Study 1. Cholesterol Intake Variability

Ting Lu

2023-10-14

Introduction

There is an increasing awareness that we should improve our life style. In Western Europe, a variety of campaigns have been set up in the last decades to give up smoking and to render our diet more healthy, for example by lowering our daily consumption of saturated fat. Around 1990, a dietary survey, the **Interregional Belgian Bank Employee Nutrition Study (IBBENS)** was set up to compare the dietary intake in different geographical areas in Belgium, especially in Flanders. The IBBENS study was performed in eight subsidiaries of one bank situated in seven Dutch-speaking cities in the north and in one French-speaking city in the south of Belgium.

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`

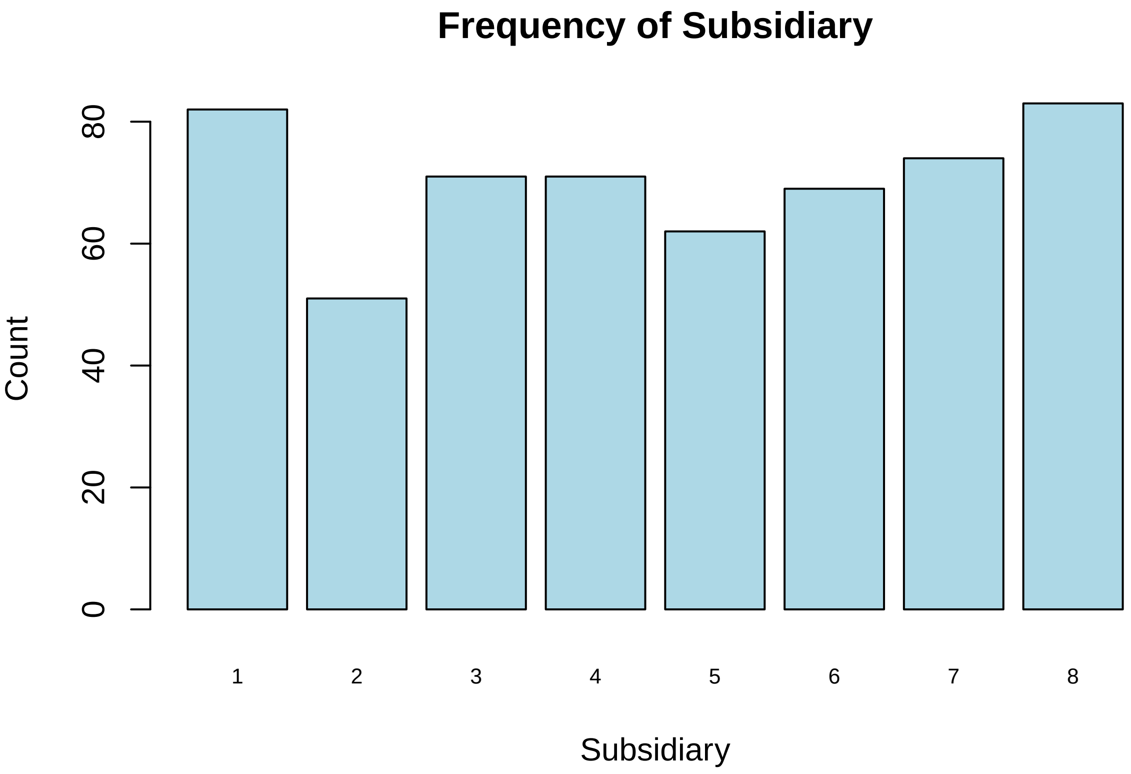
Loading data

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.

```
IBB = read.csv("~/Bayesian/Study1-Cholesterol-Intake-Variability/IBBENS.csv")
head(IBB)
```

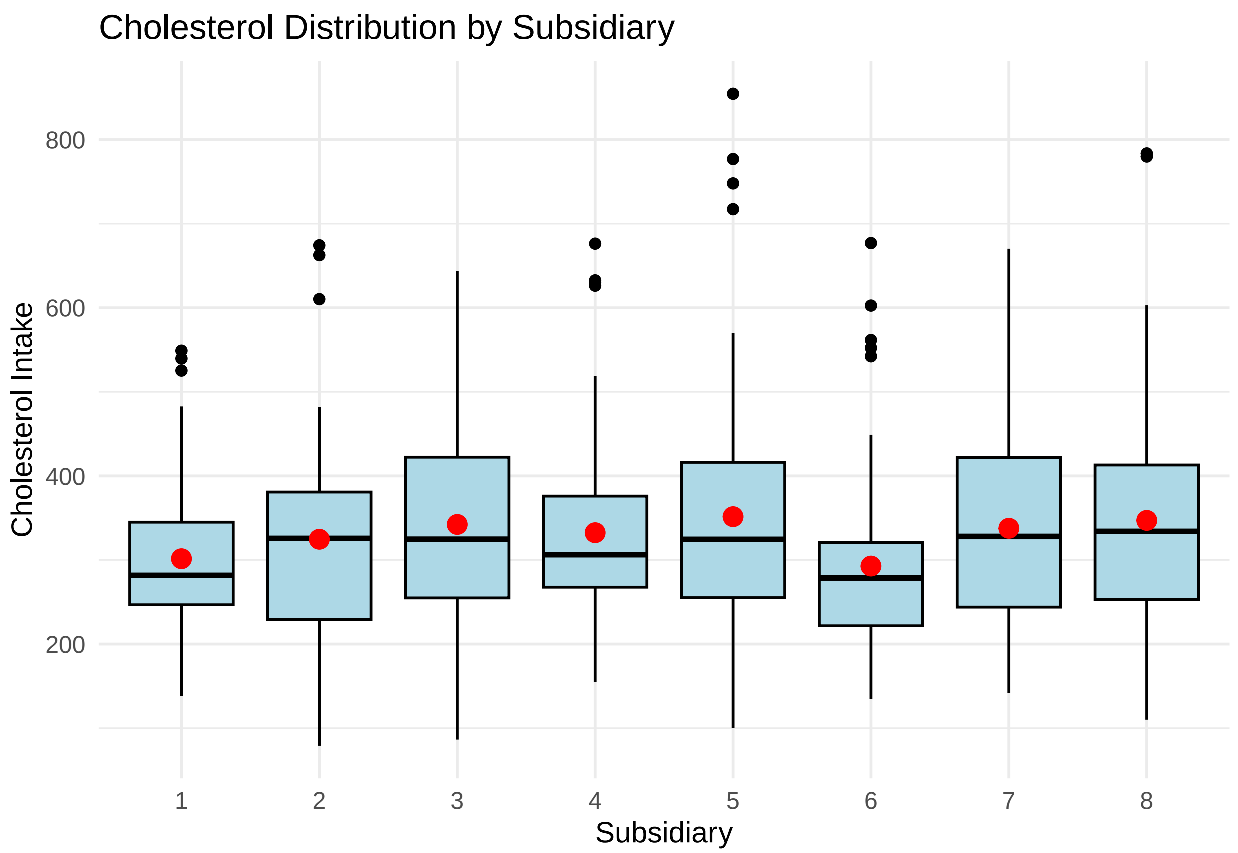
```
##     chol subsidiary
## 1 395.33          1
## 2 449.00          1
## 3 278.33          1
## 4 254.00          1
## 5 408.67          1
## 6 539.67          1
```

```
subsidiary_counts <- table(IBB$subsidiary)
barplot(subsidiary_counts,
        col = "lightblue",
        border = "black",
        names.arg = names(subsidiary_counts),
        cex.names = 0.7,
        xlab = "Subsidiary",
        ylab = "Count",
        main = "Frequency of Subsidiary")
```



EDA (Cholesterol Distribution by Subsidiary)

```
IBB$subsidiary_c <- as.character(IBB$subsidiary)

ggplot(IBB, aes(x = subsidiary_c, y = chol)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  stat_summary(fun = "mean", geom = "point", size = 3, color = "red")+
  xlab("Subsidiary") +
  ylab("Cholesterol Intake") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggtitle("Cholesterol Distribution by Subsidiary")+
  theme_minimal()
```



There is a significant difference in the distribution of daily cholesterol intake in some of the subsidiaries, so it is essential to study the variability among them in depth.

Goal & Methods

To contrast the variability of cholesterol intake (chol) in mg/day between the subsidiaries to the variability within the subsidiaries. To achieve this goal, I build a Bayesian hierarchical normal model as follows:

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

Above is a hierarchical normal model:

- In the first stage, y_{ij} is the cholesterol intake for subject $j = 1, ..., m_i$ in subsidiaries $i = 1, ..., n$. *ind* means "independent".

- In the second stage, θ_i is the average cholesterol intake for subsidiary i. *iid* means "independently and identically distributed".

- 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.

Prior setting

Setting the parameters of the inverse gamma distribution to 0.001, 0.001 implies a high degree of uncertainty regarding the variance in a Bayesian hierarchical normal model. I consider a wide range of possible variance values and have a low prior precision becaue I have little prior knowledge about the variance and rely heavily on the observed data to determine it. The choice emphasizes data-driven inference. (However, the specific prior setting should be tailored to the characteristics of the problem and available domain knowledge)

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

```r
# 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 MCMC sampling update function

```r
# 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){
```

```
# create a vector of length equal to the number of observations
# where 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 parameter value setting for MCMC sampling

```
# 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)

### 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
```

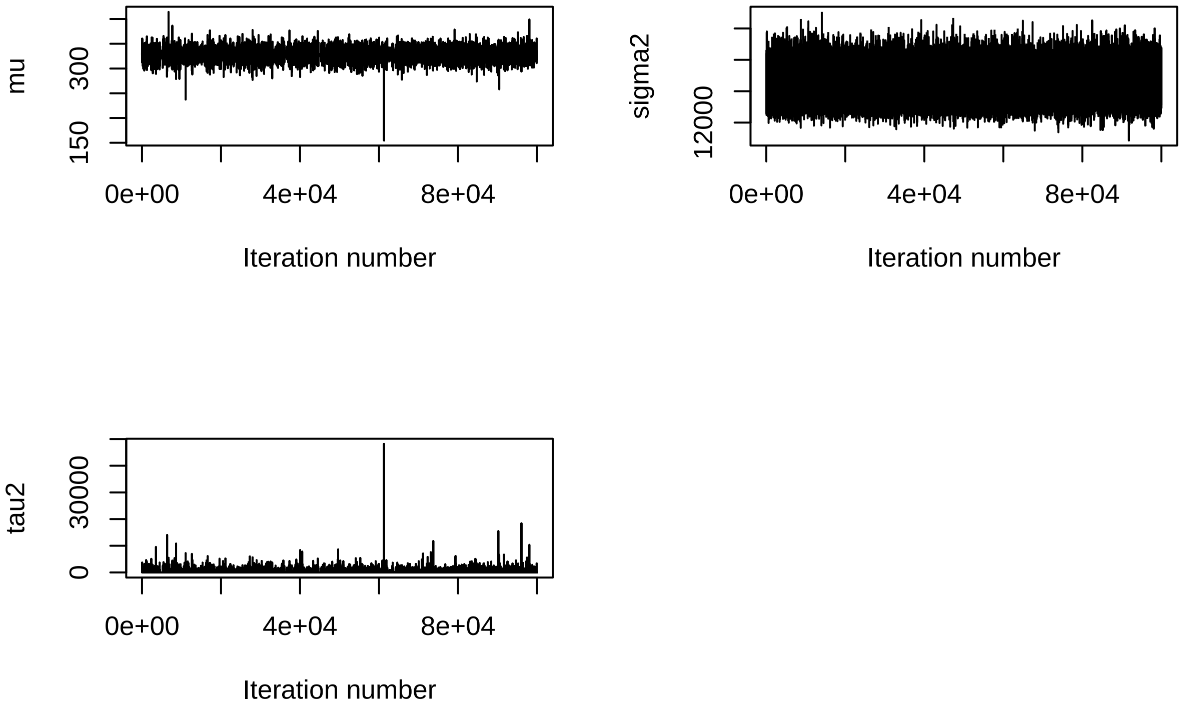
Run Gibbs sampling (one of MCMC sampling)

```
# Run Gibbs sampling (Make sure effective sample size of each parameters is at least 1000.)
S <- 100000

## store the subsidairy indicator specific parameters, the theta_js in one matrix
theta.MCMC <- matrix(0,nrow=S,ncol=m)
## 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) # set random seed.
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)
}
```

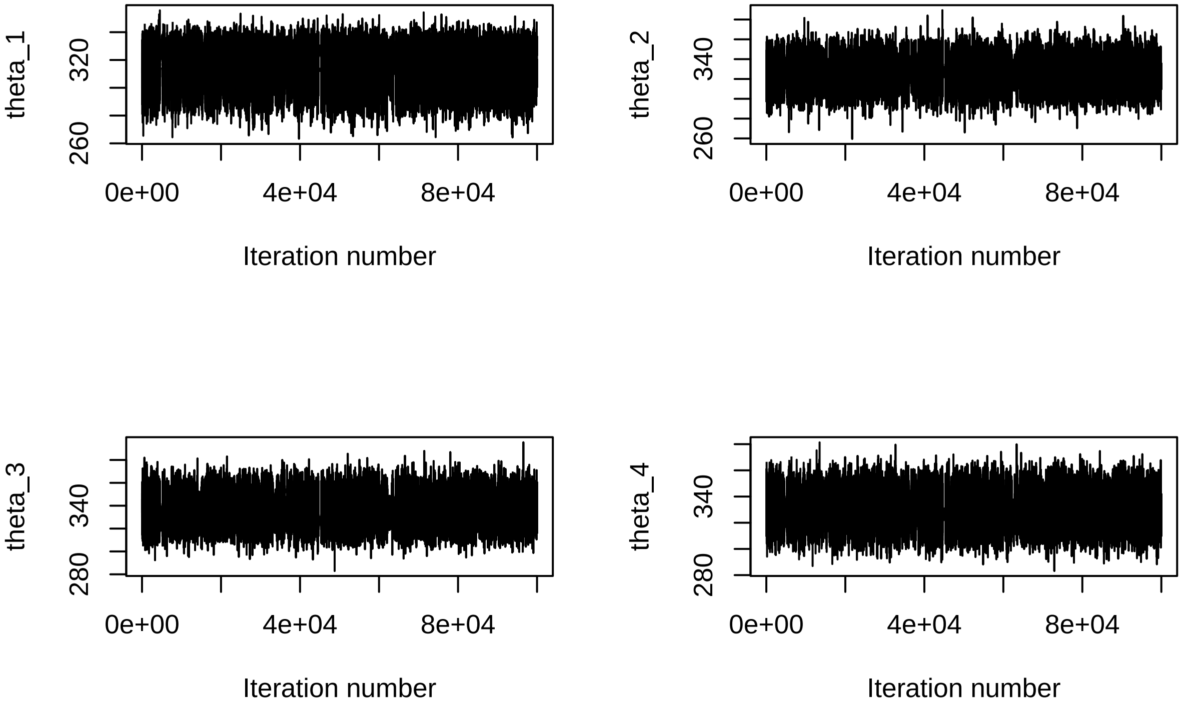
MCMC sampling diagnostic (trace plot, ACF plot, effective sampling size)

Trace plot

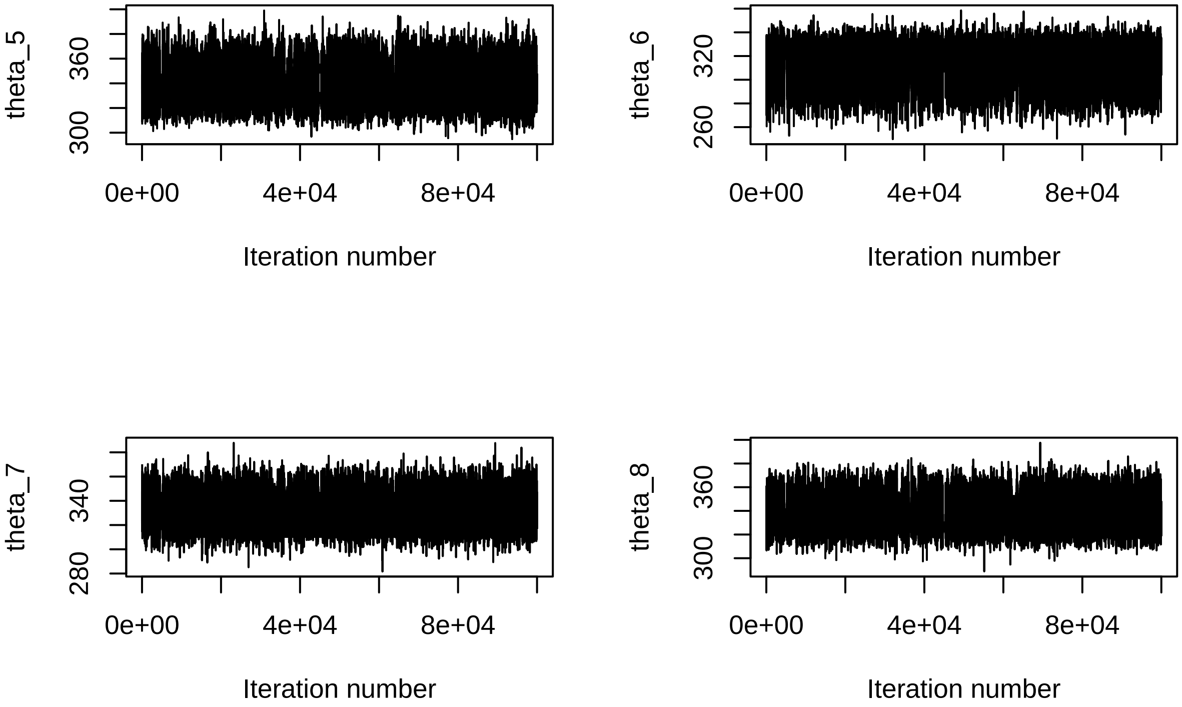
```
## Trace plots
par(mfrow=c(2,2))
plot(other.pars.MCMC[,1],xlab="Iteration number",ylab="mu",type="l")
plot(other.pars.MCMC[,2],xlab="Iteration number",ylab="sigma2",type="l")
plot(other.pars.MCMC[,3],xlab="Iteration number",ylab="tau2",type="l")

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")
```



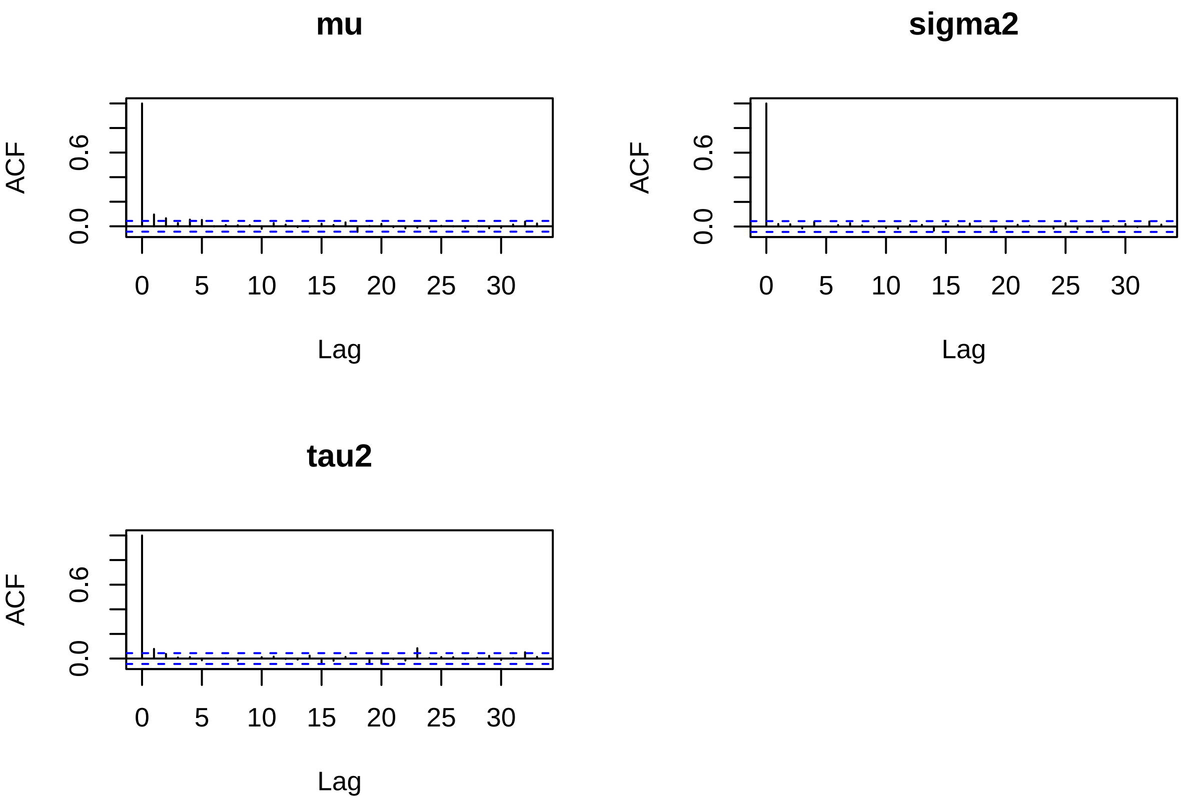
```r
plot(theta.MCMC[,5],xlab="Iteration number",ylab="theta_5",type="l")
plot(theta.MCMC[,6],xlab="Iteration number",ylab="theta_6",type="l")
plot(theta.MCMC[,7],xlab="Iteration number",ylab="theta_7",type="l")
plot(theta.MCMC[,8],xlab="Iteration number",ylab="theta_8",type="l")
```



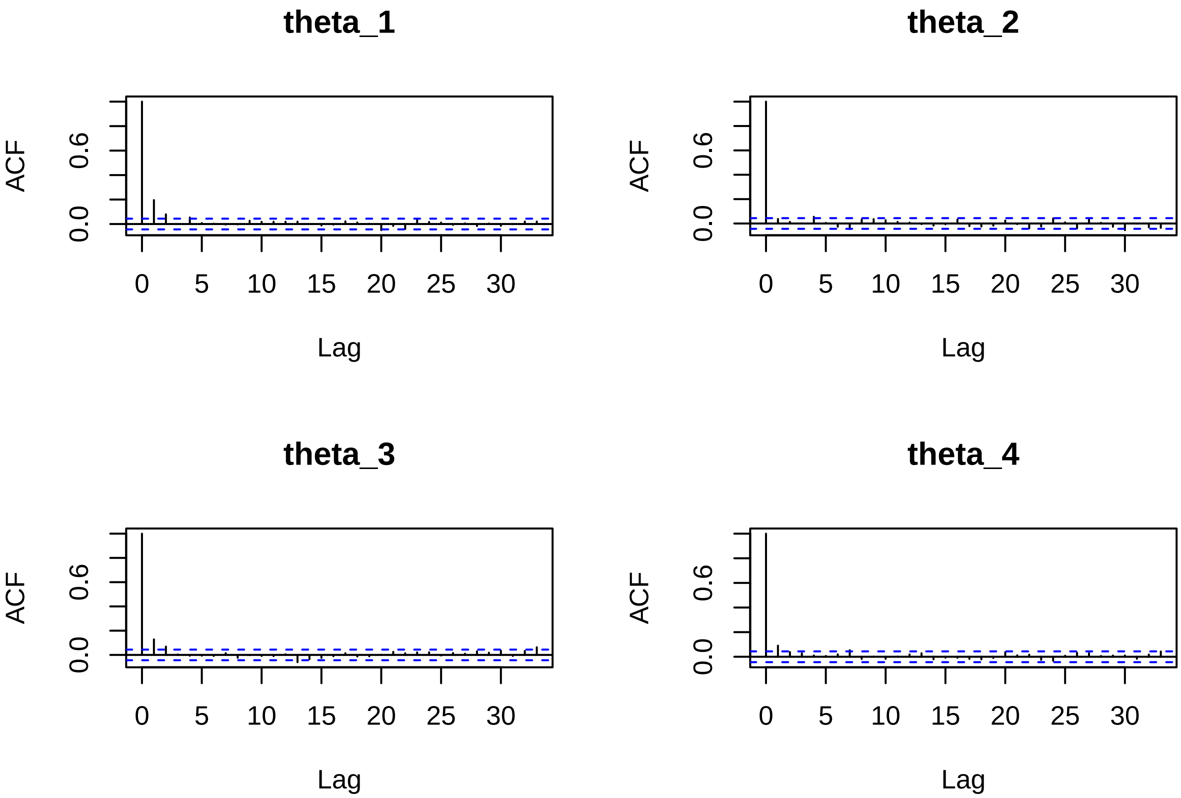
The trace plot shows fluctuations within a certain range without significant gaps, indicating that the MCMC sampling algorithm has achieved convergence during the sampling process.

ACF plot

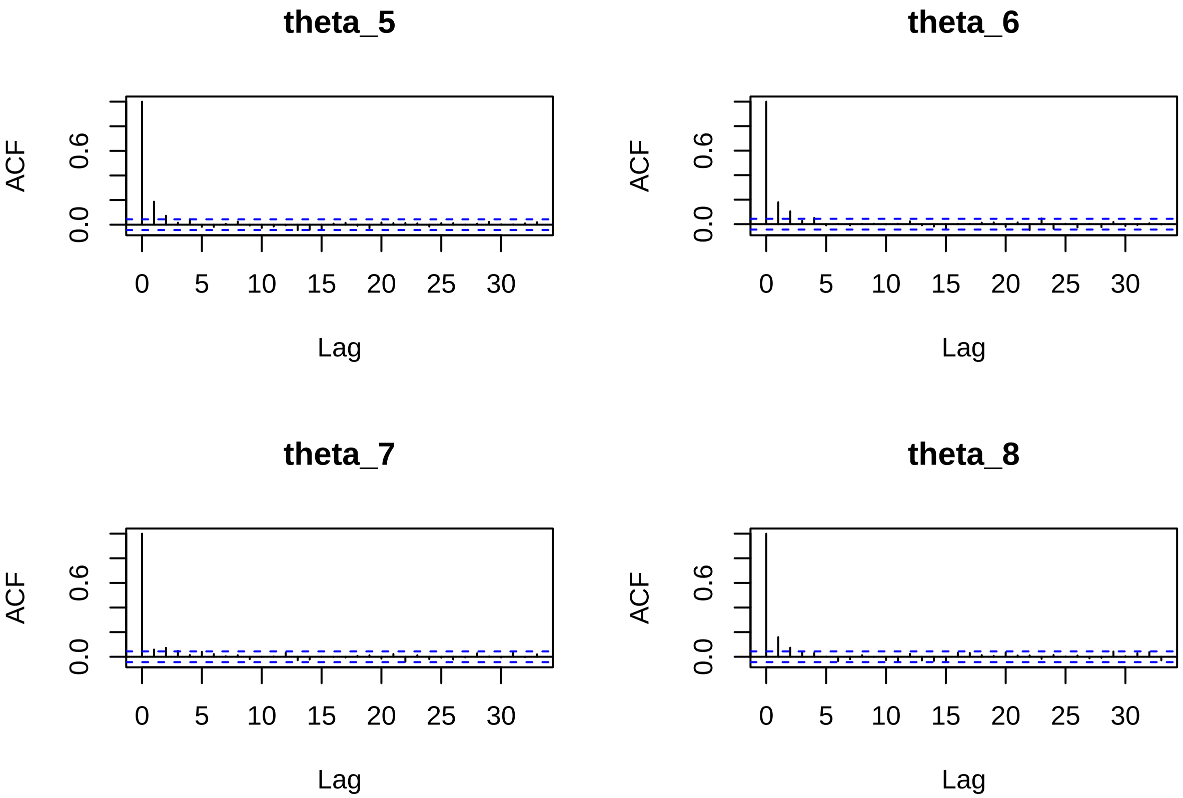
```
## ACF plots
thin = seq(1,S, by = 50)
par(mfrow=c(2,2))
acf(other.pars.MCMC[thin,1],main="mu")
acf(other.pars.MCMC[thin,2],main="sigma2")
acf(other.pars.MCMC[thin,3],main="tau2")

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")
```



The autocorrelation function (ACF) quickly decreases and approaches zero after a few lags, indicating that the MCMC chain has converged. This suggests that the correlation between parameter values has reduced, and the parameter values have stabilized, no longer being significantly influenced by their previous values.

```
## 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
```

In above MCMC sampling process, ESS for each parameter is larger than 1000. Effective Sample Size (ESS) is a metric used to assess the quality of an MCMC chain, measuring the independence of information within the generated samples. A higher ESS (>1000) indicates that more samples provide independent information, increasing the reliability of estimates. ESS calculation takes into account autocorrelation and chain length, with higher ESS implying less correlated and more information-rich samples.

Posterior inference

Average cholesterol intake in each of the eight subsidiaries: posterior means and 95% credible intervals.

Set "burnin=5000" to bring the MCMC chain to a stable state, ensuring that the sampled parameter values are sufficiently close to the posterior distribution. In the early stages of MCMC sampling, the chain may be in an initial state that differs from the posterior distribution. This initial state is typically referred to as the "burn-in" period, and the samples collected during this period are usually discarded.

```
### Marginal posterior summaries
burnin <- 5000
## Theta
post.mean.theta <- apply(theta.MCMC[(burnin+1):S,],2,mean)
post.quantile.theta <- apply(theta.MCMC[(burnin+1):S,],2,quantile, probs = c(0.025, 0.975))
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
```

```
cat("Subsidiary with highest posterior probability of the highest cholesterol intake per day:", which.ma
```

```
## Subsidiary with highest posterior probability of the highest cholesterol intake per day: 5
```

```
cat("Subsidiary with lowest posterior probability of the highest cholesterol intake per day:", which.mi
```

```
## Subsidiary with lowest posterior probability of the highest cholesterol intake per day: 6
```

Population average level of cholesterol intake:

provide posterior mean and 95% credible interval.

```
# Mu
df<-data.frame(
  population.posterior.mean = mean(other.pars.MCMC[(burnin+1):S, 1]),
  Q2.5 = quantile(other.pars.MCMC[(burnin+1):S, 1], 0.025),
  Q97.5 = quantile(other.pars.MCMC[(burnin+1):S, 1], 0.975)
)
rownames(df) <- "cholesterol intake per day"
df
```

```
##                            population.posterior.mean    Q2.5    Q97.5
## cholesterol intake per day                  328.5337 314.492 342.8445
```

Compare “between-subsidiaries” variability and “within-subsidiary” variability

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

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

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

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

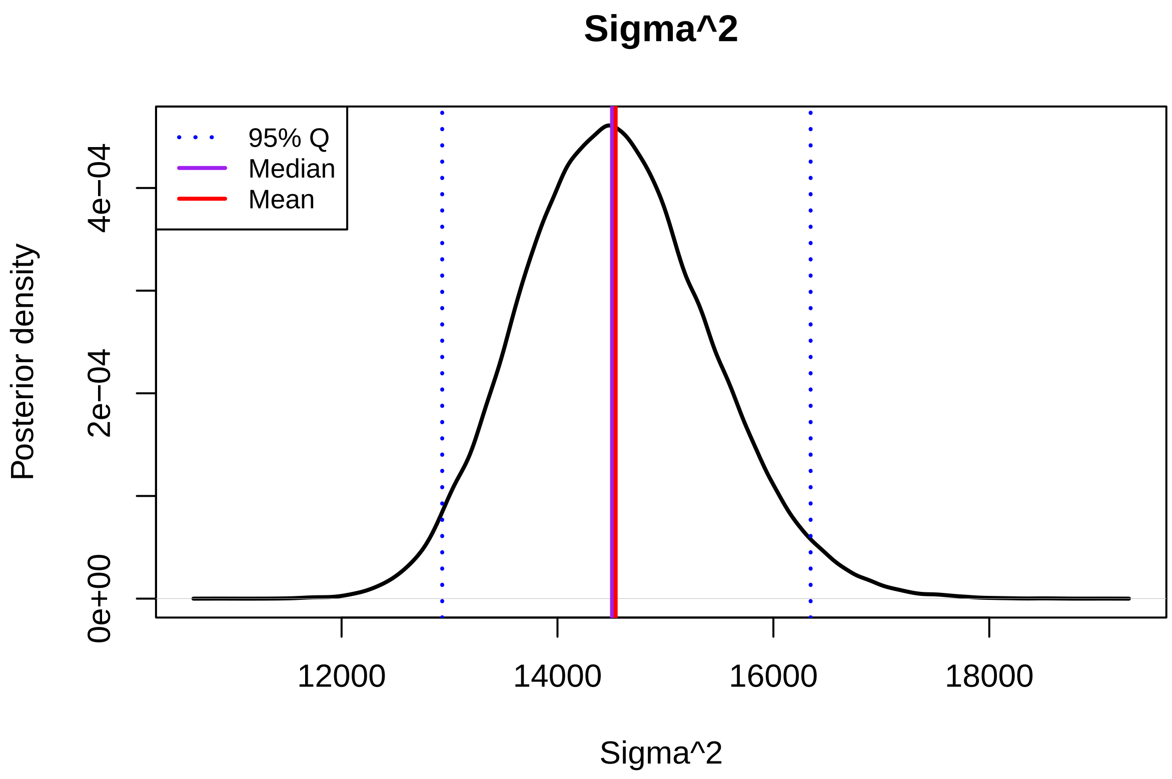
```
# "within-subsidiary" variability: sigma^2
mean(other.pars.MCMC[(burnin+1):S,2])
```

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

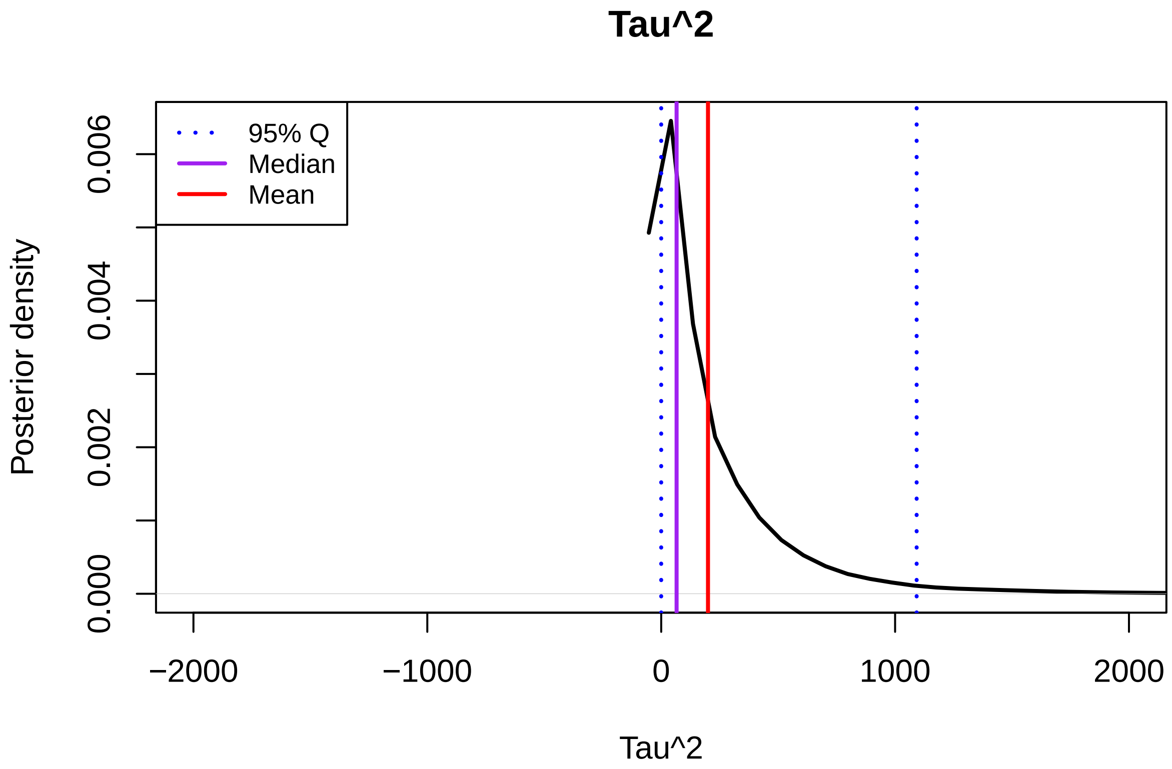
```
quantile(other.pars.MCMC[(burnin+1):S,2],c(0.025,0.975))
```

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

```
# Plot
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)
legend("topleft", legend = c("95% Q", "Median", "Mean"), col = c("blue", "purple", "red"), lty = c(3, 1
```



```
plot(density(other.pars.MCMC[(burnin+1):S,3]),col="black",lwd=2,lty=1,xlab="Tau^2",ylab="Posterior dens
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)

legend("topleft", legend = c("95% Q", "Median", "Mean"), col = c("blue", "purple", "red"), lty = c(3, 1
```

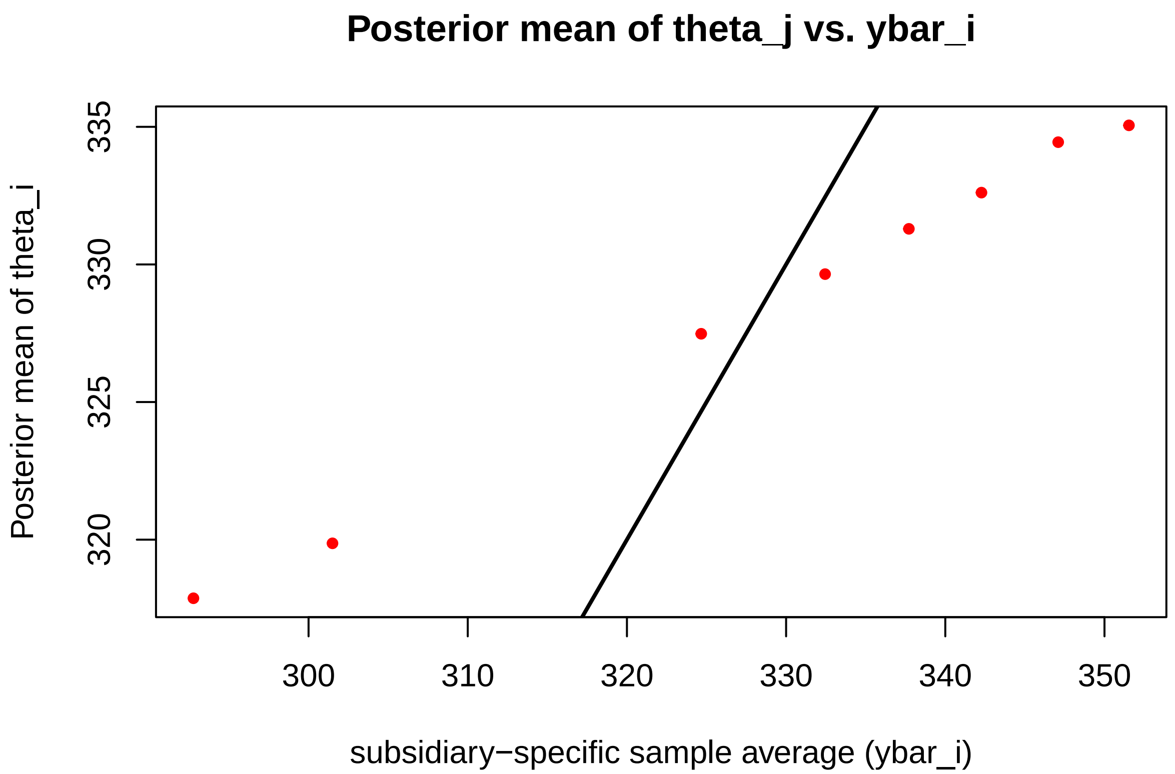


Based on above 95% credible intervals for variance, “between-subsidiaries” variability is much less than “within-subsidiary” variability because τ^2 is much less than σ^2.It indicates that most of the variability in the observed data is attributed to variations within individual subsidiaries rather than variations between different subsidiaries. This suggests that factors within each subsidiary have a more pronounced effect on daily cholesterol intake, while differences between subsidiaries are relatively minor.

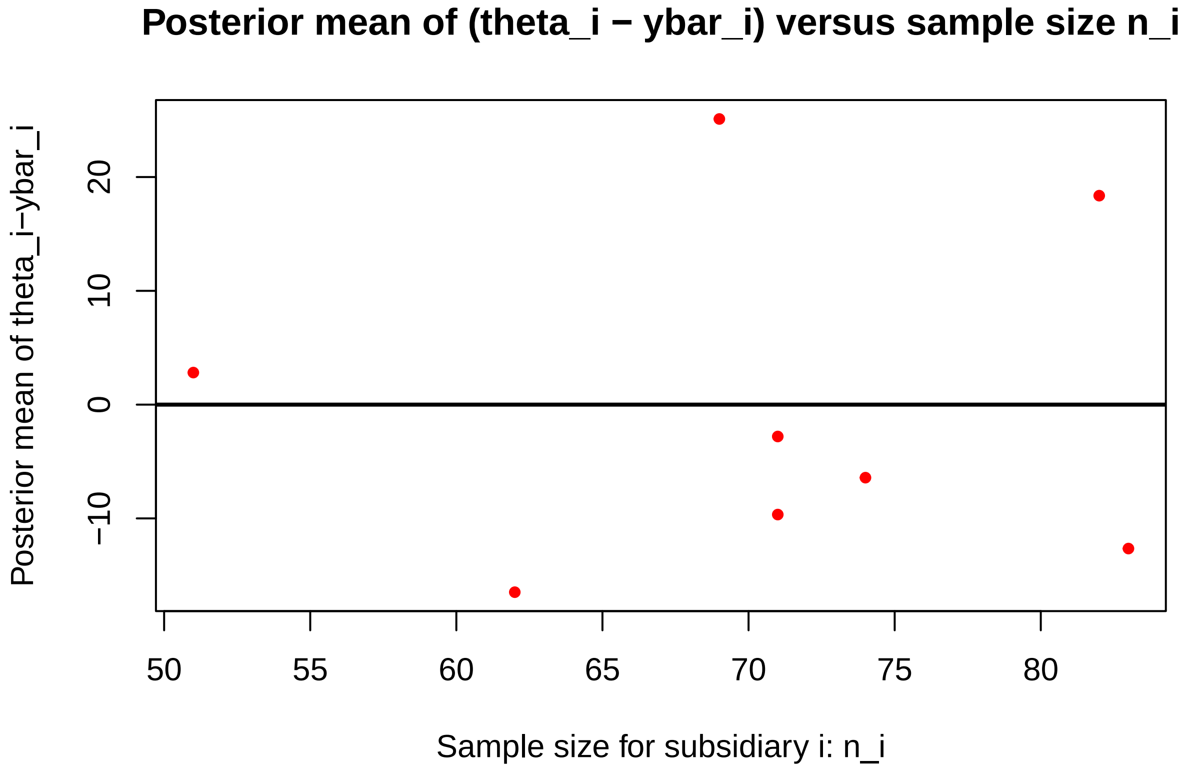
Shrinkage Evaluation

Shrinkage refers to the process in Bayesian analysis when sample sizes are small or data is not sufficiently reliable, the posterior mean θ_i is adjusted towards the overall mean μ to integrate information from a broader population. Conversely, when sample sizes are large or data is reliable, the specific information from the subsidiary becomes more crucial, and there is less reliance on the overall mean. This helps improve the stability of parameter estimation.

```
#### Shrinkage plots
plot(y.bar.sub,post.mean.theta,xlab="subsidiary-specific sample average (ybar_i)",ylab="Posterior mean 
col="red",pch=20,main="Posterior mean of theta_j vs. ybar_i")
abline(a=0,b=1,col="black", lwd = 2)
```



```
plot(n.j,post.mean.theta-y.bar.sub,xlab="Sample size for subsidiary i: n_i",ylab="Posterior mean of the
col="red",pch=20,main="Posterior mean of (theta_i - ybar_i) versus sample size n_i")
abline(h=0,col="black",lwd = 2)
```



Regarding above plot, 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.