Bayesian Modeling & Regression

Ting Lu (NetID: tl3511)

2022-12-13

Install Package

```
# library(tidyverse): data manipulation & visualization
# library(MASS): statistical methods & regression analysis
# library(MCMCpack): MCMC simulations &  Bayesian statistical modeling and parameter estimation
```

`Tidyverse` includes multiple packages: `ggplot2`, `dplyr`, `tidyr`, `readr`, `purrr`, `tibble`, `stringr`

1 (hierarchical model)

The food habits of 371 male and 192 female healthy employees with average age 38.3 years were examined by a 3-day food record with an additional interview.Our goal is to contrast the variability of cholesterol intake (chol) in mg/day between the subsidiaries to the variability within the subsidiaries. To achieve this goal, we model the data as follows:

$$
\begin{aligned}
y_{ij}|\theta_i,\sigma^2 &\overset{ind}{\sim} N(\theta_i,\sigma^2) && \text{for } j=1,\ldots,m_i;\ \ i=1,\ldots,n\\
\theta_i|\mu,\tau^2 &\overset{iid}{\sim} N(\mu,\tau^2) && \text{for } i=1,\ldots,n\\
\sigma^2 &\sim p(\sigma^2)\\
\tau^2 &\sim p(\tau^2)\\
\mu &\sim p(\mu)
\end{aligned}
$$

Data information

```
IBB = read.csv("/Users/tinglu/Bayesian/IBBENS.csv")
head(IBB)
```

```
##     chol subsidiary
## 1 395.33          1
## 2 449.00          1
## 3 278.33          1
```

```
## 4 254.00          1
## 5 408.67          1
## 6 539.67          1
```

```
table(IBB$subsidiary)
```

```
##
##  1  2  3  4  5  6  7  8
## 82 51 71 71 62 69 74 83
```

(a)

Describe in words what the various components of the hierarchical model represent and discuss how the model could be modified to account for additional sources of variability.

Ans: It is a hierarchical normal model. In the first stage, y_{ij} is the cholesterol intake for subject $j = 1, ..., m_i$ in subsidiaries $i = 1, ..., n$. In the second stage, θ_i is the average cholesterol intake for subsidiary i.σ^2 is the variance of the cholesterol intake and now we assume the $\sigma_i^2 = \sigma^2$, indicating the variability in cholesterol intake in each subsidiary is the same. μ, τ^2 is the mean and variance of θ_i, which represents the variability within the subsidiaries. σ^2, τ^2, μ all have their own distribution. If we assume $\sigma_i^2 \neq \sigma^2$ which means the variability in cholesterol intake in each subsidiary is not the same, the model will account for additional sources of variability.

(b)

Using the following prior specifications

```
knitr::include_graphics("/Users/tinglu/Bayesian/IBB2.png",error = F)
```

$$
\begin{aligned}
p(\sigma^2) &= \text{Inverse Gamma}(\tfrac{\nu_0}{2}, \tfrac{\nu_0\sigma_0^2}{2}) = \text{Inverse Gamma}(10^{-3}, 10^{-3}) \\
p(\tau^2) &= \text{Inverse Gamma}(\tfrac{\eta_0}{2}, \tfrac{\eta_0\tau_0^2}{2}) = \text{Inverse Gamma}(10^{-3}, 10^{-3}) \\
p(\mu) &= \text{N}(0, 10^6)
\end{aligned}
$$

Fit the hierarchical model above to the cholesterol intake data of the 8 subsidiaries and run the MCMC algorithm long enough so that the effective sample size of each parameters is at least 1000.

```
# prior parameters
mu.0 = 0
gamma2.0 = 10^6
nu.0 = 10^(-3)*2
sigma2.0 = 10^(-3)*2/nu.0
eta.0 = 10^(-3)*2
tau2.0 = 10^(-3)*2/eta.0
```

```
# define update function
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){

    # here we create a vector of length equal to the number of observations
    # where we have n_1 times theta_1, n_2 times theta_2,..., n_m times theta_m
    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)
}
```

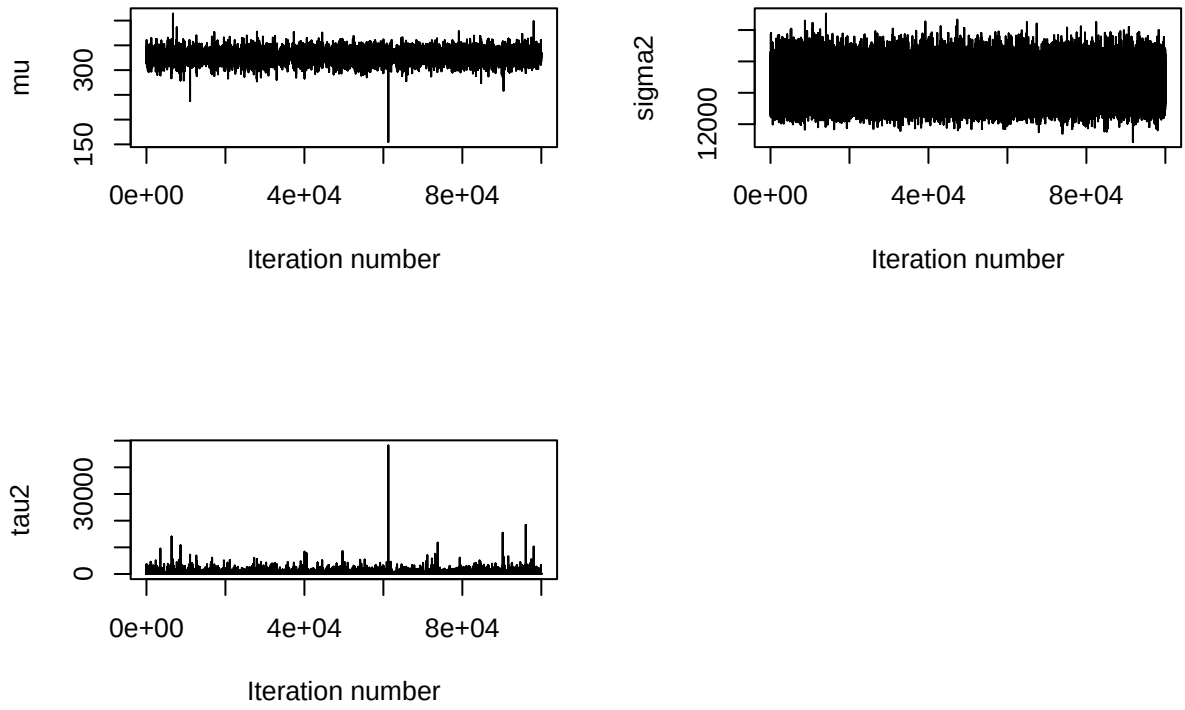
```
# initial values
## The second column in the data is the subsidairy indicator, the first is the cholesterol intake
sub <- IBB[,2]
y <- IBB[,1]
## This is the vector with the number n_j of observations per subsidairy indicator
n.j <- as.numeric(table(sub))
## Number of subsidairy
m <- length(n.j)
```

```r
### Sample averages and sample variance by subsidairy indicator
y.bar.sub <- rep(0,m)
s2.sub <- rep(0,m)

for(i in 1:m){
    y.bar.sub[i] <- mean(y[which(sub==i)])
    s2.sub[i] <- var(y[which(sub==i)])
}

## Average of the sub-specific mean and variances
mean.y.bar.sub <- mean(y.bar.sub)
mean.s2.sub <- mean(s2.sub)
var.y.bar.sub <- var(y.bar.sub)

init.theta.j <- y.bar.sub
init.mu <- mean.y.bar.sub
init.sigma2 <- mean.s2.sub
init.tau2 <- var.y.bar.sub
```

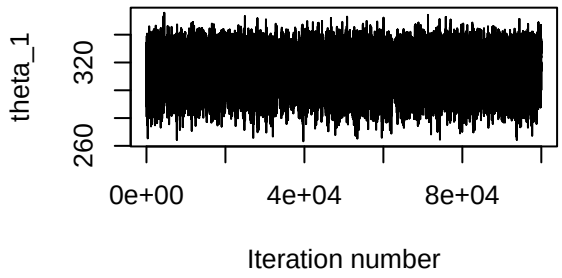
```r
# Run Gibbs sampling
S <- 100000

## We store the subsidairy indicator specific parameters, the theta_js in one matrix
theta.MCMC <- matrix(0,nrow=S,ncol=m)
## We store the other parameters, mu, sigma2, tau2, in a second matrix
other.pars.MCMC <- matrix(0,S,ncol=3)
new.theta.j <- rep(0,m)

set.seed(0) #The random seed here is different from that for the slides of the lecture 8. Compare the r
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,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,y.bar.sub[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)
```

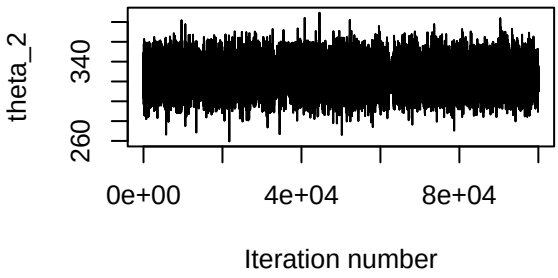
```
}
```

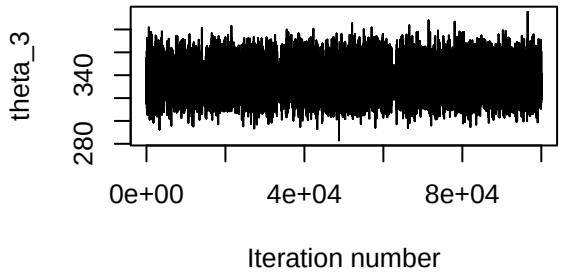
```
### MCMC diagnostic

## Trace plots
par(mfrow=c(2,2))
plot(other.pars.MCMC[,1],xlab="Iteration number",ylab="mu",type="l")
plot(other.pars.MCMC[,2],xlab="Iteration number",ylab="sigma2",type="l")
plot(other.pars.MCMC[,3],xlab="Iteration number",ylab="tau2",type="l")

par(mfrow=c(2,2))
```

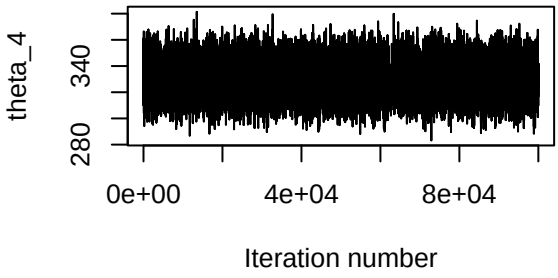



```
plot(theta.MCMC[,1],xlab="Iteration number",ylab="theta_1",type="l")
plot(theta.MCMC[,2],xlab="Iteration number",ylab="theta_2",type="l")
plot(theta.MCMC[,3],xlab="Iteration number",ylab="theta_3",type="l")
plot(theta.MCMC[,4],xlab="Iteration number",ylab="theta_4",type="l")
```

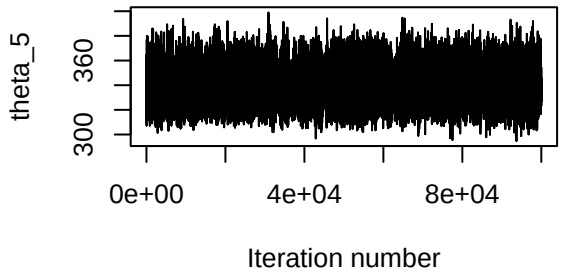


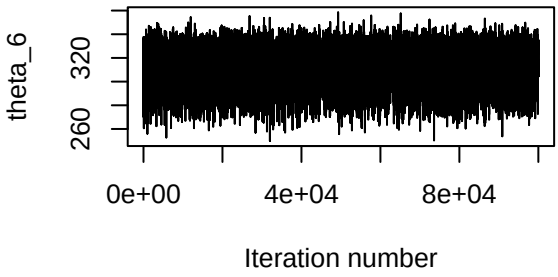


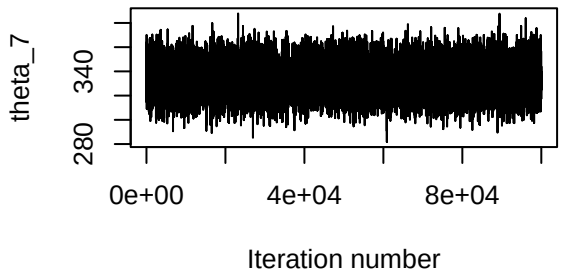


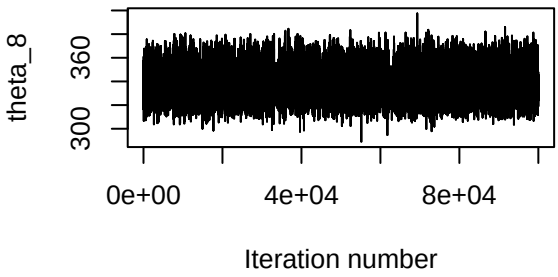



```
plot(theta.MCMC[,5],xlab="Iteration number",ylab="theta_5",type="l")
plot(theta.MCMC[,6],xlab="Iteration number",ylab="theta_6",type="l")
plot(theta.MCMC[,7],xlab="Iteration number",ylab="theta_7",type="l")
plot(theta.MCMC[,8],xlab="Iteration number",ylab="theta_8",type="l")
```



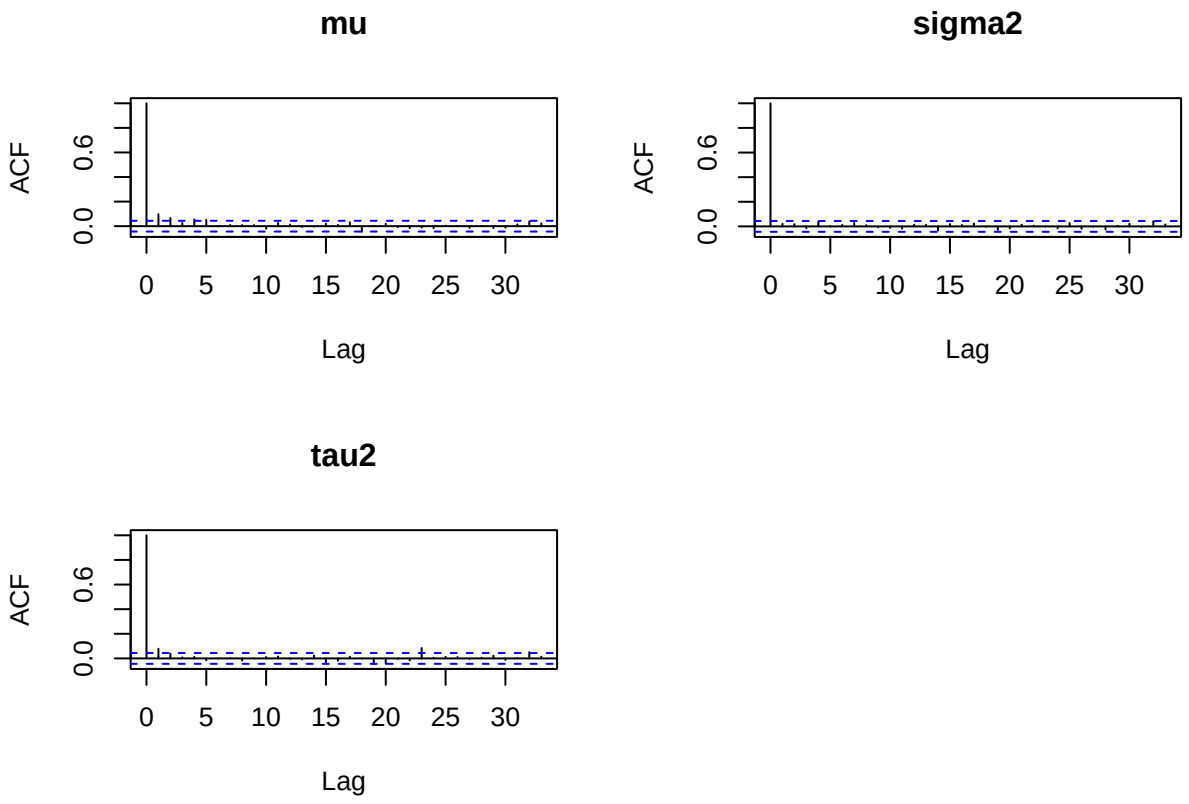




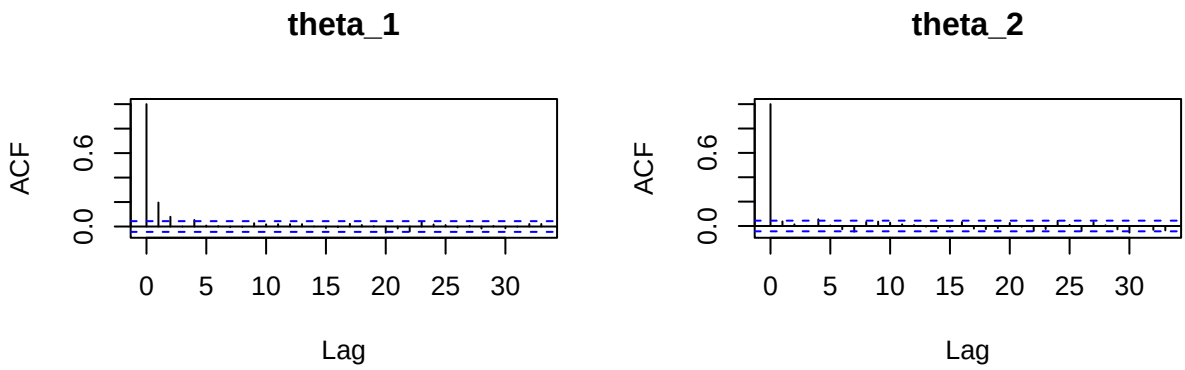


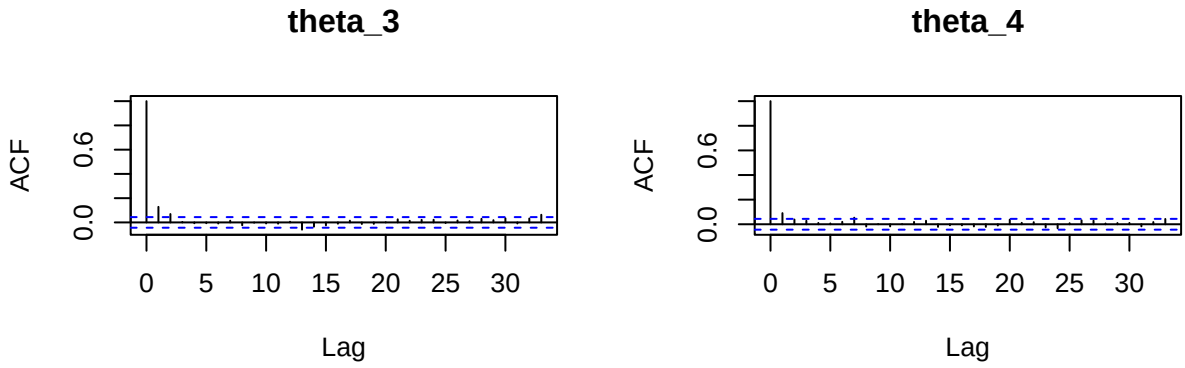

```
## ACF plots
thin = seq(1,S, by = 50)
```

```
par(mfrow=c(2,2))
acf(other.pars.MCMC[thin,1],main="mu")
acf(other.pars.MCMC[thin,2],main="sigma2")
acf(other.pars.MCMC[thin,3],main="tau2")

par(mfrow=c(2,2))
```

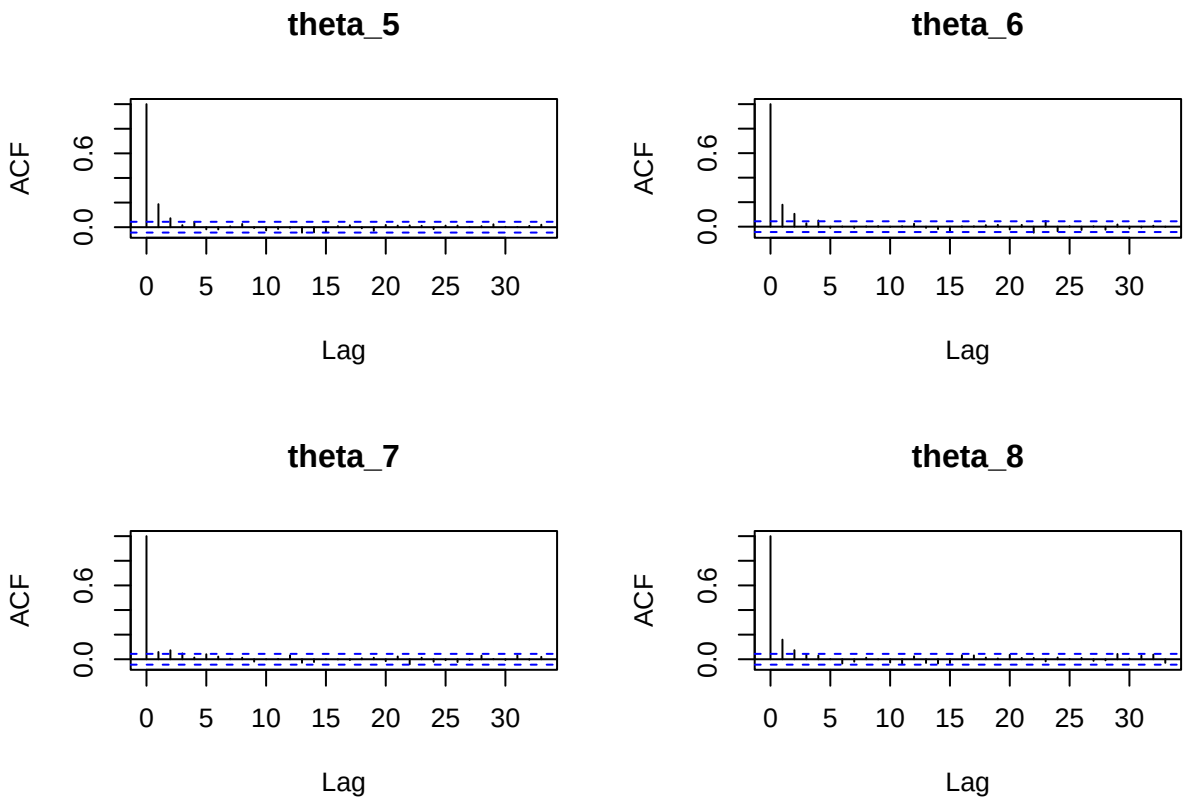



```
acf(theta.MCMC[thin,1],main="theta_1")
acf(theta.MCMC[thin,2],main="theta_2")
acf(theta.MCMC[thin,3],main="theta_3")
acf(theta.MCMC[thin,4],main="theta_4")
```






```
acf(theta.MCMC[thin,5],main="theta_5")
acf(theta.MCMC[thin,6],main="theta_6")
acf(theta.MCMC[thin,7],main="theta_7")
acf(theta.MCMC[thin,8],main="theta_8")
```




```
## Effective sample size
e1<-effectiveSize(other.pars.MCMC[thin,1])
e2<-effectiveSize(other.pars.MCMC[thin,2])
e3<-effectiveSize(other.pars.MCMC[thin,3])

e4<-effectiveSize(theta.MCMC[thin,1])
e5<-effectiveSize(theta.MCMC[thin,2])
e6<-effectiveSize(theta.MCMC[thin,3])
e7<-effectiveSize(theta.MCMC[thin,4])
e8<-effectiveSize(theta.MCMC[thin,5])
e9<-effectiveSize(theta.MCMC[thin,6])
e10<-effectiveSize(theta.MCMC[thin,7])
e11<-effectiveSize(theta.MCMC[thin,8])

cbind(e1,e2,e3,e4,e5,e6,e7,e8,e9,e10,e11)
```

```
##            e1   e2       e3       e4       e5       e6       e7       e8
## var1 1190.955 2000 1601.508 1154.466 1628.081 1387.089 1667.285 1265.759
##            e9      e10     e11
## var1 1195.876 1425.747 1311.99
```

(c)

Make posterior inference on the average cholesterol intake in each of the eight subsidiaries: provide posterior means and 95% credible intervals

```r
### Marginal posterior summaries
burnin <- 5000
## Theta
post.mean.theta <- apply(theta.MCMC[(burnin+1):S,],2,mean)
post.mean.theta
```

```
## [1] 319.8671 327.4810 332.6090 329.6485 335.0535 317.8713 331.2932 334.4419
```

```r
post.quantile.theta <- apply(theta.MCMC[(burnin+1):S,],2,quantile, probs = c(0.025, 0.975))
post.quantile.theta
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
## 2.5%  292.6141 306.0166 316.0885 312.0623 318.0547 286.3692 314.8587 318.4491
## 97.5% 337.3800 346.9019 355.5699 349.2813 362.2636 337.0854 352.6175 358.4930
```

```r
rbind(posterior.mean = post.mean.theta,post.quantile.theta)
```

```
##                    [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## posterior.mean 319.8671 327.4810 332.6090 329.6485 335.0535 317.8713 331.2932
## 2.5%           292.6141 306.0166 316.0885 312.0623 318.0547 286.3692 314.8587
## 97.5%          337.3800 346.9019 355.5699 349.2813 362.2636 337.0854 352.6175
##                    [,8]
## posterior.mean 334.4419
## 2.5%           318.4491
## 97.5%          358.4930
```

(d)

Determine which subsidiary has the highest posterior probability of having employees with the highest cholesterol intake. Analogously, given the data, determine which subsidiary is more likely to have employees with the lowest cholesterol intake per day.

```r
which.max(post.mean.theta)
```

```
## [1] 5
```

```r
which.min(post.mean.theta)
```

```
## [1] 6
```

Subsidiary 5 has the highest posterior probability of having employees with the highest cholesterol intake.

Subsidiary 6 is more likely to have employees with the lowest cholesterol intake per day.

(e)

Make posterior inference on the population average level of cholesterol intake: provide posterior mean and 95% credible interval.

```r
# Mu
mean(other.pars.MCMC[(burnin+1):S,1])
```

```
## [1] 328.5337
```

```r
quantile(other.pars.MCMC[(burnin+1):S,1],c(0.025,0.975))
```

```
##     2.5%    97.5%
## 314.4920 342.8445
```

(d)

Discuss whether the data suggest that there is more "between-subsidiaries" variability than "within-subsidiary" variability: provide posterior means and 95% credible intervals for the appropriate parameters to illustrate this point.

```r
# "between-subsidiaries" variability: tau^2
mean(other.pars.MCMC[(burnin+1):S,3])
```

```
## [1] 200.0268
```

```r
quantile(other.pars.MCMC[(burnin+1):S,3],c(0.025,0.975))
```

```
##         2.5%        97.5%
## 2.106449e-03 1.092173e+03
```

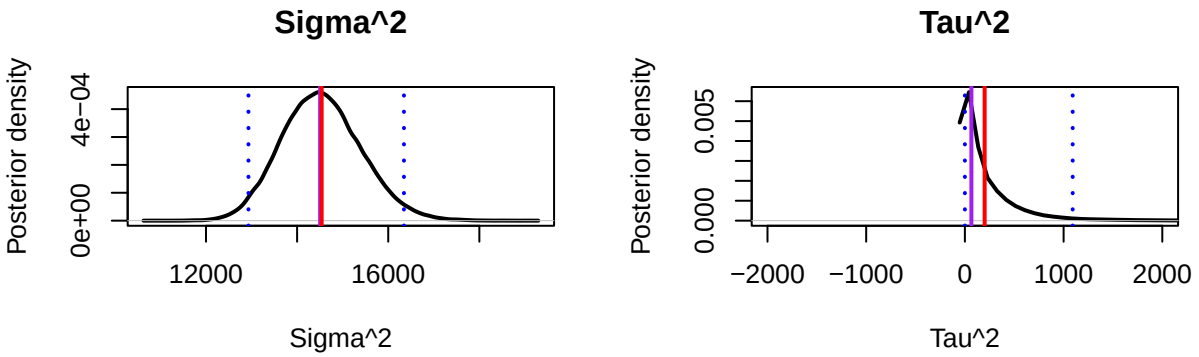
```r
# "within-subsidiary" variability: sigma^2
mean(other.pars.MCMC[(burnin+1):S,2])
```

```
## [1] 14538.73
```

```r
quantile(other.pars.MCMC[(burnin+1):S,2],c(0.025,0.975))
```

```
##    2.5%   97.5%
## 12932.0 16345.1
```

```r
# Plot
par(mfrow = c(2,2))
plot(density(other.pars.MCMC[(burnin+1):S,2]),col="black",lwd=2,lty=1,xlab="Sigma^2",ylab="Posterior de
abline(v=quantile(other.pars.MCMC[(burnin+1):S,2],0.025),col="blue",lwd=2,lty=3)
abline(v=quantile(other.pars.MCMC[(burnin+1):S,2],0.975),col="blue",lwd=2,lty=3)
abline(v=median(other.pars.MCMC[(burnin+1):S,2]),col="purple",lwd=2,lty=1)
abline(v=mean(other.pars.MCMC[(burnin+1):S,2]),col="red",lwd=2,lty=1)


plot(density(other.pars.MCMC[(burnin+1):S,3]),col="black",lwd=2,lty=1,xlab="Tau^2",ylab="Posterior densi
abline(v=quantile(other.pars.MCMC[(burnin+1):S,3],0.025),col="blue",lwd=2,lty=3)
abline(v=quantile(other.pars.MCMC[(burnin+1):S,3],0.975),col="blue",lwd=2,lty=3)
abline(v=median(other.pars.MCMC[(burnin+1):S,3]),col="purple",lwd=2,lty=1)
abline(v=mean(other.pars.MCMC[(burnin+1):S,3]),col="red",lwd=2,lty=1)
```

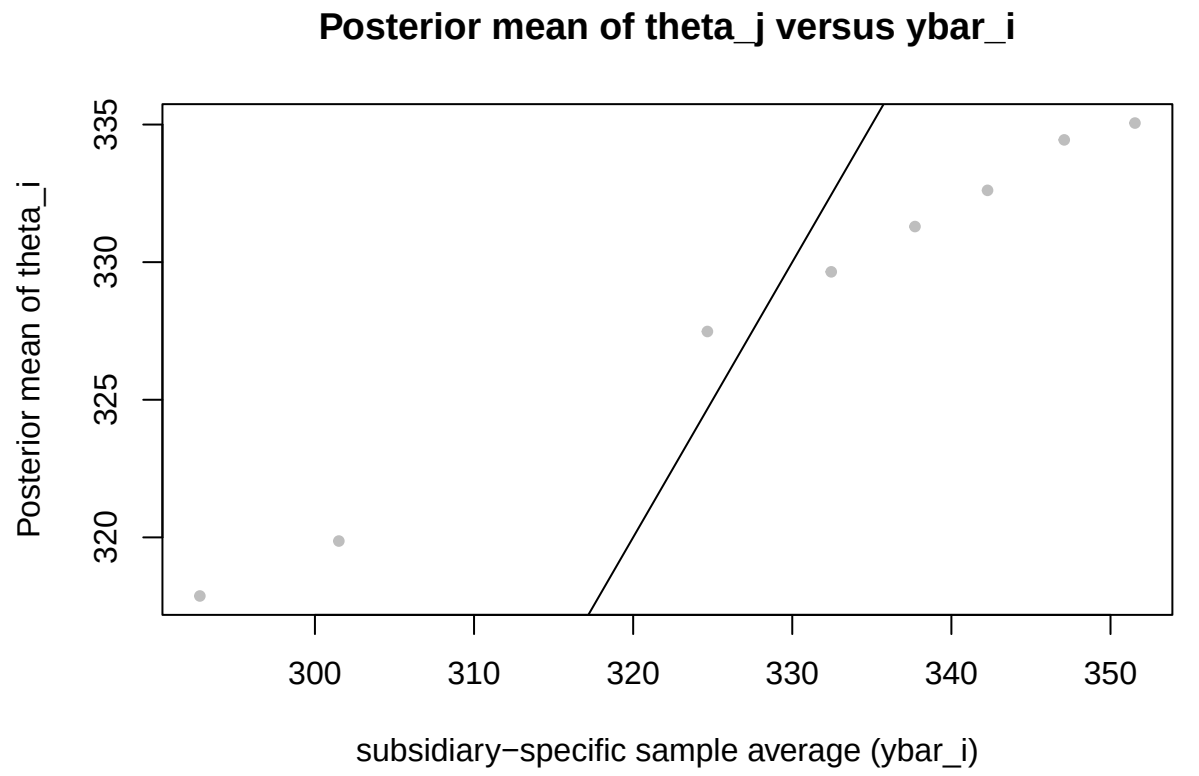


There is less "between-subsidiaries" variability than "within-subsidiary" variability because σ^2 is much larger than τ^2.

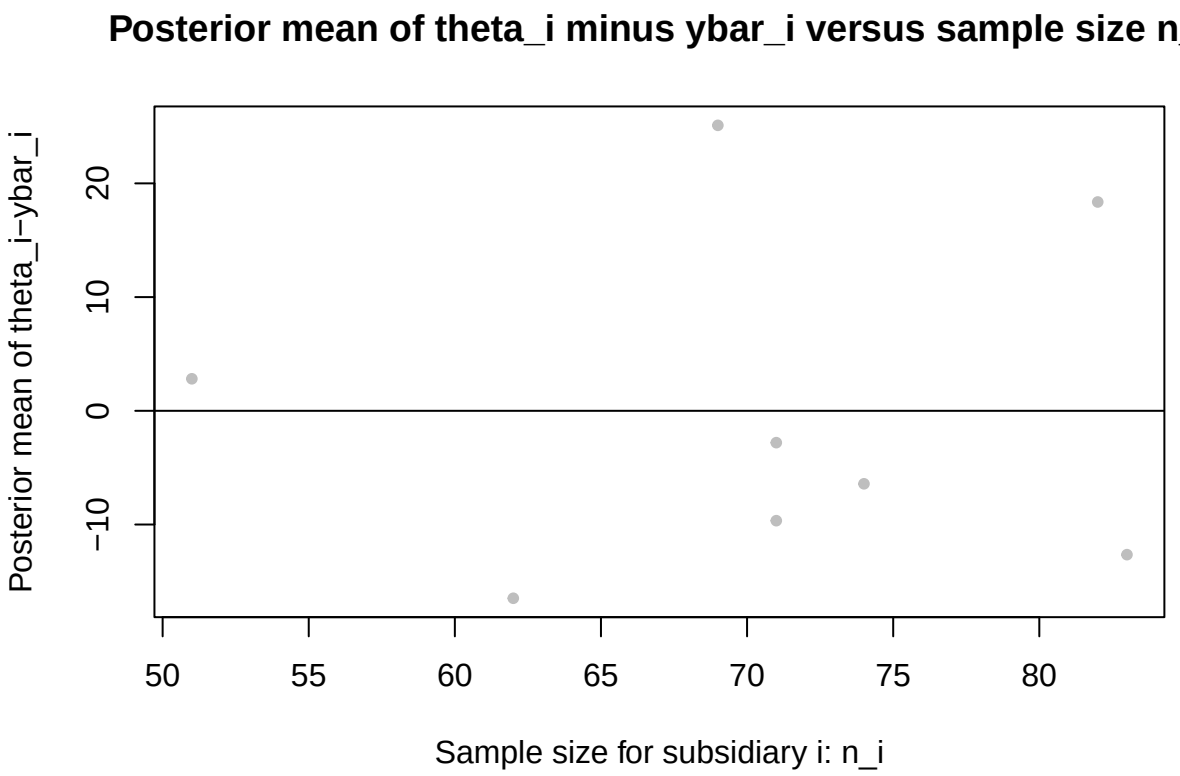
(g)

Produce an appropriate plot that illustrates the concept of shrinkage and discuss whether your analysis suggests that there is a large or moderate level of shrinkage

```
#### Shrinkage plots

plot(y.bar.sub,post.mean.theta,xlab="subsidiary-specific sample average (ybar_i)",ylab="Posterior mean
col="grey",pch=20,main="Posterior mean of theta_j versus ybar_i")
abline(a=0,b=1,col="black")
```




```
plot(n.j,post.mean.theta-y.bar.sub,xlab="Sample size for subsidiary i: n_i",ylab="Posterior mean of the
col="grey",pch=20,main="Posterior mean of theta_i minus ybar_i versus sample size n_i")
abline(h=0,col="black")
```



Shrinkage: the expected value of posterior mean θ_i is pulled from the subsidiary-specific sample average $\hat{y}_i$ towards the overall average μ by an amount that depends on sample size of each subsidiary n_i. If n_i is smaller, the amount of the shrinkage is larger. This says that if there is not a lot of data from subsidiary i, then we borrow information from the rest of the population. On the other hand, if there is a lot of data from subsidiary i, then we don't need borrow information from the rest of the population to make inference about θ_i.

From the scatter plot, we can see the dots is far away from the 45° degree line, indicating the shrinkage is large.

2 (hierachical regression)

Double-blinded RCT,: Itraconazol 250 mg daily (treat=0) or Lamisil 250 mg daily (treat=1). The patients received treatment for 12 weeks and were evaluated at 0, 1, 2, 3, 6, 9 and 12 months. Outcome is the unaffected nail length for the big toenail. We want to determine whether the treatment works on average and if there is a difference among the treatments.

Use linear fixed-effect regression model:

$$y_{ij} = \beta_0 + \beta_1 * t_{ij} + \beta_2 * (t_{ij} * treat_i) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

y_{ij}: outcome of patient i at time $t_{ij} = 0, 1, 2, 3, 6, 9, 12$ under treatment effect. No main effect for treatment was included in the model since at baseline the effect of treatment must be zero given the randomized character of the study

```
toe = read.table("/Users/tinglu/Bayesian/toenail.txt", header=T)
head(toe)
```

```
##   obs treat id time response
```

```
## 1   1     0  2    0        4
## 2   2     0  2    1        6
## 3   3     0  2    2        7
## 4   4     0  2    3        9
## 5   5     0  2    6       13
## 6   6     0  2    9        0
```

(a)

Fit the above fixed effect model using the following priors:

```
knitr::include_graphics("/Users/tinglu/Bayesian/toe.png",error = F)
```

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \sim N_3 \left(\begin{pmatrix} 2.5 \\ 0.6 \\ 0 \end{pmatrix}, 100 \times \mathbf{I}_3 \right)$$

$$\sigma^2 \sim \text{Inverse Gamma}(1, 3.7)$$

Derive posterior means and 95% credible intervals for the regression coefficients and for s2 and determine whether (i) the unaffected toe nail grows significantly over time; (ii) there is a difference between the two treatments over time.

Ans: Transform the equation, we can find it is a fixed effect that impact the slope.

$$y_{ij} = \beta_0 + \beta_1 * t_{ij} + \beta_2 * (t_{ij} * treat_i) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

$$y_{ij} = \beta_0 + (\beta_1 + \beta_2 * treat_i) t_{ij} + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

```
S <- 10000
burn_in <- 1000
X <- cbind(1, toe$time, toe$time * toe$treat)
y <- toe$response
n <- nrow(X)
p <- ncol(X)
# prior
beta0 <- c(2.5, 0.6, 0)
sigma0 <- diag(rep(100, 3))
nu0 <- 2
s20 <- 3.7
# initial values
beta <- beta0
s2 <- s20

Fix.beta <- matrix(nrow = S, ncol = length(beta0))
```

```
Fix.sigma <- numeric(S)
set.seed(0)

# Gibbs sampling
for (s in 1:S) {
  V <- solve(solve(sigma0) + (t(X) %*% X)/s2)
  m <- V %*% (solve(sigma0) %*% beta0 + (t(X) %*% y)/s2)
  beta <- mvrnorm(1, m, V)
  ssr <- (t(y) %*% y) - (2 * t(beta) %*% t(X) %*% y) + (t(beta) %*%
                                                        t(X) %*% X %*% beta)
  s2 <- 1/rgamma(1, (nu0 + n)/2, (nu0 * s20 + ssr)/2)

  Fix.beta[s, ] <- beta
  Fix.sigma[s] <- s2
}
```

```
post.mean.fix <- apply(cbind(Fix.beta[burn_in:S, ], Fix.sigma[burn_in:S]), 2, mean)
post.quantile.fix <- apply(cbind(Fix.beta[burn_in:S, ], Fix.sigma[burn_in:S]), 2, quantile,
c(0.025, 0.975))
fix <- rbind(post.mean.fix, post.quantile.fix)
fix
```

```
##                   [,1]      [,2]       [,3]     [,4]
## post.mean.fix 2.664289 0.5609139 0.06346246 13.55644
## 2.5%          2.423144 0.5096115 0.00667572 12.70317
## 97.5%         2.909417 0.6125981 0.12015786 14.45544
```

Ans: Both 95% credible intervals of time and the interaction term of treat and time don't include 0, so unaffected toe nail grows significantly over time and there is significant difference between the two treatments over time.

(b)

Each patients has his/her own physiological and biological characteristics and might respond to treatment differently.

Use following random intercept and slope model:

$$y_{ij} = \beta_{0i} + \beta_{1i} * t_{ij} + \beta_2 * (t_{ij} * treat_i) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

```
knitr::include_graphics("/Users/tinglu/Bayesian/toe2.png",error = F)
```

$$\begin{pmatrix} \beta_{0i} \\ \beta_{1i} \end{pmatrix} | \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \boldsymbol{\Sigma} \overset{iid}{\sim} N_2\left(\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \boldsymbol{\Sigma}\right) \qquad i = 1, \ldots, 298$$

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \sim N_2\left(\begin{pmatrix} 2.5 \\ 0.6 \end{pmatrix}, 100 \times \mathbf{I}_2\right)$$

$$\boldsymbol{\Sigma} \sim \text{Inverse Wishart}(4, 0.1 \times \mathbf{I}_2)$$

$$\beta_2 \sim N(0, 100)$$

$$\sigma^2 \sim \text{Inverse Gamma}(1, 3.7)$$

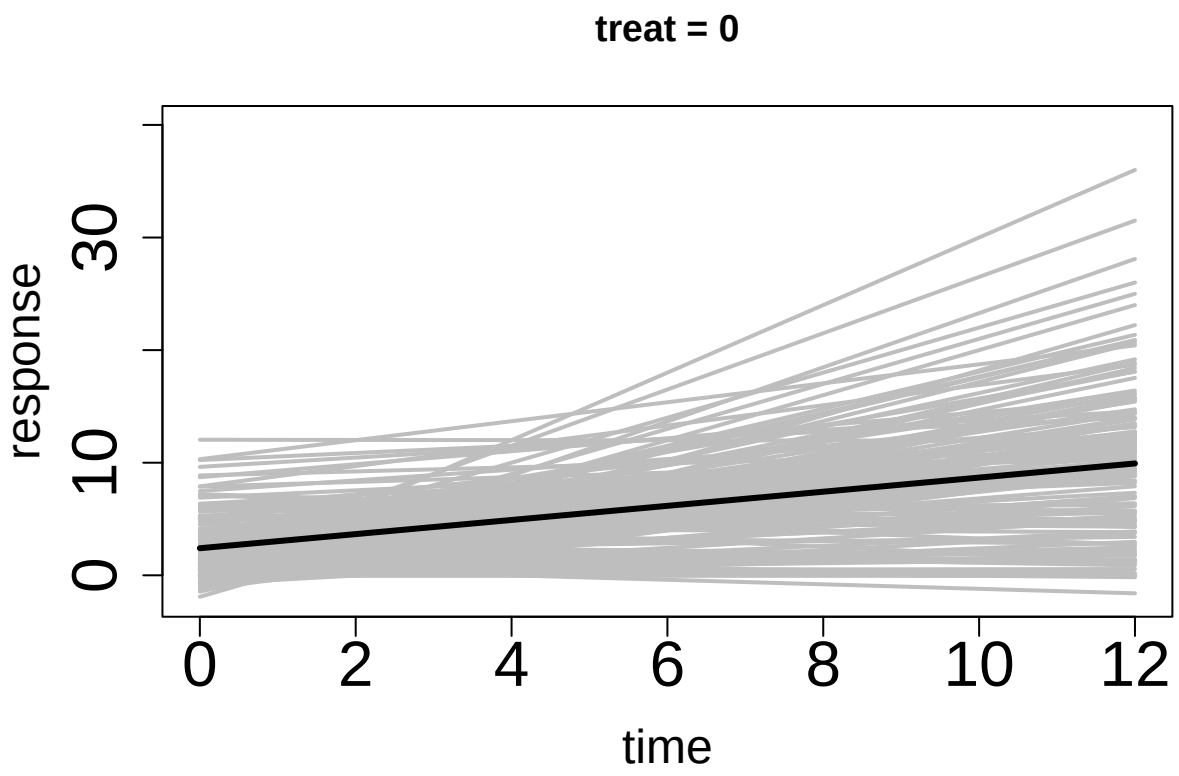
Before fitting the model in (2), conduct an exploratory analysis by fitting the OLS regression model separately for each patient i:

$$y_{ij} = \beta_{0i} + \beta_{1i} t_{ij} + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma_i^2)$$

Plot the estimated regression lines for each patient grouped by treatment (make two separate plots for the Itraconazol and the Lamisil group). In each plot also, draw the average evolution of toenail length over time by using the means of the estimated regression coefficients for all patients in each treatment group.Discuss what the plots reveal

```
# For Itraconazol group (treat = 0)
toe0 = toe[toe$treat == 0,]
pat <- toe0$id
m <- length(unique(pat))
y <- toe0$response
n <- length(y)
d <- toe0$time
p <- 2
X <- matrix(cbind(rep(1,n),d),nrow=n,ncol=p)

## determining the number of observations for each group(each patients)
n.j <- as.numeric(table(pat))
## OLS estimates of regression coefficients
beta.ols <- matrix(0,nrow=m,ncol=p)
var.pat <- rep(0,m)
sigma2 <- NULL
for(j in 1:m){
    y.j <- y[which(pat==unique(pat)[j])]
    d.j <- d[which(pat==unique(pat)[j])]
    reg.j <- lm(y.j~d.j)
    beta.ols[j,] <- as.numeric(reg.j$coeff)
    var.pat[j] <- var(y.j)
    sigma2[j] <- deviance(reg.j)/reg.j$df.residual
}
```

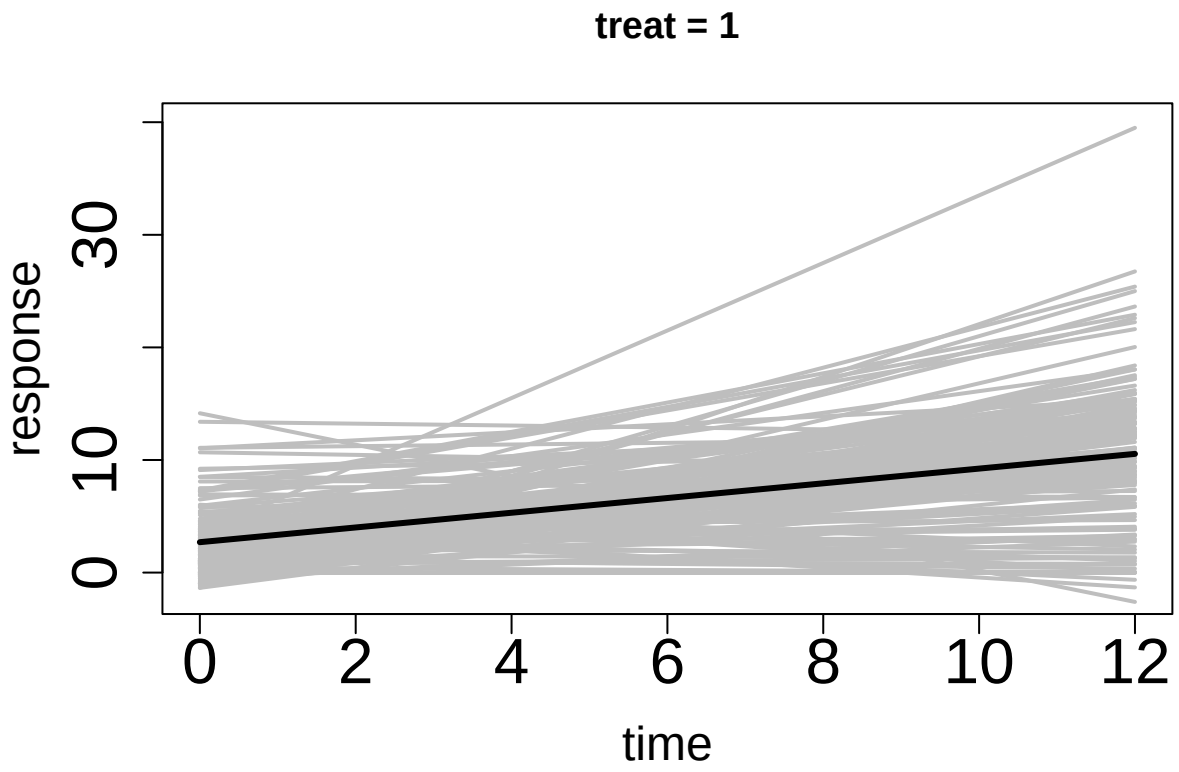
```
d_index = seq(0,12, length.out = 12)
plot(d_index,beta.ols[1,1]+beta.ols[1,2]*d_index,xlab="time",ylab="response",col="grey",type="l",lwd=2,

for(j in 2:m){
 lines(d_index,beta.ols[j,1]+beta.ols[j,2]*d_index,col="grey",lwd=2)
}
lines(d_index,mean(beta.ols[,1],na.rm = T)+mean(beta.ols[,2],na.rm = T)*d_index,col="black",lwd=3)
```



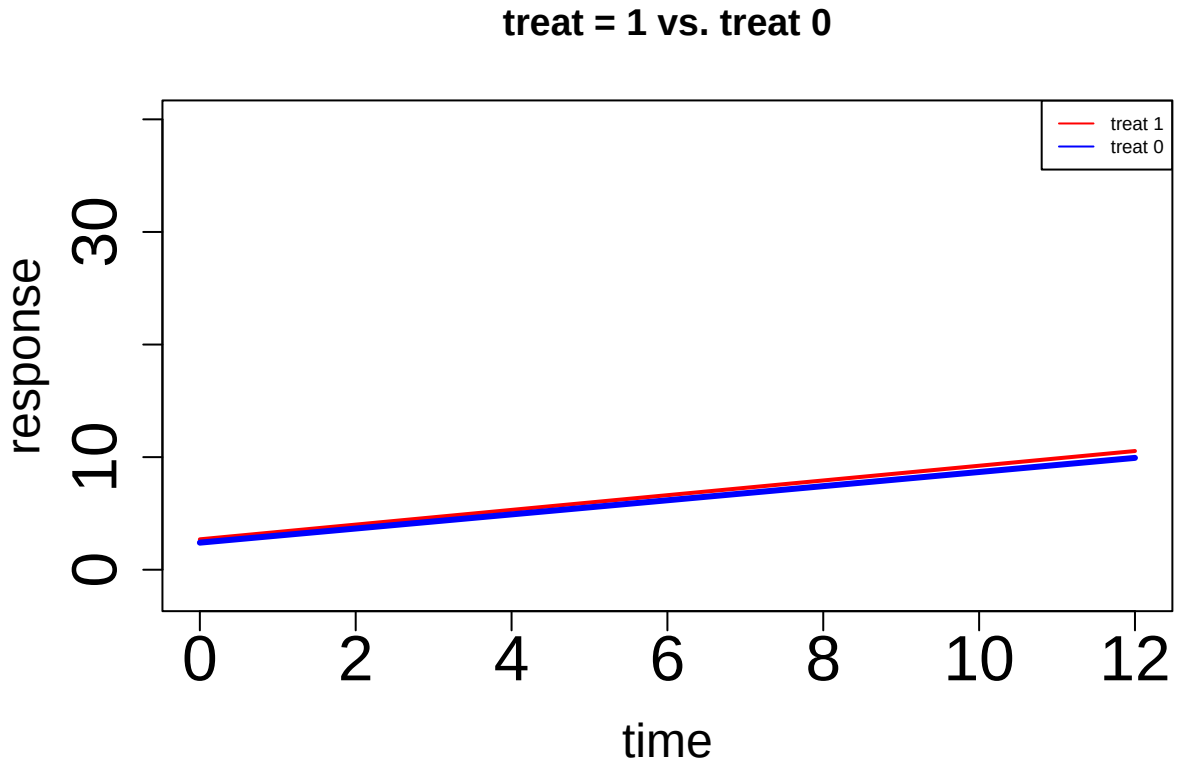

```
beta.ols0 <- beta.ols
```

```
# For Lamisil group (treat = 1)
toe1 = toe[toe$treat == 1,]
pat <- toe1$id
m <- length(unique(pat))
y <- toe1$response
n <- length(y)
d <- toe1$time
p <- 2
X <- matrix(cbind(rep(1,n),d),nrow=n,ncol=p)

## determining the number of observations for each group(each patients)
n.j <- as.numeric(table(pat))
## OLS estimates of regression coefficients
beta.ols <- matrix(0,nrow=m,ncol=p)
var.pat <- rep(0,m)
sigma2 <- NULL
for(j in 1:m){
    y.j <- y[which(pat==unique(pat)[j])]
```

```
    d.j <- d[which(pat==unique(pat)[j])]
    reg.j <- lm(y.j~d.j)
    beta.ols[j,] <- as.numeric(reg.j$coeff)
    var.pat[j] <- var(y.j)
    sigma2[j] <- deviance(reg.j)/reg.j$df.residual
}
d_index = seq(0,12, length.out = 12)
plot(d_index,beta.ols[1,1]+beta.ols[1,2]*d_index,xlab="time",ylab="response",col="grey",type="l",lwd=2,

for(j in 2:m){
 lines(d_index,beta.ols[j,1]+beta.ols[j,2]*d_index,col="grey",lwd=2)
}
lines(d_index,mean(beta.ols[,1],na.rm = T)+mean(beta.ols[,2],na.rm = T)*d_index,col="black",lwd=3)
```




```
plot(d_index,mean(beta.ols[,1],na.rm = T)+mean(beta.ols[,2],na.rm = T)*d_index,xlab="time",ylab="respons
lines(d_index,mean(beta.ols0[,1],na.rm = T)+mean(beta.ols0[,2],na.rm = T)*d_index,col="blue",lwd=3)
legend("topright", legend = c("treat 1","treat 0"), col = c("red","blue"),lty=1,cex=0.6)
```



In either the Itraconazol($treat = 0$) or the Lamisil($treat = 1$) group, the average evolution of toenail length over time grows. Regression line of the Lamisil group is slightly higher than that of Itraconazol group.

In either the Itraconazol($treat = 0$) or the Lamisil($treat = 1$) group, each individual has different regression line, indicating much variability. For most of individuals, evolution of toenail length over time grows. However, for some other individuals, evolution of toenail length over time decreases.

(c)

Fit the hierarchical linear mixed model (hierarchical regression) with random intercept and random slope.

$$y_{ij} = \beta_{0i} + \beta_{1i} * t_{ij} + \beta_2 * (t_{ij} * treat_i) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

Use prior information below:

```
knitr::include_graphics("/Users/tinglu/Bayesian/toe3.png",error = F)
```

$$\begin{pmatrix} \beta_{0i} \\ \beta_{1i} \end{pmatrix} | \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \Sigma \overset{iid}{\sim} N_2\left(\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \Sigma\right) \qquad i = 1, \ldots, 298$$

$$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \sim N_2\left(\begin{pmatrix} 2.5 \\ 0.6 \end{pmatrix}, 100 \times \mathbf{I}_2\right)$$

$$\Sigma \sim \text{Inverse Wishart}(4, 0.1 \times \mathrm{I}_2)$$

$$\beta_2 \sim N(0, 100)$$

$$\sigma^2 \sim \text{Inverse Gamma}(1, 3.7)$$

```
# Normal, for beta
mu0 <- c(2.5, 0.6, 0)
Sigma0 <- diag(rep(100, 3))
init.mu <- mu0
init.Sigma <- Sigma0
# Inverse Gamma, for residuals
nu0 <- 1
delta0 <- 3.7
# Inverse Wishart, for random effect
r <- 4
R <- 0.1* diag(rep(1, 2))/r
toe$time.treat <- toe$time*toe$treat
```

```
model <- MCMChregress(fixed = response ~ time + time.treat,
                      random = ~time,
                      group = "id",
                      data = toe,
                      burnin = 1000, mcmc = 10000,
                      thin = 1, verbose = 1, seed = 0,
                      mubeta = init.mu, Vbeta = init.Sigma,
                      r = r, R = R,
                      nu = nu0, delta0 = 3.7)
```

```
##
## Running the Gibbs sampler. It may be long, keep cool :)
##
## **********:10.0%
## **********:20.0%
## **********:30.0%
## **********:40.0%
## **********:50.0%
```

```
## **********:60.0%
## **********:70.0%
## **********:80.0%
## **********:90.0%
## **********:100.0%
```

Effect sample size of all parameters is large than 1000

```
dim(model$mcmc) # contains all posterior samples
```

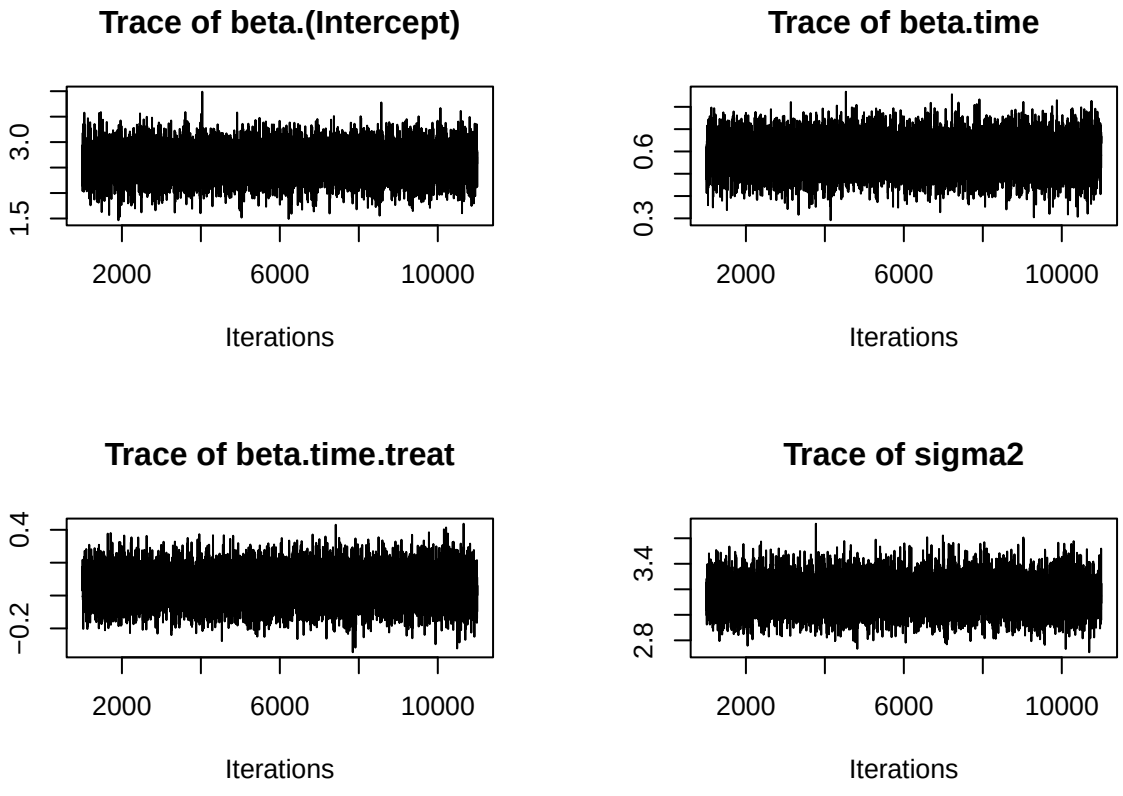
```
## [1] 10000   605
```

```
sum(sapply(1:605, function(i) effectiveSize(model$mcmc[,i]))<=1000)
```

```
## [1] 0
```

Assess convergence for $\beta_0, \beta_1, \beta_2, \sigma^2$, and Traceplot show the Markov chain is converged.

```
par(mfrow = c(2,2))
traceplot(model$mcmc[, c(1:3, 604)], smooth = TRUE)
```




```
post.mean <-apply(model$mcmc[, c(1:3, 604)][1000:10000, ], 2, mean)
post.quantile<-apply(model$mcmc[, c(1:3, 604)][1000:10000, ], 2, quantile, c(0.025, 0.975))
mixed <- rbind(post.mean,post.quantile)
mixed
```

```
##           beta.(Intercept) beta.time beta.time.treat   sigma2
## post.mean         2.596517 0.5823375      0.05661579 3.155056
## 2.5%              2.002294 0.4319638     -0.14672889 2.915619
## 97.5%             3.188220 0.7329455      0.25761339 3.411383
```

95% credible interval of β_1 (time) don't include 0, which means average unaffected toenail length grows over time.

95% credible interval of β_2 (time*treat) include 0, which means there is no significant difference between these two treatment groups.

(d)

Compare the fixed effect model vs. the mixed effect model, is there variability among patients in the way their unaffected toenail grows over time?

```
fix
```

```
##                   [,1]      [,2]       [,3]     [,4]
## post.mean.fix 2.664289 0.5609139 0.06346246 13.55644
## 2.5%          2.423144 0.5096115 0.00667572 12.70317
## 97.5%         2.909417 0.6125981 0.12015786 14.45544
```

```
mixed
```

```
##           beta.(Intercept) beta.time beta.time.treat   sigma2
## post.mean         2.596517 0.5823375      0.05661579 3.155056
## 2.5%              2.002294 0.4319638     -0.14672889 2.915619
## 97.5%             3.188220 0.7329455      0.25761339 3.411383
```

Compare the σ^2 in the the fixed effect model vs. the mixed effect model, it is obvious that there is variability among patients in the way their unaffected toenail grows over time because $\sigma^2_{fix} >> \sigma^2_{mixed}$

(e)

Suppose two new patients are enrolled in the study, one taking Itraconzol and the other one taking Lamisil. Predict how long their unaffected toe nail will be at 0, 1, 2, 3, 6, 9 and 12 months since enrollment in the study and provide 95% confidence bands around your predictions (provide a plot of the predictions and the 95% confidence bands for for both patients).

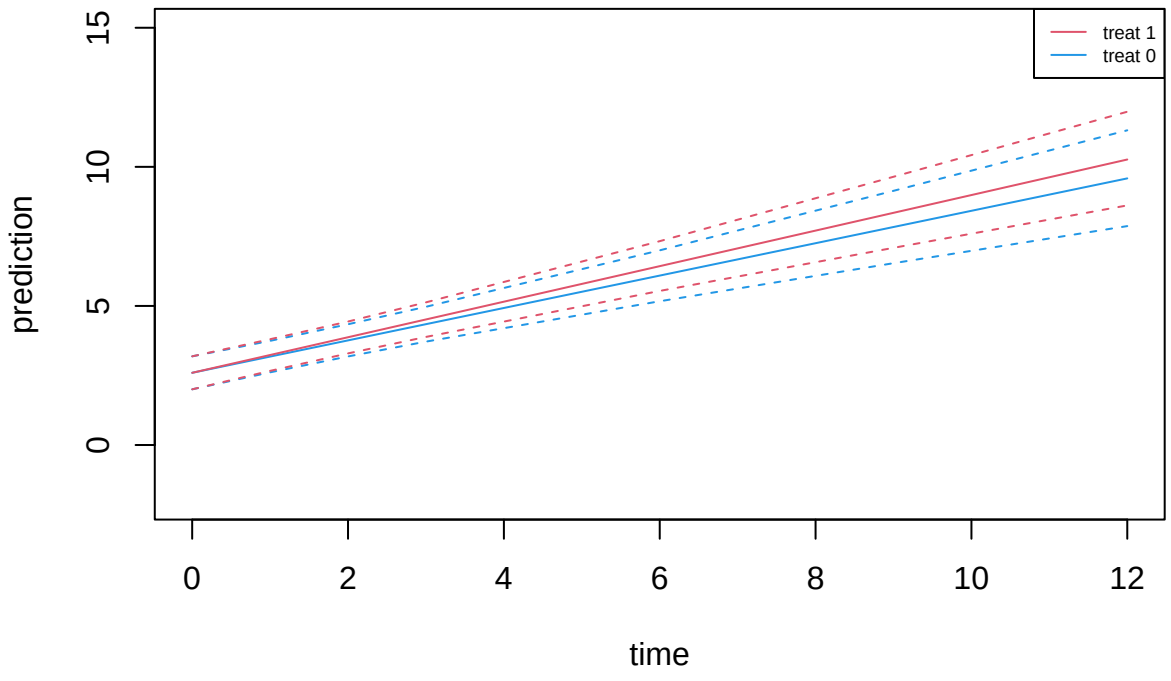
```
beta <- model$mcmc[1000:10000, 1:3]
ti <- c(0,1,2,3,6,9,12)
X0 <- cbind(1,ti,ti*0)
X1 <- cbind(1,ti,ti*1)
pred0 <- X0 %*% t(beta)
pred1 <- X1 %*% t(beta)
pred0.mean <- apply(pred0, 1, mean)
pred0.quantile <- apply(pred0, 1, quantile, c(0.025, 0.975))
pred1.mean <- apply(pred1, 1, mean)
pred1.quantile <- apply(pred1, 1, quantile, c(0.025, 0.975))
df0 = data.frame(time = ti,low = pred0.quantile[1, ],mean = pred0.mean, high = pred0.quantile[2, ])
df1 = data.frame(time = ti,low = pred1.quantile[1, ],mean = pred1.mean, high = pred1.quantile[2, ])
# prediction for treat 0
df0
```

```
##   time      low     mean      high
## 1    0 2.002294 2.596517  3.188220
## 2    1 2.612615 3.178854  3.745270
## 3    2 3.183807 3.761192  4.341042
## 4    3 3.717252 4.343529  4.969401
## 5    6 5.168475 6.090542  6.998649
## 6    9 6.536127 7.837554  9.137973
## 7   12 7.869733 9.584566 11.313601
```

```
# prediction for treat 1
df1
```

```
##   time      low      mean      high
## 1    0 2.002294  2.596517  3.188220
## 2    1 2.665828  3.235470  3.803099
## 3    2 3.293756  3.874423  4.439218
## 4    3 3.885706  4.513377  5.127893
## 5    6 5.539844  6.430236  7.328427
## 6    9 7.088695  8.347096  9.643622
## 7   12 8.614097 10.263956 11.981705
```

```
plot(df0$time,df0$mean,type= 'l',xlab = 'time', ylab="prediction",ylim=c(-2,15),col = "4")
lines(df0$time,df0$low,type = 'l', lty = 2,col = '4')
lines(df0$time,df0$high,type = 'l',lty = 2,col = '4')

lines(df1$time,df1$mean,type = 'l', lty = 1,col = '2')
lines(df1$time,df1$high,type = 'l',lty = 2,col = '2')
lines(df1$time,df1$low,type = 'l', lty = 2,col = '2')

legend("topright", legend = c("treat 1","treat 0"), col = c(2,4),lty=1,cex=0.6)
```



3. hierarchical model (poisson)

```
knitr::include_graphics("/Users/tinglu/Bayesian/poi.png",error = F)
```

- $Y_i|\theta_i, x_i \sim \text{Poisson}(\theta_i x_i)$;
- $\theta_1, \ldots, \theta_6|a, b \sim \text{gamma}(a, b)$;
- $a \sim \text{gamma(1,1)}$; $b \sim \text{gamma(10,1)}$.

(a)

Y_i: the number of occurrences of disease for 6 neighboring counties, therefore, y_i/x_i represents the observed disease rate.

θ_i: represents the expected disease rate where $E(\theta_i) = a/b$

a and b are the parameters of the gamma distribution sampling model of θ_i.

(b)

Conditional distribution of $p(\theta_1, ..., \theta_6|a, b, x, y)$:

$$p(\theta_1, ..., \theta_6|a, b, x, y) = \frac{p(\theta_1, ..., \theta_6, y|a, b, x)}{p(y|a, b, x)}$$

$$\begin{aligned} p(\theta_1, ..., \theta_6, y|a, b, x) &= p(\theta_1, ..., \theta_6|a, b) * p(y|\theta_1, ..., \theta_6, x) \\ &= \prod_{i=1}^{6} \frac{(\theta_i x_i)^{y_i} exp(-\theta_i x_i)}{y_i!} * \prod_{i=1}^{6} \frac{(b)^a}{\Gamma(a)} \theta_i^{a-1} exp(-b\theta_i) \\ &= f(y) \prod_{i-1}^{6} \theta_i^{a+y_i-1} exp(-(b + x_i)\theta_i) \\ &\propto \prod_{i-1}^{6} \theta_i^{a+y_i-1} exp(-(b + x_i)\theta_i) \end{aligned}$$

θ_i are conditional independent given a, b, x, y:

$$p(\theta_i|\theta_{-i}, a, b, x, y) \propto \prod_{i-1}^{6} \theta_i^{a+y_i-1} exp(-(b + x_i)\theta_i) \sim gamma(a + y_i, b + x_i)$$

(c)

Write out the ratio of the posterior densities comparing a set of proposal values (a^*, b^*, θ) to values (a, b, θ). Note the value of θ, the vector of county-specific rates, is unchanged.

$$
\begin{aligned}
r &= \frac{p(\theta_1, ..., \theta_6, y|a^*, b^*, x)}{p(\theta_1, ..., \theta_6, y|a, b, x)} \\
&= \prod_{i=1}^{6} \frac{\theta_i^{a^*+y_i-1} exp(-(b^*+x_i)\theta_i)}{\theta_i^{a+y_i-1} exp(-(b+x_i)\theta_i)} \\
&= \prod_{i-1}^{6} \theta_i^{a^*-a} exp(-(b^*-b)\theta_i)
\end{aligned}
$$

(d)

Construct a Metropolis-Hastings Algorithm which generates samples of (a, b, θ) from the posterior.

```
x <- c(33, 14, 27, 90, 12, 17)
y <- c(1, 3, 2, 12, 1, 1)
a <- 1
b <- 10
theta <- rgamma(6, a + y, b + x)
a_delta <- 1
b_delta <- 10
ratio <- function(a, b, a_new, b_new, theta) {
  prod(theta^(a_new - a) * exp(-(b_new - b) * theta))
}
S <- 10000
set.seed(123)
THETA <- NULL
for (j in 1:S) {
  if (j%%(S/10) == 0) {
    print(j)
    }
  a_new <- abs(runif(1, a - a_delta, a + a_delta))
  b_new <- abs(runif(1, b - b_delta, b + b_delta))
  r <- ratio(a, b, a_new, b_new, theta) #compute the acceptance ratio
  if (runif(1) < r) {
    # accept
    theta <- rgamma(6, a_new + y, b_new + x)
    THETA <- rbind(THETA, c(a_new, b_new, theta))
    a <- a_new
    b <- b_new
    } else {
      theta <- rgamma(6, a + y, b + x)
      THETA <- rbind(THETA, c(a, b, theta))
    }
}
```

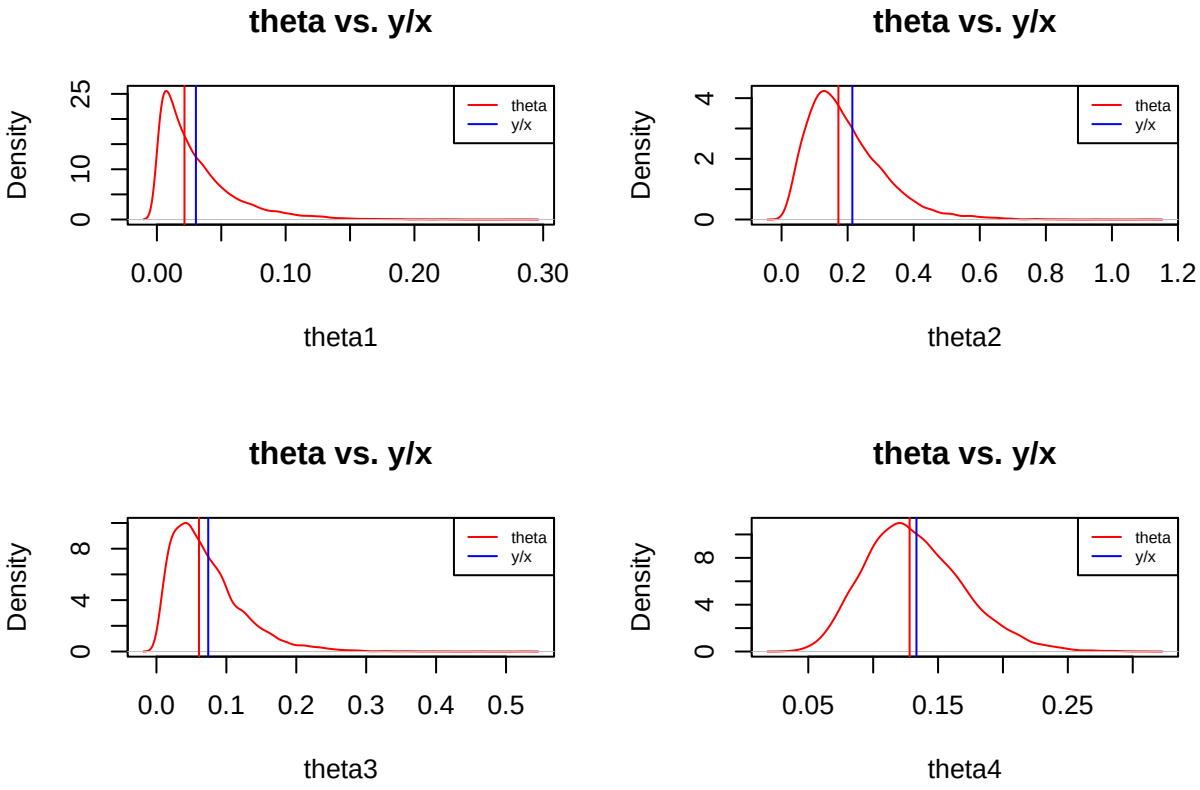
```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7000
## [1] 8000
```

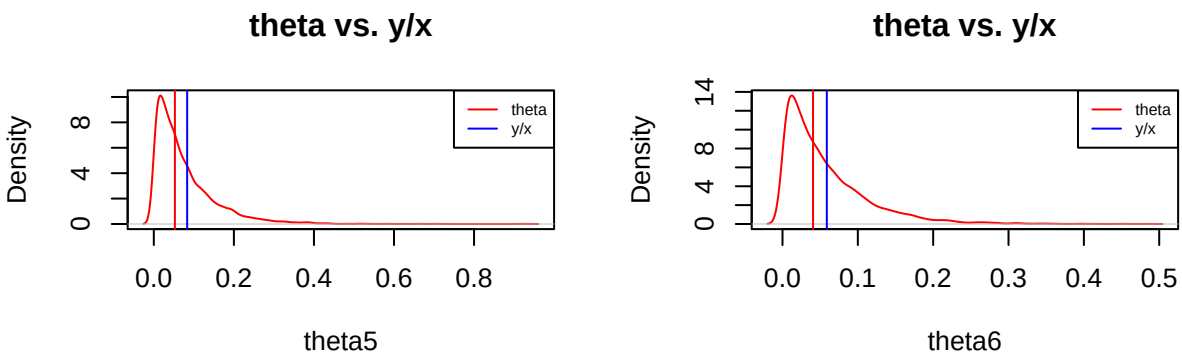
```
## [1] 9000
## [1] 10000
```

i

Compute marginal posterior distributions of $\theta_1, ..., \theta_6$ and compare them to $y_1/x_1, ..., y_6/x_6$.

```
burn_in <- 1000
theta.post <- THETA[, -(1:2)]
y.x <- y/x

par(mfrow = c(2,2))

for (j in 1:6) {
  plot(density(theta.post[,j]), xlab = paste0("theta", j),main = paste0("theta vs. y/x"),
       col='red')
  abline(v = y.x[j],col = 'blue')
  abline(v = median(theta.post[,j]), col = "red")
  legend('topright', legend=c("theta", "y/x"),
       col=c("red", "blue"), lty=1,cex=0.6)

}
```





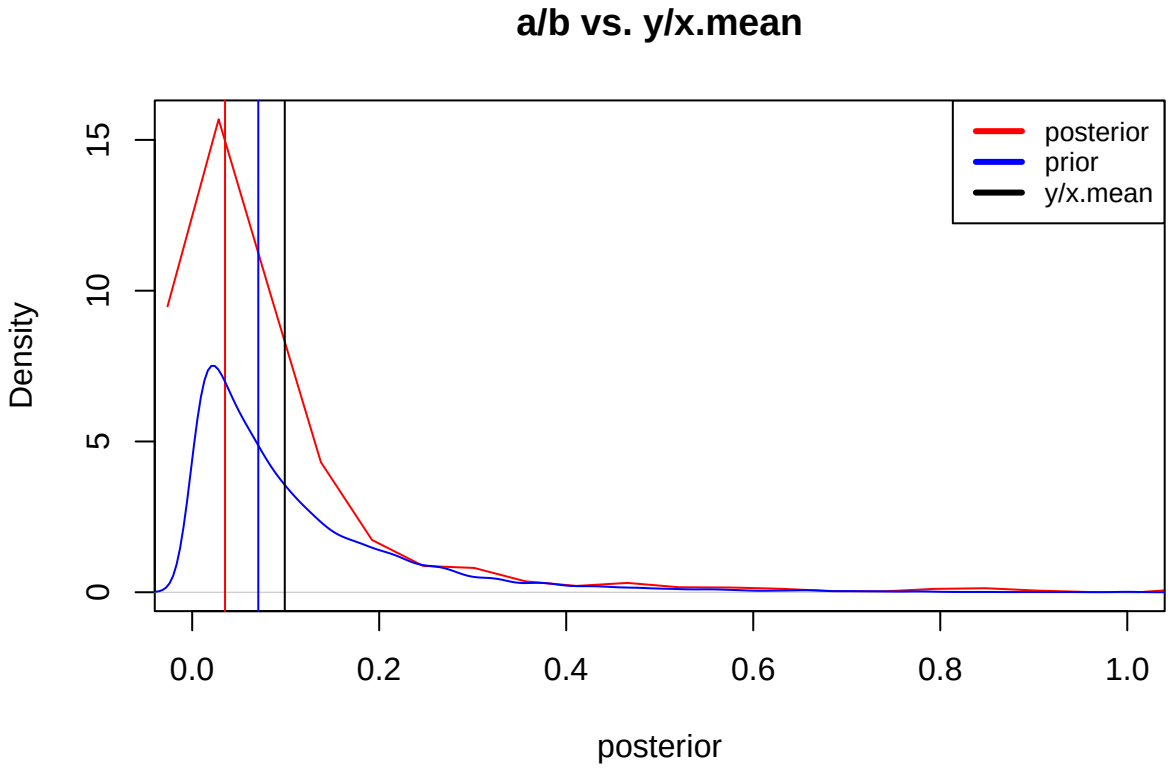
Based on the plot, we can discover that y_i/x_i is larger than the median value of posterior distribution of $\theta_1, ...\theta_6$.

ii

Examine the posterior distribution of a/b, and compare it to the corresponding prior distribution as well as to the average of y_i/x_i across the six counties.

$$E(\theta_i) = a/b$$

```
a.b <- THETA[burn_in:S, 1]/THETA[burn_in:S,2]
a.b.prior <- rgamma(S-burn_in + 1, 1, 1)/rgamma(S-burn_in + 1,10, 1)
y.x <- mean(y/x)

plot(density(a.b), xlab = "posterior",main = "a/b vs. y/x.mean", col='red',xlim=c(0,1))
abline(v = median(a.b),col = "red")
abline(v = median(a.b.prior),col="blue")
lines(density(a.b.prior),col = 'blue')
abline(v = y.x, col = "black")
legend('topright', legend=c("posterior", "prior","y/x.mean"),
       col=c("red", "blue","black"), lty=1,cex=0.8,lwd = 3)
```

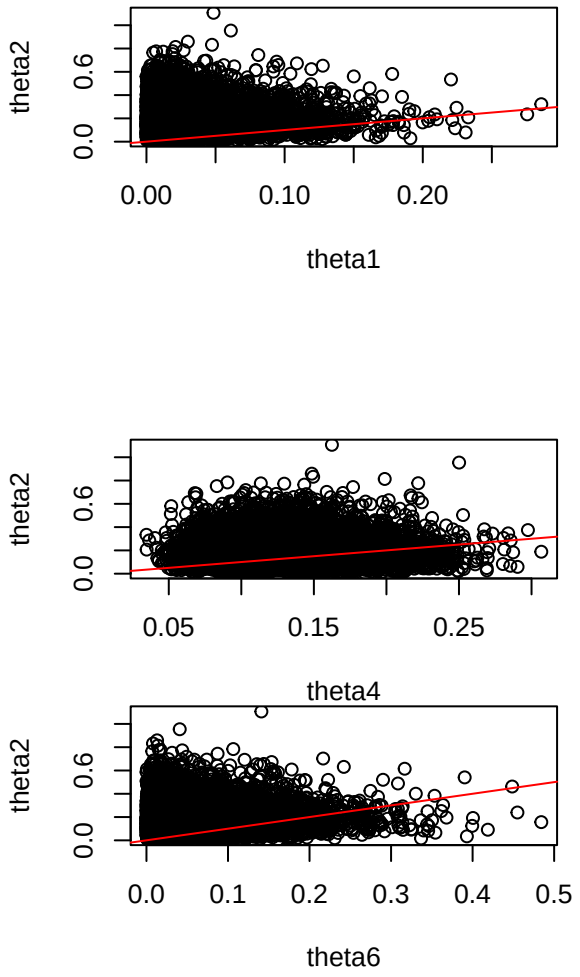


Based on the plot, we can discover that average of y_i/x_i is larger than the median value of both posterior and prior distribution of a/b

iii

Plot samples of θ_2 versus θ_j for each $j \neq 2$, and draw a 45 degree line on the plot as well. Also estimate $Pr(\theta_2 > \theta_j|x, y)$ for each j and $Pr(\theta_2 = max\{\theta_1, ..., \theta_6\}|x, y)$. Interpret the results of these calculations, and compare them to the conclusions one might obtain if they just examined yj/xj for each county j.

```
par(mfrow=c(2,2))
for (j in setdiff(1:6, 2)) {
  plot(theta.post[, j], theta.post[, 2], xlab = paste0("theta",j),
       ylab = "theta2")
  abline(a = 0, b = 1, col = "red")
}
```



$$Pr(\theta_2 > \theta_j | x, y)$$

:

```
sapply(setdiff(1:6, 2), function(j) {
  mean(theta.post[, 2] > theta.post[,j])
})
```

```
## [1] 0.9665 0.8697 0.6656 0.8448 0.9030
```

```
mean(theta.post[, 2] == apply(theta.post, 1, max))
```

```
## [1] 0.5783
```

```
y/x
```

```
## [1] 0.03030303 0.21428571 0.07407407 0.13333333 0.08333333 0.05882353
```

$Pr(\theta_2 = max\{\theta_1, ..., \theta_6\} | x, y) = 0.58$

$y_2/x_2 = 0.21$ which is the highest disease rate among six counties. Data information shows the second county has the highest disease rate. Also, by Bayesian analysis, posterior estimates show that the second county has high probability to be the county with the largest disease rate.