ESC_WK5_HW

Sangjun Eom

2019 11 10

8.1

###a

$\mathrm{var}[y_{ij}|\mu,\tau^2]$ 이 더 클 것이다. 왜냐하면 within group sampling variability 뿐만 아니라 between group sampling variability 도 포함하기 때문이다.

###b

1. $\mathrm{Cov}[y_{i1,j}, y_{i2,j}|\theta_j,\sigma^2]$ 은 0 일 것이다. 왜냐하면 θ_j,σ^2가 알려진 상태에서 $y_{i,j}$는 conditionally iid 이기 때문이다.

2. 그러나 θ_j가 주어져 있지 않은 상황에서는 $y_{i1,j}$가 θ_j에 대한 정보를 제공하며 따라서 $y_{i2,j}$에 대한 정보를 제공해준다. 그리고 같은 θ_j에서 온 value 들은 서로 비슷한 값을 가질 것이다. 따라서 positive 한 cov 값을 가질 것으로 예상된다.

###c

1.

$\mathrm{Var}(y_{i,j}|\theta_j,\sigma^2) = \sigma^2$

2.

$\mathrm{Var}(\bar{y}_{\cdot,j}|\theta_j,\sigma^2) = \frac{\sigma^2}{n_j}$

3.

$\mathrm{Cov}(y_{i1,j}, y_{i2,j}|\theta_j,\sigma^2) = E(y_{i1,j}y_{i2,j}) - E(y_{i1,j})E(y_{i2,j}) = E(y_{i1,j})E(y_{i2,j}) - E(y_{i1,j})E(y_{i2,j}) = 0$

4.

$\mathrm{Var}(y_{i,j}|\mu,\tau^2) = \mathrm{Var}(\mathrm{E}(y_{ij}|\theta_j,\sigma^2)|\mu,\tau^2) + \mathrm{E}(\mathrm{Var}(y_{ij}|\theta_j,\sigma^2)|\mu,\tau^2) = \mathrm{Var}(\theta_j|\mu,\tau^2) + E(\sigma^2|\mu,\tau^2)$
$= \tau^2 + \sigma^2$

5.

$\mathrm{Var}(\bar{y}_{\cdot,j}|\mu,\tau^2) = \mathrm{Var}(\mathrm{E}(\bar{y}_{\cdot,j}|\theta_j,\sigma^2)|\mu,\tau^2) + \mathrm{E}(\mathrm{Var}(\bar{y}_{\cdot,j}|\theta_j,\sigma^2)|\mu,\tau^2) = \mathrm{Var}(\theta_j|\mu,\tau^2) + E\left(\frac{\sigma^2}{n_j}\middle|\mu,\tau^2\right)$
$= \tau^2 + \frac{\sigma^2}{n_j}$

6.

$\mathrm{Cov}(y_{i1,j}, y_{i2,j}|\mu,\tau^2)$= $\mathrm{E}(\mathrm{Cov}(y_{i1,j}, y_{i2,j}|\theta_j,\sigma^2)|\mu,\tau^2) + \mathrm{Cov}\left(\mathrm{E}(y_{i1,j}|\theta_j,\sigma^2), \mathrm{E}(y_{i2,j}|\theta_j,\sigma^2)\right) =$
$E(0|\mu,\tau^2) + Cov(\theta_j,\theta_j) = Var(\theta_j) = \tau^2$

a 와 b 에서 예측한대로 나왔다.

###d

Let

Y={$y_1,\cdots,y_m$}

$\theta = \{\theta_1,\cdots,\theta_m\}$

$$\mathrm{p}(\mu|\mathrm{Y},\theta,\sigma^2,\tau^2) = \frac{\mathrm{p}(\mu,\mathrm{Y},\theta,\sigma^2,\tau^2)}{\int \mathrm{p}(\mu,\mathrm{Y},\theta,\sigma^2,\tau^2)d\mu} = \frac{p(\mu)p(\tau^2)p(\sigma^2)p(Y|\theta,\sigma^2)p(\theta|\mu,\tau^2)}{\int p(\mu)p(\tau^2)p(\sigma^2)p(Y|\theta,\sigma^2)p(\theta|\mu,\tau^2)d\mu}$$
$$= \frac{p(\mu)p(\tau^2)p(\sigma^2)p(Y|\theta,\sigma^2)p(\theta|\mu,\tau^2)}{p(\tau^2)p(\sigma^2)p(Y|\theta,\sigma^2)\int p(\mu)p(\theta|\mu,\tau^2)d\mu} = \frac{p(\mu)p(\theta|\mu,\tau^2)}{\int p(\mu)p(\theta|\mu,\tau^2)d\mu} = \mathrm{p}(\mu|\theta,\tau^2)$$

즉, μ는 $\theta = \{\theta_1,\cdots,\theta_m\}$ 가 알려져 있는 경우, data 나 σ^2에 의존하지 않는다.

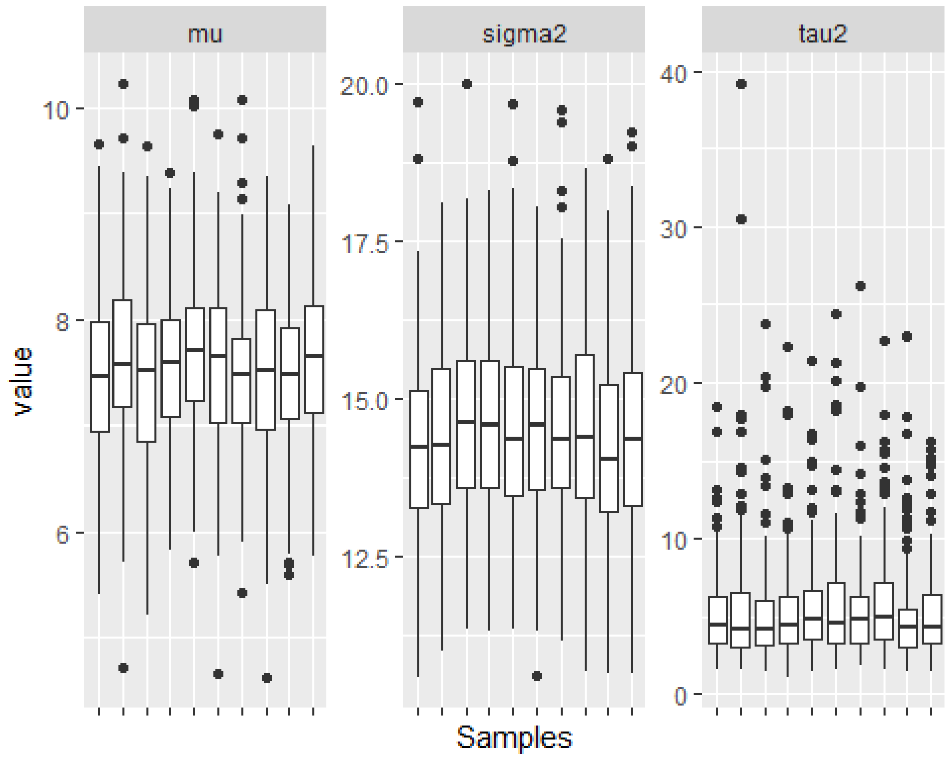
8.3

```
### a

# Load data
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

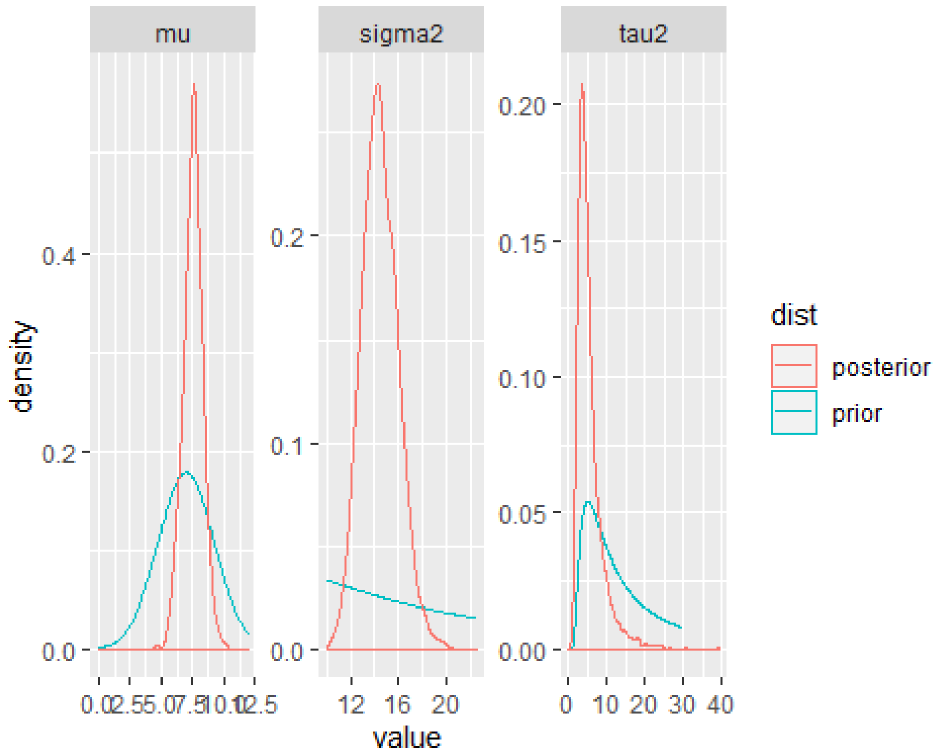
```
library(tidyr)
schools.list = lapply(1:8, function(i) {
  s.tbl = paste0('http://www.stat.washington.edu/people/pdhoff/Book/Data/hwda
ta/school', i, '.dat') %>%
    url %>%
    read.table

  data.frame(
    school = i,
    hours = s.tbl[, 1] %>% as.numeric
  )
})
schools.raw = do.call(rbind, schools.list)
Y = schools.raw
# Prior
mu0 = 7
g20 = 5
t20 = 10
eta0 = 2
s20 = 15
nu0 = 2
# Number of schools. Y[, 1] are school ids
m = length(unique(Y[, 1]))
# Starting values - use sample mean and variance
n = sv = ybar = rep(NA, m)
for (j in 1:m) {
  Y_j = Y[Y[, 1] == j, 2]
  ybar[j] = mean(Y_j)
  sv[j] = var(Y_j)
  n[j] = length(Y_j)
}
# Let initial theta estimates be the sample means
# Similarly, let initial values of sigma2, mu, and tau2 be "sample mean and
# variance"
theta = ybar
sigma2 = mean(sv)
mu = mean(theta)
tau2 = var(theta)
# MCMC
S = 1500
THETA = matrix(nrow = S, ncol = m)
# Storing sigma, mu, theta together
SMT = matrix(nrow = S, ncol = 3)
colnames(SMT) = c('sigma2', 'mu', 'tau2')
for (s in 1:S) {
```

```r
  # Sample thetas
  for (j in 1:m) {
    vtheta = 1 / (n[j] / sigma2 + 1 / tau2)
    etheta = vtheta * (ybar[j] * n[j] / sigma2 + mu / tau2)
    theta[j] = rnorm(1, etheta, sqrt(vtheta))
  }

  # Sample sigma2
  nun = nu0 + sum(n) # TODO: Could cache this
  ss = nu0 * s20
  # Pool variance
  for (j in 1:m) {
    ss = ss + sum((Y[Y[, 1] == j, 2] - theta[j])^2)
  }
  sigma2 = 1 / rgamma(1, nun / 2, ss / 2)

  # Sample mu
  vmu = 1 / (m / tau2 + 1 /g20)
  emu = vmu * (m * mean(theta) / tau2 + mu0 / g20)
  mu = rnorm(1, emu, sqrt(vmu))

  # Sample tau2
  etam = eta0 + m
  ss = eta0 * t20 + sum((theta - mu)^2)
  tau2 = 1 / rgamma(1, etam / 2, ss / 2)

  # Store params
  THETA[s, ] = theta
  SMT[s, ] = c(sigma2, mu, tau2)
}



smt.df = data.frame(SMT)
colnames(smt.df) = c('sigma2', 'mu', 'tau2')
smt.df$s = 1:S
cut_size = 10
smt.df = smt.df %>%
  tbl_df %>%
  mutate(scut = cut(s, breaks = cut_size)) %>%
  gather('variable', 'value', sigma2:tau2)

library(ggplot2)
ggplot(smt.df, aes(x = scut, y = value)) +
  facet_wrap(~ variable, scales = 'free_y') +
  geom_boxplot() +
  theme(axis.text.x = element_blank()) +
  xlab('Samples')
```

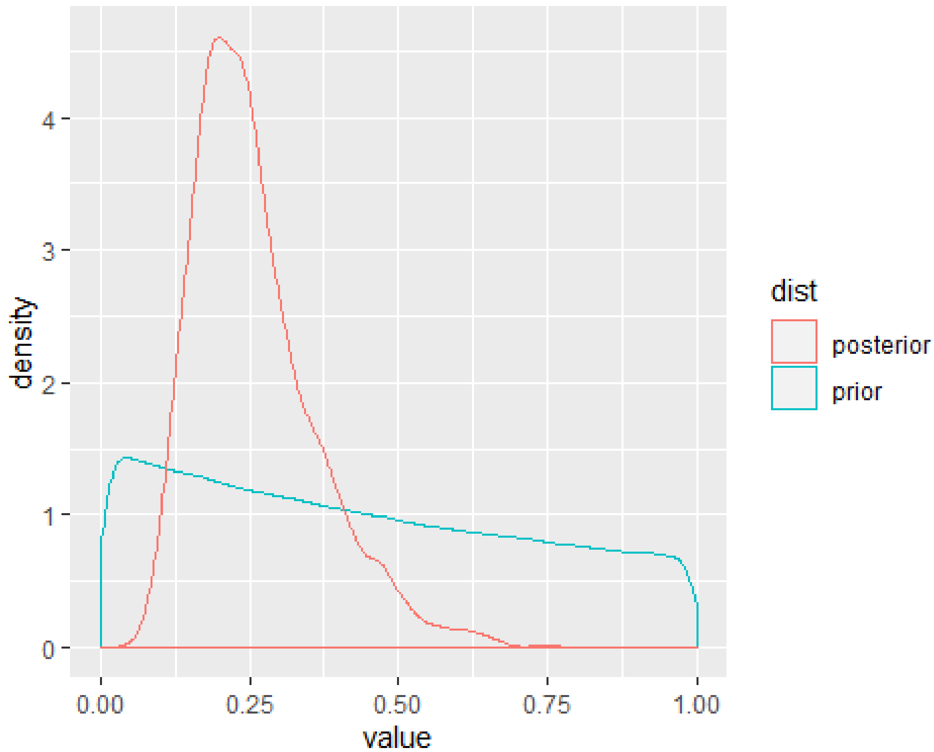



```
# Tweak number of samples until all of the below are above 1000
library(coda)
effectiveSize(SMT[, 1])

## var1
## 1500

effectiveSize(SMT[, 2])

##     var1
## 1091.984

effectiveSize(SMT[, 3])

##    var1
## 1079.57

### b

t(apply(SMT, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.5, 0.975)))

##             2.5%       50%     97.5%
## sigma2 11.674406 14.364126 17.565276
## mu      5.936803  7.567911  9.032519
## tau2    1.876914  4.481536 15.019585
```

```r
# For dinvgamma
library(MCMCpack)

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2019 Andrew D. Martin, Kevin M. Quinn, and Jong Hee
Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##

sigma2_prior = data.frame(
  value = seq(10, 22.5, by = 0.1),
  density = dinvgamma(seq(10, 22.5, by = 0.1), nu0 / 2, nu0 * s20 / 2),
  variable = 'sigma2'
)
tau2_prior = data.frame(
  value = seq(0, 30, by = 0.1),
  density = dinvgamma(seq(0, 30, by = 0.1), eta0 / 2, eta0 * t20 / 2),
  variable = 'tau2'
)
mu_prior = data.frame(
  value = seq(0, 12, by = 0.1),
  density = dnorm(seq(0, 12, by = 0.1), mu0, sqrt(g20)),
  variable = 'mu'
)
priors = rbind(sigma2_prior, tau2_prior, mu_prior)
priors$dist = 'prior'
smt.df$dist = 'posterior'
ggplot(priors, aes(x = value, y = density, color = dist)) +
  geom_line() +
  geom_density(data = smt.df, mapping = aes(x = value, y = ..density..)) +
  facet_wrap(~ variable, scales = 'free')
```




```r
### c
t20_prior = (1 / rgamma(1e6, eta0 / 2, eta0 * t20 / 2))
s20_prior = (1 / rgamma(1e6, nu0 / 2, nu0 * s20 / 2))
R_prior = data.frame(
  value = (t20_prior) / (t20_prior + s20_prior),
  dist = 'prior'
)
R_post = data.frame(
  value = SMT[, 'tau2'] / (SMT[, 'tau2'] + SMT[, 'sigma2']),
  dist = 'posterior'
)
ggplot(R_prior, aes(x = value, y = ..density.., color = dist)) +
  geom_density(data = R_prior) +
  geom_density(data = R_post)
```




```
mean(R_post$value)

## [1] 0.2581611

### d


theta7_lt_6 = THETA[, 7] < THETA[, 6]
mean(theta7_lt_6)

## [1] 0.492

theta7_smallest = (THETA[, 7] < THETA[, -7]) %>%
  apply(MARGIN = 1, FUN = all)
mean(theta7_smallest)

## [1] 0.31

### e


relationship = data.frame(
  sample_average = ybar,
  post_exp = colMeans(THETA),
  school = 1:length(ybar)
)
```

```r
ggplot(relationship, aes(x = sample_average, y = post_exp, label = school)) +
  geom_text() +
  geom_abline(slope = 1, intercept = 0) +
  geom_hline(yintercept = mean(schools.raw[, 'hours']), lty = 2) +
  annotate('text', x = 10, y = 7.9, label = paste0("Pooled sample mean ", rou
nd(mean(schools.raw[, 'hours']), 2))) +
  geom_hline(yintercept = mean(SMT[, 'mu']), color = 'red') +
  annotate('text', x = 10, y = 7.4, label = paste0("Posterior exp. mu ", roun
d(mean(SMT[, 'mu']), 2)), color = 'red')
```