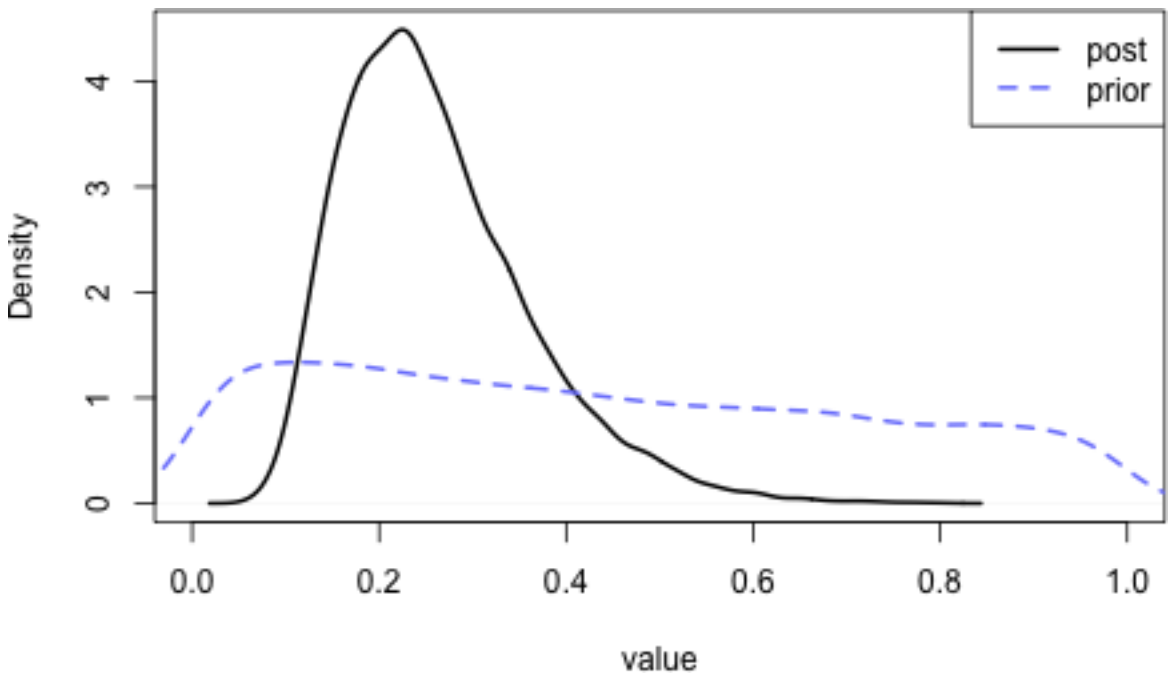
prior estimates for μ and τ^2 have similar shape, but σ^2 varies After this analysis, we have estimates for mu, the average amount of hours of schoolwork spent at a typical school, tau2, the variability between schools in the average hours of schoolwork, and sigma2, the variability among students' hours in each school.

(c)

Plot the posterior density for $R = \frac{\tau^2}{\sigma^2+\tau^2}$ and compare it to a plot of the prior density on R. Describe the evidence for between-school variation.

```
# random sample 10000 pairs of tau20 and sigma20
prior.sims <- 10000
prior.tau20 <- (1/rgamma(prior.sims,eta0/2,eta0*tau20/2))
prior.sigma20 <- (1/rgamma(prior.sims,nu0/2,nu0*sigma20/2))
plot(density(post.tau2/(post.tau2+post.sigma2)),main="",xlab="value",
       lwd=2,lty=1,xlim=c(0,1))
lines(density(prior.tau20/(prior.tau20+prior.sigma20)),col="#7B84FC",lwd=2,lty=2)
legend("topright",lty=c(1,2),lwd=c(2,2),col=c("black","#7B84FC"),
```

```
       legend=c("post","prior"))
```




```
mean(prior.tau20/(prior.tau20+prior.sigma20))
```

```
## [1] 0.4311038
```

```
mean(post.tau2/(post.tau2+post.sigma2))
```

```
## [1] 0.2612869
```

RR measures how much of the total variance in our data is between-group. Our prior didn't contain much information about this quantity, but after inference, we expect that around 25% of our variance comes from between group variance (tau2).

(d)

Obtain the posterior probability that θ_7 is smaller than θ_6, as well as the posterior probability that θ_7 is the smallest of all the θ's.

```
mean(paras[,7]<paras[,6])
```

```
## [1] 0.5235
```

```
theta7.smallest <- 0
for (i in 1:K) {
    if (sort(paras[i,1:8],decreasing=F)[1]==paras[i,7]) {
        theta7.smallest <- theta7.smallest+1
    }
}
theta7.smallest/K
```

```
## [1] 0.3207
```

$$\Pr(\theta_7 < \theta_6 \mid \mathbf{y}) =$$
$$\Pr(\theta_7 \text{ is the smallest among all } \theta\text{s} \mid \mathbf{y}) =$$

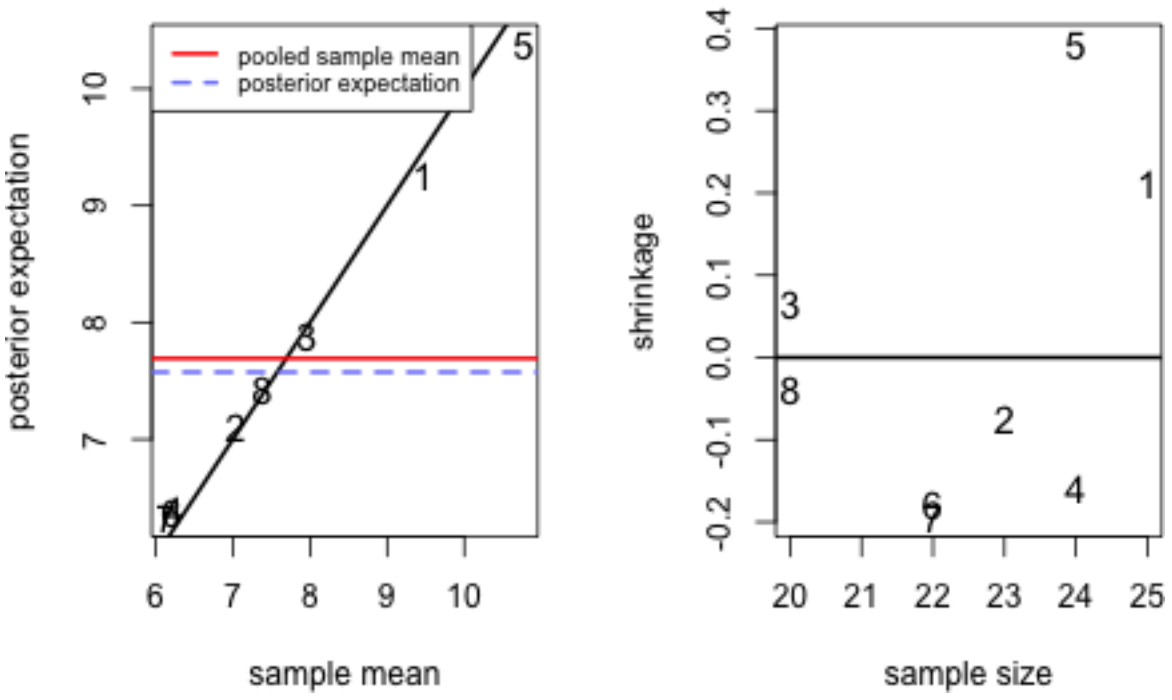
(e)

Plot the sample averages $\bar{y}_1, \ldots, \bar{y}_8$ against the posterior expectations of $\theta_1, \ldots, \theta_8$, and describe the relationship. Also compute the sample mean of all observations and compare it to the posterior mean of μ. Estimate the shrinkage effect for each school.

```
comparison <- data.frame(school=1:8,
                         sample.mean=ybar,
                         post.expectation=colMeans(paras[,1:8]))
comparison
```

```
##        school sample.mean post.expectation
## theta1      1    9.464000         9.252736
## theta2      2    7.033478         7.108896
## theta3      3    7.953000         7.888101
## theta4      4    6.232083         6.390356
## theta5      5   10.765833        10.384387
## theta6      6    6.205000         6.384127
## theta7      7    6.132727         6.327621
## theta8      8    7.381000         7.419471
```

```
par(mfrow=c(1,2))
plot(comparison$sample.mean,comparison$post.expectation,
     xlab="sample mean",ylab="posterior expectation",
     pch=paste0("",comparison$school),cex=1.2)
abline(0,1,lwd=2)
abline(h=mean(hw[,"hours"]),lty=1,col="red",lwd=2)
abline(h=mean(post.mu),lty=2,col="#7B84FC",lwd=2)
legend("topleft",legend=c("pooled sample mean","posterior expectation"),
       lty=c(1,2),col=c("red","#7B84FC"),lwd=c(2,2),cex=0.75)
plot(as.numeric(table(hw[,"school"])),
     comparison$sample.mean-comparison$post.expectation,
     pch=paste0("",comparison$school),ylab="shrinkage",xlab="sample size",
     cex=1.2)
abline(h=0,lty=1,lwd=2)
```




```
par(mfrow=c(1,1))
c(mean(hw[,"hours"]), mean(post.mu)) # sample mean of all obs vs post mean of mu
```

```
## [1] 7.691278 7.574251
```

There is a quite tight correspondence between the sample average and the posterior expectation, although mild shrinkage can be observed with schools with very high and low sample averages being pulled towards the mean.

> The relationship between sample size and the amount of variance shrinkage is shown in the second panel of the plot. As with estimation for the group means, the larger amounts of shrinkage generally occur for groups with smaller sample sizes.

it gets pulled toward . . .

> This makes sense: The larger the sample size for a group, the more information we have for that group and the less information we need to "borrow" from the rest of the population.

Problem 2

Younger male sparrows may or may not nest during a mating season, perhaps depending on their physical characteristics. Researchers have recorded the nesting success of 43 young male sparrows of the same age, as well as their wingspan, and the data appear in the file `msparrownest.dat`. Let Y_i be the binary indicator that sparrow i successfully nests, and let x_i denote their wingspan. Our model for Y_i is $\text{logit}[\Pr(Y_i = 1 \mid \alpha, \beta, x_i)] = \alpha + \beta x_i$, where the logit function is given by $\text{logit}\,[\theta] = \log\,[\theta/(1-\theta)]$.

(a)

Write out the joint sampling distribution $\prod_{i=1}^{n} p(y_i \mid \alpha, \beta, x_i)$ and simplify as much as possible.

Solution:

Let $\theta_i = P(Y_i = 1 \mid \alpha, \beta, x_i)$, then we can solve for θ_i:

$$\begin{aligned}
\log\left(\frac{\theta_i}{1-\theta_i}\right) &= \alpha + \beta x_i \\
\frac{\theta_i}{1-\theta_i} &= e^{\alpha+\beta x_i} \\
\theta_i &= \frac{\exp(\alpha+\beta x_i)}{1+\exp(\alpha+\beta x_i)}
\end{aligned}$$

We can treat the fraction expression as a probability $p_i = \frac{\exp\,(\alpha+\beta x_i)}{1+\exp(\alpha+\beta x_i)}$ so that $Y_i \sim \text{Bernoulli}(p_i)$. Then we can write

$$p(y_i \mid \alpha, \beta, x_i) = p_i^{y_i}(1-p_i)^{1-y_i}$$

Suppose again $\eta_i = \exp\,(\alpha + \beta x_i)$, then the required joint sampling distribution can be rewritten as:

$$\begin{aligned}
\prod_{i=1}^{n} p(y_i \mid \alpha, \beta, x_i) &= \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i} \\
&= \prod_{i=1}^{n} \left(\frac{\eta_i}{1+\eta_i}\right)^{y_i} \left(1 - \frac{\eta_i}{1+\eta_i}\right)^{1-y_i} \\
&= \prod_{i=1}^{n} \left(\frac{\eta_i}{1+\eta_i}\right)^{y_i} \left(\frac{1}{1+\eta_i}\right)^{1-y_i} \\
&= \prod_{i=1}^{n} \frac{\eta_i^{y_i}}{(1+\eta_i)} \\
&= \prod_{i=1}^{n} \frac{\exp\, y_i(\alpha+\beta x_i)}{1+\exp\,(\alpha+\beta x_i)}
\end{aligned}$$

(b)

Formulate a prior probability distribution over α and β by considering the range of $\Pr(Y = 1 \mid \alpha, \beta, x)$ as x ranges over 10 to 15, the approximate range of the observed wingspans.

Solution:

Consider the model in logit form, as Gelman says in his book, "TAKE LOG(it)!" If we want to have an unit information prior about intercept, we should center our parameter of wingspan, i.e. β, around 0. Similarly, if we want to have an unit information prior about the coefficient of wingspan, then we should center our parameter of intercept, i.e. α, around 0.

The question is what to use for a prior distribution and how diffuse to make our priors. We would like priors for α and β to be symmetric, so normal distributions for both make sense. And we want our prior to be uninformative, so we should set the variance of these normals high.

If α is always 0, then as x moves from 10 to 15, we want our possible values of β to allow for a change in the log-odds ratio from approximately 0 to 1. Notice the log-odds ratio of some sufficiently small number e.g. 1e-5 is -11.5129155 which is roughly 10. Since x at a minimum is 10, it makes sense to have most of our prior on beta in the range $(-10/10, 10/10) = (-1, 1)$. So I'll set the standard deviation of the β prior to 0.5 and the variance to 0.25.

Similarly I'll let the standard deviation of our α prior to be 5 and the variance 25, so that if $\beta = 0$, the most of the α prior falls in the logit interval $(-10, 10)$.

(c)

Implement a Metropolis algorithm that approximates $p(\alpha, \beta \mid \mathbf{y}, \mathbf{x})$. Adjust the proposal distribution to achieve a reasonable acceptance rate, and run the algorithm long enough so that the effective smaple size is at least 1,000 for each parameter.

```
sparrow <- read.table("msparrownest.dat",header=F)
K <- 10000
y <- sparrow[,1]
n <- length(y)

ALPHA <- numeric(K)
BETA <- numeric(K)
accept.cts <- 0

x <- cbind(rep(1,n),sparrow[,2]) # wingspans

var.prop  <- 7*solve(t(x)%*%x) # proposal
prior.mean.theta <- c(0,0) # prior parameters
prior.sd.theta <- sqrt(c(25,0.25))

theta <- c(0,0) # initial values
log.p.y <- function(x,y,theta){
    p <- exp(x%*%theta)/(1+exp(x%*%theta))
    sum(dbinom(y,1,p,log=T))
}

p.theta <- function(theta){
    sum(dnorm(theta,prior.mean.theta,prior.sd.theta,log=T))
}

for (i in 1:K) {
    theta.star <- mvrnorm(1,theta,var.prop)
    if (log(runif(1))
        <(log.p.y(x,y,theta.star)+p.theta(theta.star))
```

```
        -(log.p.y(x,y,theta)+p.theta(theta))) {
        theta <- theta.star
        accept.cts <- accept.cts+1
    }
    ALPHA[i] <- theta[1]
    BETA[i] <- theta[2]
}
accept.cts/K
```

```
## [1] 0.3986
```

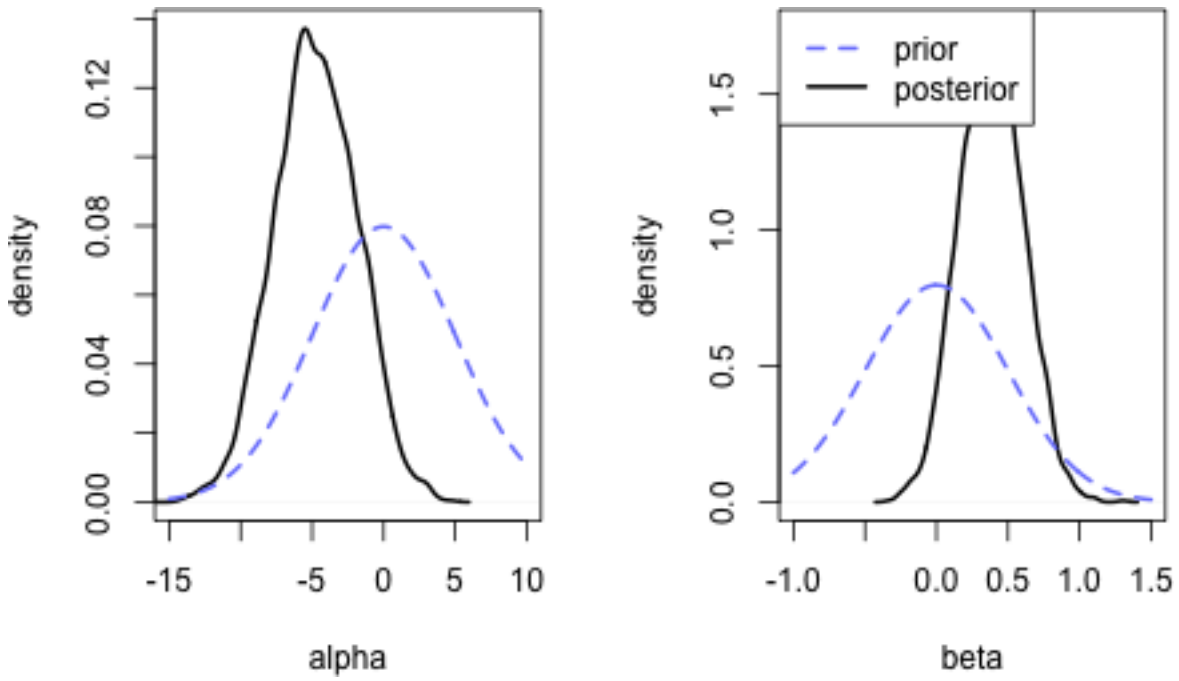
```
c(effectiveSize(ALPHA),effectiveSize(BETA))
```

```
##     var1     var1
## 1462.229 1451.637
```

(d)

Compare the posterior densities for α and β to their prior densities.

```
par(mfrow=c(1,2))
# alpha prior
plot(density(ALPHA),type="l",xlab="alpha",ylab="density",
     main="",xlim=c(-15,10),lwd=2)
lines(seq(-15,10,0.1),dnorm(seq(-15,10,0.1),0,5),lty=2,col="#7B84FC",lwd=2)
plot(density(BETA),type="l",xlab="beta",ylab="density",
     main="",xlim=c(-1,1.5),lwd=2)
lines(seq(-1,1.5,0.01),dnorm(seq(-1,1.5,0.01),0,0.5),lty=2,col="#7B84FC",lwd=2)
legend("topleft",
       legend=c("prior","posterior"),lty=c(2,1),
       col=c("#7B84FC","black"),lwd=c(2,2))
```



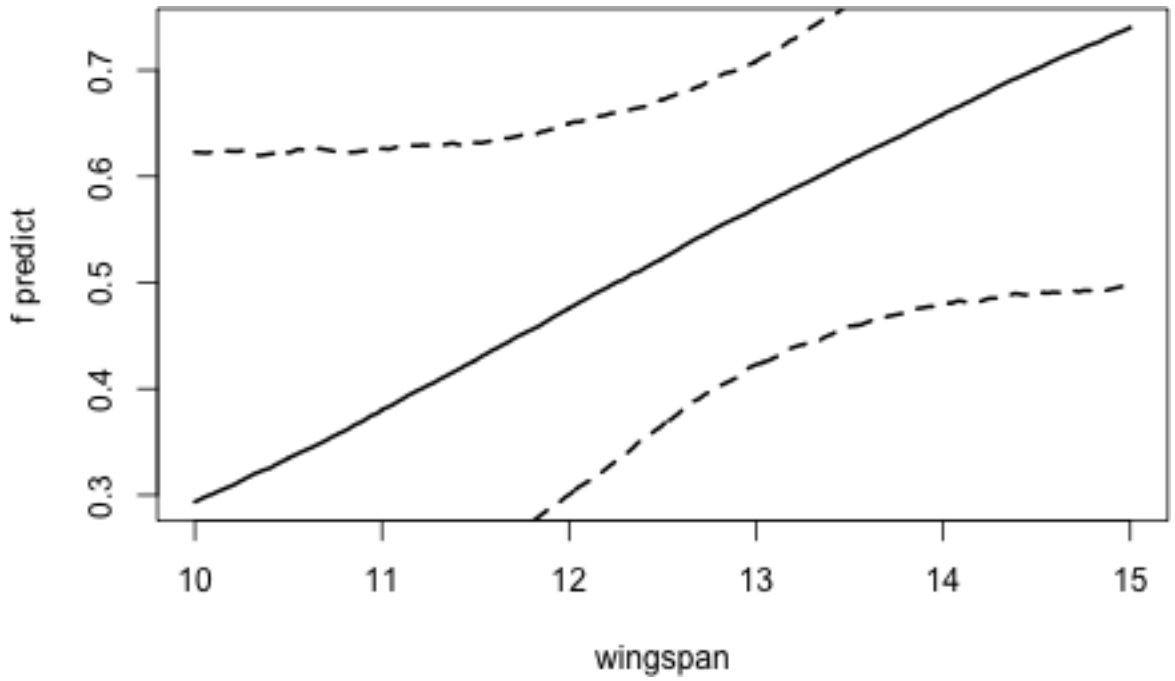
(e)

Using output from the Metropolis algorithm, come up with a way to make a confidence band for the following function $f_{\alpha,\beta}(x)$ of wingspan:

$$f_{\alpha,\beta}(x) = \frac{\exp^{\alpha+\beta x}}{1+\exp^{\alpha+\beta x}}$$

where α and β are the parameters in your sampling model. Make a plot of such a band.

```
span <- seq(10,15,0.005)
qtiles <- sapply(span,function(x){
    quantile(exp(ALPHA+BETA*x)/(1+exp(ALPHA+BETA*x)),probs=c(0.025,0.5,0.975))
})
plot(span,qtiles[2,],xlab="wingspan",type="l",ylab="f predict",main="",
     xlim=c(min(span),max(span)),
     ylim=c(min(qtiles[2,]),max(qtiles[2,])),lwd=2)
lines(span,qtiles[1,],lwd=2,lty=2)
lines(span,qtiles[3,],lwd=2,lty=2)
```



This is the logistic regressions' approximate probability of nesting based on wingspan, although the confidence intervals are quite wide.

Problem 3

The file `tplant.dat` contains data on the heights of ten tomato plants, grown under a variety of soil pH conditions. Each plant was measured twice. During the first measurement, each plant's height was recorded and a reading of pH soil was taken. During the second measurement only plant height was measured, although it is assumed that pH levels did not vary much from measurement to measurement.

(a)

Using ordinary least squares, fit a linear regression to the data, modelling plant height as a function of time (measurement period) and pH level. Interpret your model parameters.

```
tplant <- read.table("tplant.dat",header=F)
y <- tplant[,1]
x <- cbind(tplant[,2],tplant[,3])
fit <- lm(y~x)
summary(fit)
```

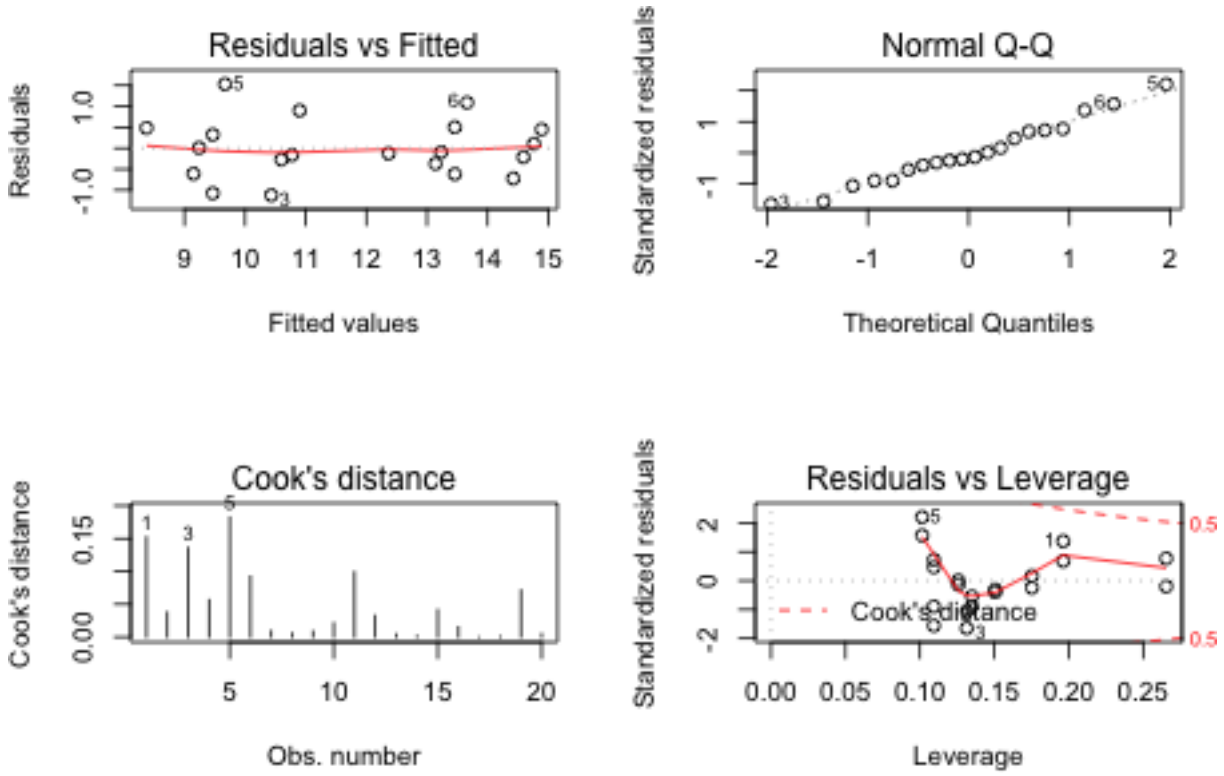
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1225 -0.4307 -0.1101  0.4574  1.5301
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.2087     0.5891   12.24 7.45e-10 ***
## x1            3.9910     0.3280   12.17 8.14e-10 ***
## x2            0.5778     0.1204    4.80 0.000167 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7334 on 17 degrees of freedom
## Multiple R-squared:  0.9096, Adjusted R-squared:  0.899
## F-statistic: 85.55 on 2 and 17 DF,  p-value: 1.339e-09
```

mean intercept 7.2087, x2 the coefficient before pH value, x1 the height increased from t0 to t1

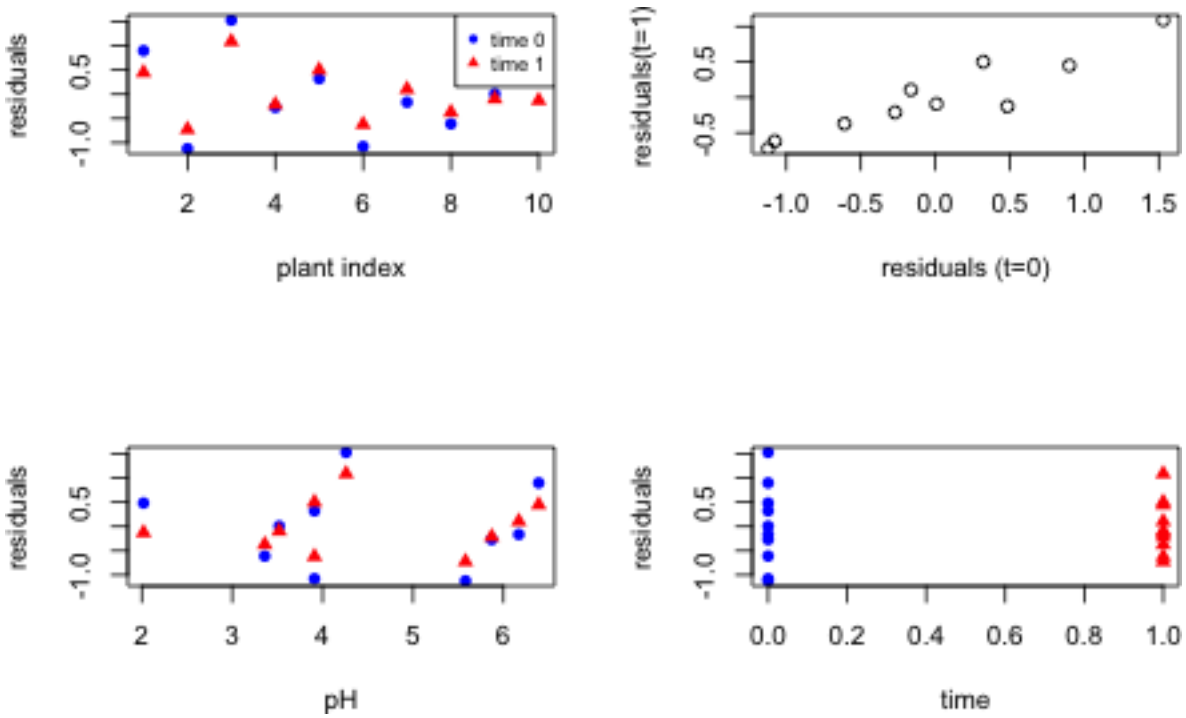
(b)

Perform model diagnostics. In particular, carefully analyse the residuals and comment on possible violations of assumptions. In particular, assess (graphically or otherwise) whether or not the residuals within a plant are independent. What parts of your ordinary linear regression model do you think are sensitive to any violations of assumptions you may have detected?

```
par(mfrow=c(2,2))
# 4 basic diagnostic plot
plot(fit,which=c(1,2,4,5))
```




```
res <- fit$residuals
res0 <- res[which(x[,1]==0)]
res1 <- res[which(x[,1]==1)]

### Plot the residuals as a function of plant
plot(1:10,res0,ylim=c(min(res),max(res)),col="blue",pch=16,
     xlab="plant index",ylab="residuals")
points(1:10,res1,col="red",pch=17)
# triangle 1, circle 0
legend("topright",legend=c("time 0","time 1"),pch=c(16,17),col=c("blue","red"),
       cex=0.75)

### Plot scatterplot of residuals for each plant
plot(res0,res1,xlab="residuals (t=0)",ylab="residuals(t=1)")

### Plot the residuals as a function of soil pH
plot(x[which(x[,1]==0),2],res0,xlab="pH",ylab="residuals",
     ylim=c(min(res),max(res)),xlim=c(min(x[,2]),max(x[,2])),
     col="blue",pch=16)
points(x[which(x[,1]==1),2],res1,col="red",pch=17)

### Plot the residuals as function of time
# plot(x[,1],res,xlab="time",ylab="residuals")
plot(x[which(x[,1]==0),1],res0,xlab="time",ylab="residuals",
     ylim=c(min(res),max(res)),xlim=c(0,1),
     col="blue",pch=16)
points(x[which(x[,1]==1),1],res1,col="red",pch=17)
```



Really, the basic diagnostic plots suggest no problems. While a further investigation of residuals suggest the residuals within plants, i.e. the residual of a plant at different times are positively correlated.

(c)

Hypothesise a new model for your data which allows for observations within a plant to be correlated. Fit the model using a MCMC approximation to the posterior distribution, and present diagnostics for your approximation.

```
K <- 10000
BETA <- array(NA,dim=c(K,3,10)) # only stores beta
X <- cbind(rep(1,20),x)
# initial values
Sigma <- solve(t(X)%*%X)
theta <-fit$coefficients
sigma2 <- var(fit$residuals)

# Gibbs
for (i in 1:K) {
    B <- NULL
    for (j in 1:10) {
        indices <- (2*j-1):(2*j) # for plant j, data 2j-1 and 2j correspond such plant
        delta <- solve(solve(Sigma)+t(X[indices,])%*%X[indices,]/sigma2)
        alpha <- delta%*%(solve(Sigma)%*%theta+t(X[indices,])%*%y[indices]/sigma2)
        beta <- t(as.matrix(rmvnorm(1,alpha,delta)))
        B <- cbind(B,beta)
    }
    BETA[i,,] <- B

    # update theta
    Lambda <- solve(diag(3)+10*solve(Sigma))
```

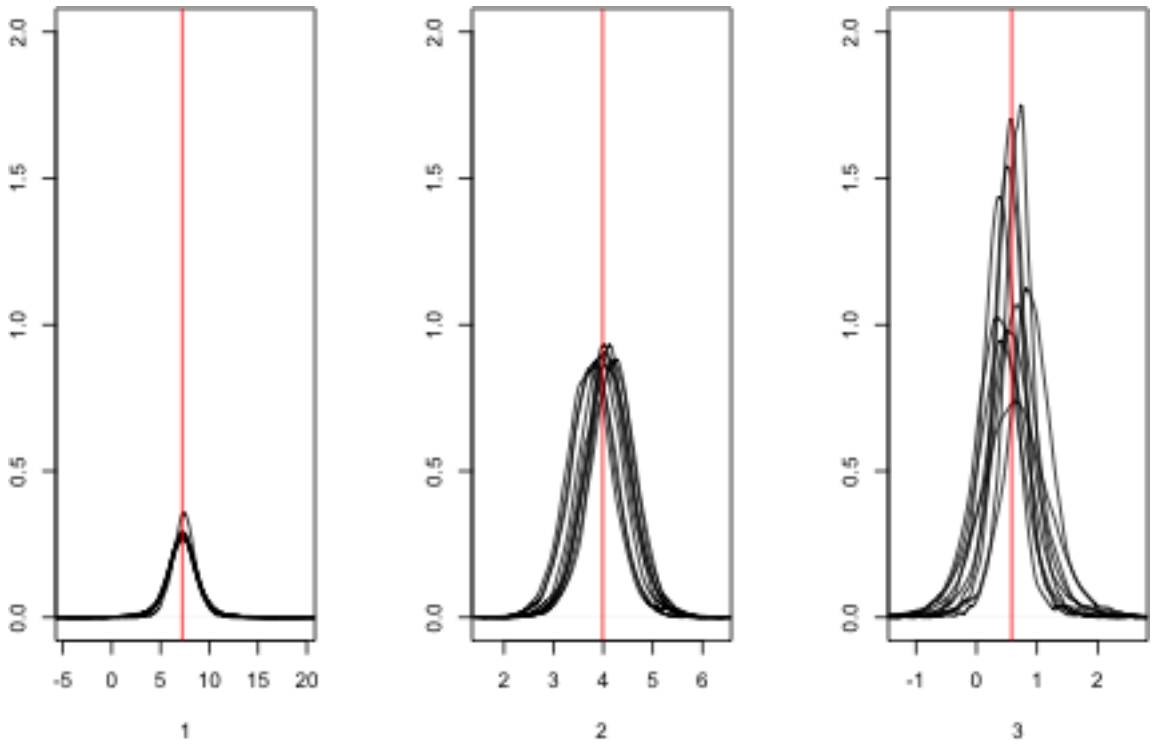
```
    beta.bar <- as.matrix(apply(B,1,mean))
    mu <- Lambda%*%(fit$coefficients+10*solve(Sigma)%*%beta.bar)
    theta <- t(rmvnorm(1,mu,Lambda))

    # update Sigma
    holder <- 0
    for (k in 1:10) {
        holder <- holder+(B[,k]-theta)%*%t(B[,k]-theta)
    }
    Sigma <- solve(rwish(11,solve(diag(3)+holder)))

    # update sigma2
    SSR <- 0
    for (l in 1:10) {
        indices2 <- (2*l-1):(2*l)
        SSR <- SSR+
            sum((y[indices2]-t(as.matrix(B[,l]))%*%t(X[indices2,]))^2)
    }
    sigma2 <- 1/rgamma(1,21/2,(1+SSR)/2)
}

eff.vec <- NULL
for (para in 1:3) {
    for (plant in 1:10) {
        eff.vec <- c(eff.vec,effectiveSize(BETA[,para,plant]))
    }
}
eff.vec
```

```
##     var1     var1     var1     var1     var1     var1     var1     var1
## 2228.331 2652.026 2769.636 2622.044 2481.123 2589.614 2503.097 2315.643
##     var1     var1     var1     var1     var1     var1     var1     var1
## 2376.197 2078.259 6366.902 6356.150 5793.153 6308.321 6283.358 5817.627
##     var1     var1     var1     var1     var1     var1     var1     var1
## 6614.345 5742.134 5746.523 5718.348 2241.198 2775.179 2816.379 2767.550
##     var1     var1     var1     var1     var1     var1
## 2572.795 2774.366 2454.889 2393.860 2533.222 2762.546
```

(d)

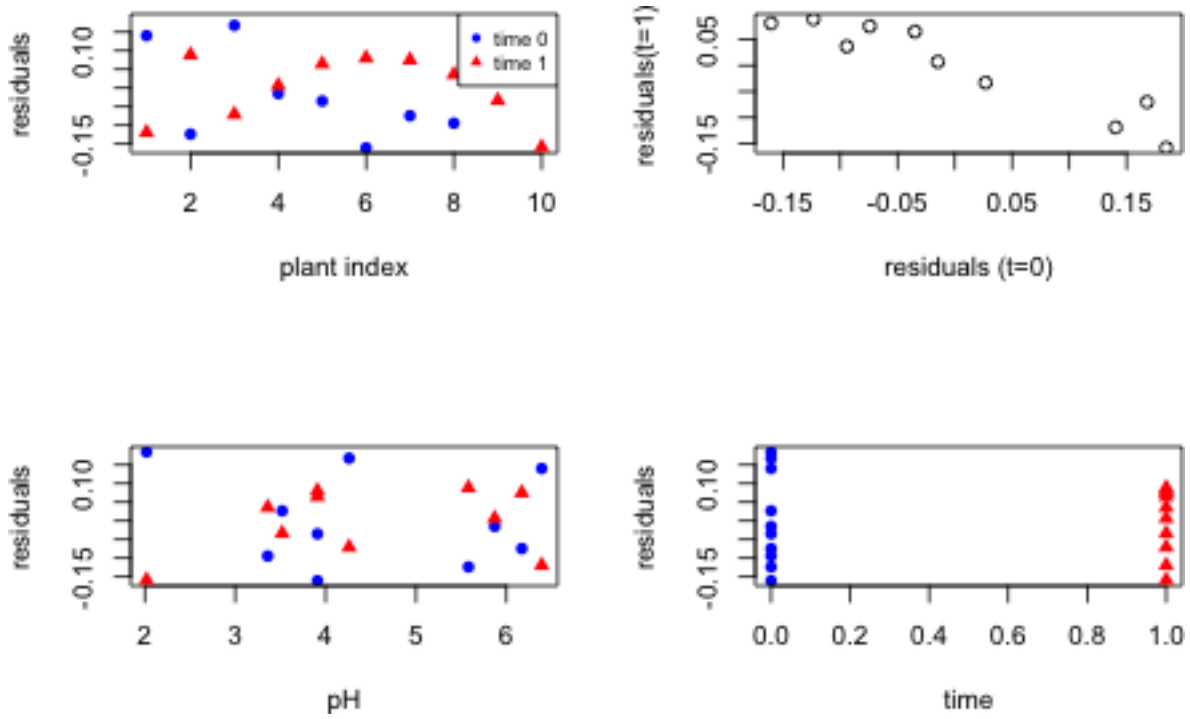
Discuss the results of your data analysis. In particular, discuss similarities and differences between the ordinary linear regression and the model fit with correlated responses. Are the conclusions different?

```
### Plot the ten posteriors and the OLS estimator for each coefficent
par(mfrow=c(1,3))
for (i in 1:3) {
    plot(density(BETA[,i,1]),type="l",ylim=c(0,2),ylab="",
         xlab=i,main="")
    for (j in 2:10) {
        lines(density(BETA[,i,j]))
    }
    abline(v=fit$coefficients[i],col="red")
}
```




```
### Testing the residuals
post.mean <- matrix(rep(NA,3*10),ncol=10)
for (i in 1:3) {
    for (j in 1:10) {
        post.mean[i,j] <- mean(BETA[,i,j])
    }
}
res2 <- NULL
for (i in 1:10) {
    ind <- (2*i-1):(2*i)
    res2 <- c(res2,y[ind]-post.mean[,i]%*%t(X[ind,]))
}

par(mfrow=c(2,2))
ind <- which(x[,1]==0)
plot(1:10,res2[ind],ylim=c(min(res2),max(res2)),col="blue",pch=16,
     xlab="plant index",ylab="residuals")
points(1:10,res2[-ind],col="red",pch=17)
# triangle 1, circle 0
legend("topright",legend=c("time 0","time 1"),pch=c(16,17),col=c("blue","red"),
       cex=0.75)

### Plot scatterplot of residuals for each plant
plot(res2[ind],res2[-ind],xlab="residuals (t=0)",ylab="residuals(t=1)")

### Plot the residuals as a function of soil pH
plot(x[which(x[,1]==0),2],res2[ind],xlab="pH",ylab="residuals",
     ylim=c(min(res2),max(res2)),xlim=c(min(x[,2]),max(x[,2])),
     col="blue",pch=16)
points(x[which(x[,1]==1),2],res2[-ind],col="red",pch=17)
```

```
### Plot the residuals as function of time
# plot(x[,1],res,xlab="time",ylab="residuals")
plot(x[which(x[,1]==0),1],res2[ind],xlab="time",ylab="residuals",
     ylim=c(min(res2),max(res2)),xlim=c(0,1),
     col="blue",pch=16)
points(x[which(x[,1]==1),1],res2[-ind],col="red",pch=17)
```




```
cor(cbind(fit$residuals[ind],fit$residuals[-ind]))
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.9321597
## [2,] 0.9321597 1.0000000
```

```
cor(cbind(res2[ind],res2[-ind]))
```

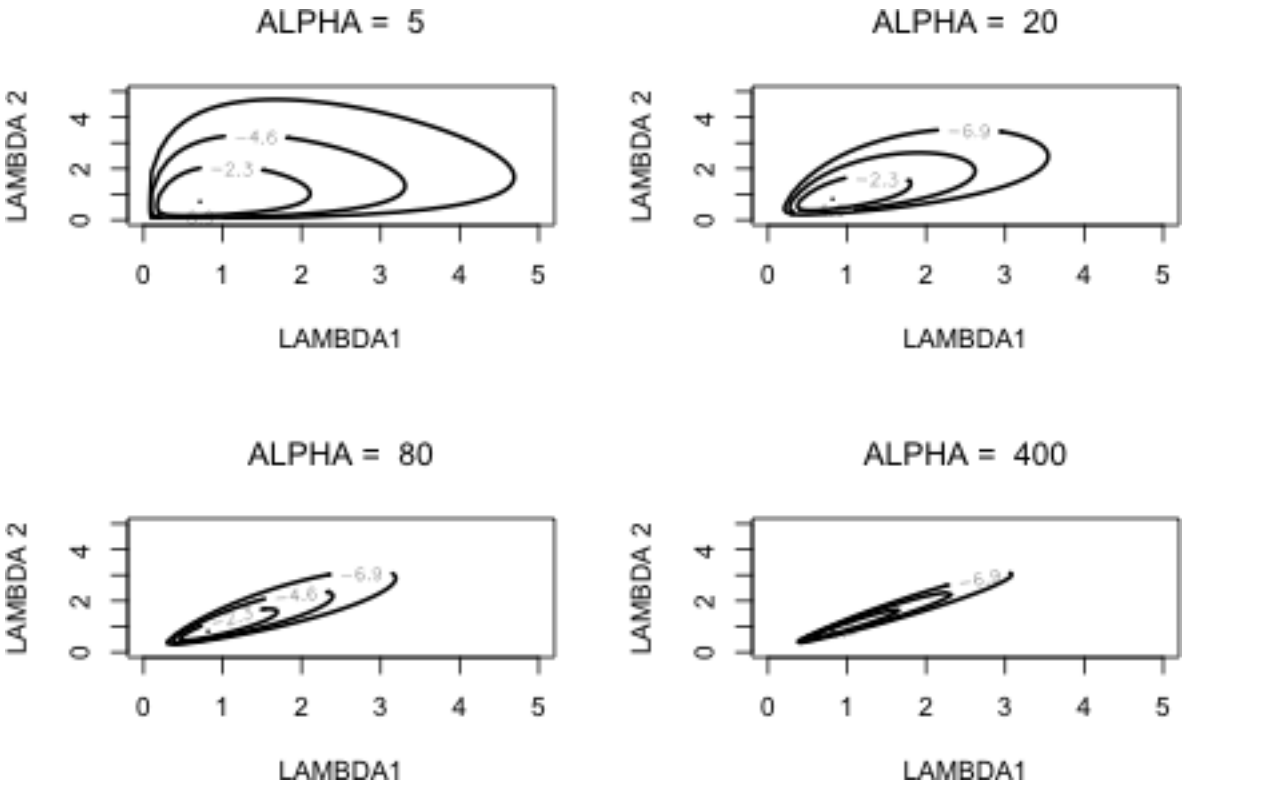
```
##            [,1]       [,2]
## [1,]  1.0000000 -0.9445689
## [2,] -0.9445689  1.0000000
```

?overreacting?

Problem 4

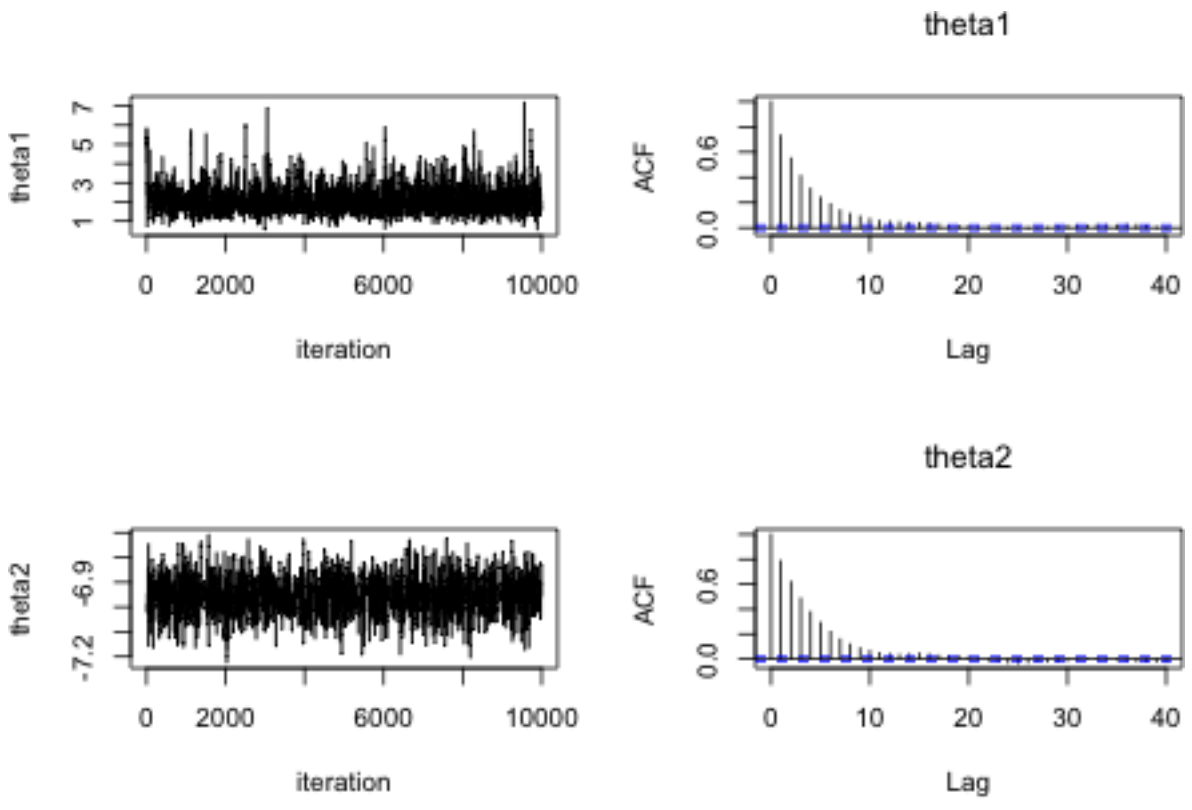
Recall the heart transplant mortality data exercise discussed in class at the end of Chapter 8. The data are in the file `hearttransplants.csv`. Implement a Metropolis-Hastings algorithm to fit a Bayesian hierarchical model to estimate the mortality rates and the shrinkage effect for each hospital. Rank the hospitals based on the mortality rates after heart transplant surgery. Provide some posterior predictive checks to demonstrate that your MH algorithm has converged.

```
heart <- read.csv("hearttransplants.csv")

SIMS <- 10000
y <- heart$y
e <- heart$e

pgexchprior=function(lambda,pars) {
    # lambda = vector of true rates
    alpha=pars[1]; a=pars[2]; b=pars[3]
    (alpha-1)*log(prod(lambda))-(2*alpha+a)*log(alpha*sum(lambda)+b)
}

alpha=c(5,20,80,400)
par(mfrow=c(2,2))
for (j in 1:4){
    mycontour(pgexchprior,c(.001,5,.001,5),c(alpha[j],10,10),
              main=paste("ALPHA = ",alpha[j]),
              xlab="LAMBDA1",ylab="LAMBDA 2")
}
```




```
poissgamexch=function (theta, datapar)
{
```

```
    y = datapar$data[, 2]
    e = datapar$data[, 1]
    z0 = datapar$z0
    alpha = exp(theta[1])
    mu = exp(theta[2])
    beta = alpha/mu
    logf = function(y, e, alpha, beta)
        lgamma(alpha + y) - (y + alpha) * log(e + beta) + alpha * log(beta) - lgamma(alpha)
    val = sum(logf(y, e, alpha, beta))
    val = val + log(alpha) - 2 * log(alpha + z0)
    return(val)
}

par(mfrow=c(1,1))
datapar = list(data = heart, z0 = 0.53)
start = c(4, -7)
fitgibbs = gibbs(poissgamexch, start, SIMS, c(2.5,.5), datapar)
# scale(1,.15)

par(mfrow=c(2,2))
# 1000 as burning
plot(fitgibbs$par[1:SIMS,1],type="l",xlab="iteration",ylab="theta1")
acf(fitgibbs$par[1:SIMS,1],main="theta1")
plot(fitgibbs$par[1:SIMS,2],type="l",xlab="iteration",ylab="theta2")
acf(fitgibbs$par[1:SIMS,2],main="theta2")
```




```
par(mfrow=c(1,1))
```

```
fitgibbs$accept
```

```
##       [,1]  [,2]
## [1,] 0.263 0.181
```

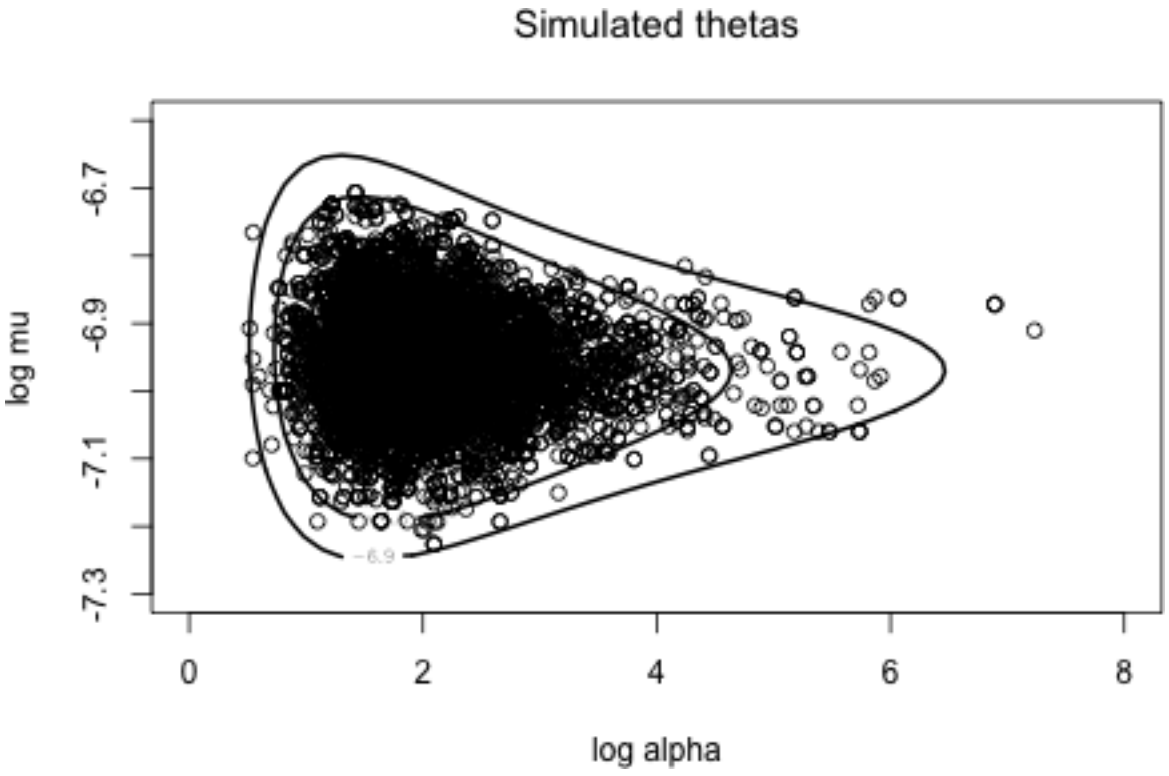
```
effectiveSize(fitgibbs$par[,1])
```

```
##     var1
## 1447.831
```

```
effectiveSize(fitgibbs$par[,2])
```

```
##     var1
## 1199.132
```

```
mycontour(poissgamexch, c(0, 8, -7.3, -6.6), datapar,
          xlab="log alpha",ylab="log mu",main = "Simulated thetas")
points(fitgibbs$par[, 1], fitgibbs$par[, 2])
```

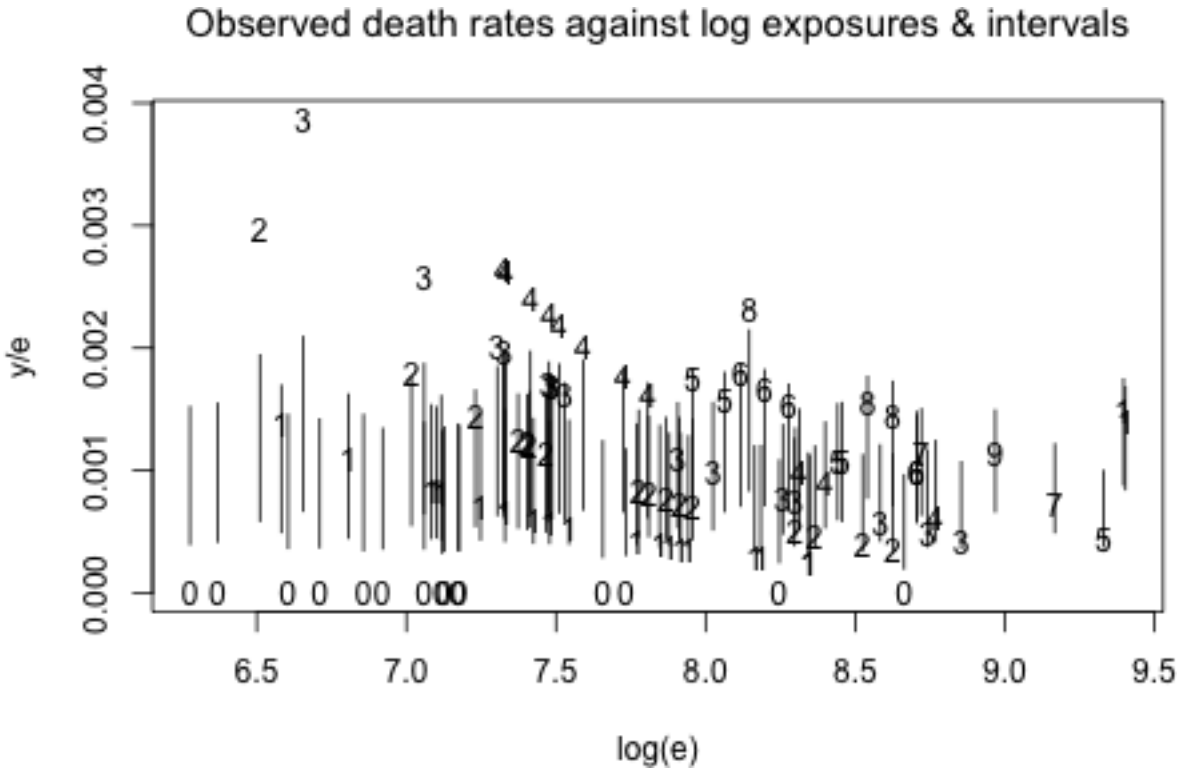



```
alpha = exp(fitgibbs$par[, 1])
mu = exp(fitgibbs$par[, 2])
lam1 = rgamma(1000, y[1] + alpha, e[1] + alpha/mu)
quantile(lam1,c(0.05,0.95))
```

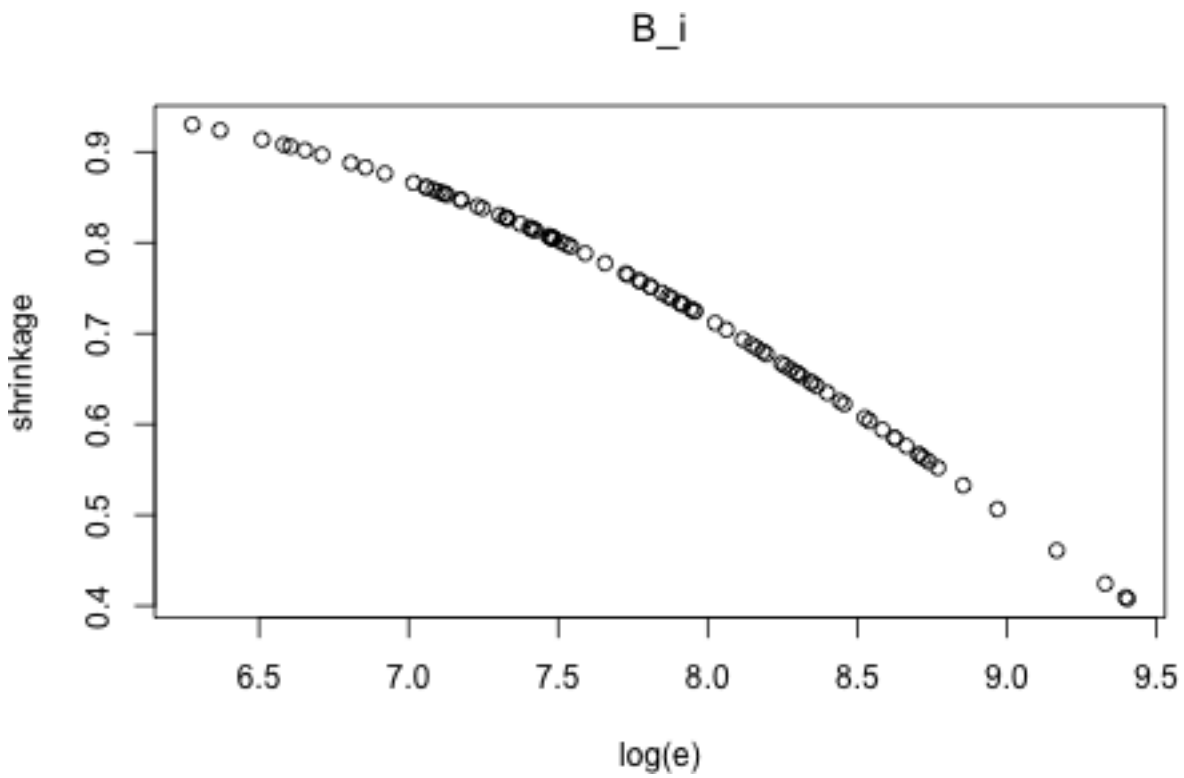
```
##           5%          95%
## 0.0003675816 0.0015310454
```

```
alpha = exp(fitgibbs$par[, 1])
mu = exp(fitgibbs$par[, 2])
plot(log(e), y/e, pch = as.character(y),main = "Observed death rates against log exposures & intervals"
for (i in 1:94) {
    lami = rgamma(1000, y[i] + alpha, e[i] + alpha/mu)
```

```
    probint = quantile(lami, c(0.05, 0.95))
    lines(log(e[i]) * c(1, 1), probint)
}
```




```
shrink=function(i) mean(alpha/(alpha + e[i] * mu))
shrinkage=sapply(1:94, shrink)
plot(log(e), shrinkage,main = "B_i")
```




```
mrate=function(i) mean(rgamma(1000, y[i] + alpha, e[i] + alpha/mu))
hospital=1:94
meanrate=sapply(hospital,mrate)
c("best:",hospital[meanrate==min(meanrate)],"worst:",hospital[meanrate==max(meanrate)])
```

```
## [1] "best:"  "85"     "worst:" "68"
```

Problem 5

Non-conjugate hierarchical models: An experiment is conducted to estimate θ, the probability of developing a tumor in a population of female rats that receive a dose of drug X. (Such studies are routinely done in the evaluation of drugs for possible clinical application). Suppose J such experiments have been conducted historically. In the j^{th} historical experiment, let the number of rats with tumors be y_j and let n_j be the total number of rats tested in the j^{th} experiment. We model the y_j's as independent binomial data, given the sample size n_j, and study-specific means θ_j.

Suppose we assume that the tumor probabilities θ follow a normal distribution on the log-odds scale, that is, $\text{logit}(\theta_j) \sim N(\mu, \tau^2)$.

(a)

Write the joint posterior density $p(\theta, \mu, \tau^2 \mid \mathbf{y})$ (where $\theta = (\theta_1, \ldots, \theta_J)$ and $\mathbf{y} = (y_1, \ldots, y_J)$).

Solution:

$$\begin{aligned} p(\theta, \mu, \tau^2 \mid \mathbf{y}) &\propto p(\theta, \mu, \tau^2) p(\mathbf{y} \mid \theta, \mu, \tau^2) \\ &= p(\theta, \mu, \tau^2) p(\mathbf{y} \mid \theta) \\ &= p(\theta \mid \mu, \tau^2) p(\mu, \tau^2) \\ &= p(\mu, \tau^2) \cdot \prod_{j=1}^{J} \left(\theta_j^{-1} (1-\theta_j)^{-1} \tau^{-1} \exp\left(-(\frac{1}{2}\text{logit}(\theta_j) - \mu)^2 / \tau^2 \right) \right. \\ &\quad \left. \cdot \prod_{j=1}^{J} \left(\theta_j^{y_j} (1-\theta_j)^{n_j - y_j} \right) \right. \end{aligned}$$

(b)

To obtain the marginal posterior distribution $p(\mu, \tau^2 \mid \mathbf{y})$, we can integrate the joint distribution in (a) over θ. Show that this integral has no closed-form expression.

Solution:

There is no way to integrate such complex integrand above. Nothing looks familiar.

(c)

We can also compute the marginal posterior distribution of (μ, τ^2) using the conditional probability formula,

$$p(\mu, \tau^2 \mid \mathbf{y}) = \frac{p(\theta, \mu, \tau^2 \mid \mathbf{y})}{p(\theta \mid \mu, \tau^2, \mathbf{y})}$$

Why is the above expression not helpful to evaluate $p(\mu, \tau^2 \mid \mathbf{y})$?

In practice, we can solve this problem by normal approximation, importance sampling, and MCMC simulation.

Solution:

Here we only know $p(\theta \mid \mu, \tau^2, \mathbf{y})$ is proportional to the target formula. However, in order to evalutate $p(\mu, \tau^2 \mid \mathbf{y})$, we need to know $p(\theta \mid \mu, \tau^2, \mathbf{y})$ explicitly.

because our goal is to use this density to find another density that depends on μ and τ.