Assignment #8

Yutae Lee

2022-11-08

Problem 8.1

a)

I think $\text{Var}(y_{i,j} \mid \mu, \tau^2)$ will be bigger than $\text{Var}(y_{i,j} \mid \theta_i, \sigma^2)$. Intuitively speaking, while the first value is sampling from fixed group, second group will have to include variability from within the group and between the groups.

b) I think $\text{Cov}(y_{i_1,j}, y_{i_2,j} \mid \theta_j, \sigma^2)$ is zero, assuming exchangeability, we can say that $y_{i_1,j}$, $y_{i_2,j}$ are conditionally i.i.d., when we have a known θ_j.

For $\text{Cov}(y_{i_1,j}, y_{i_2,j} \mid \theta_j, \sigma^2)$, since θ_j is unknown, I can assume that $y_{i_1,j}, y_{i_2,j}$ have positive covariance.

c)

First, $Var[y_{i,j} \mid \theta_i, \sigma^2] = \sigma^2$ by the definition of variance with known parameter.

$Var[\bar{y}., j \mid \theta_i, \sigma^2] = \frac{\sigma^2}{n_j}$ by the Sample Group Mean

$Cov[y_{i_1,j}, y_{i_2,j} \mid \theta_j, \sigma^2] = \mathbb{E}(y_{i_1,j}, y_{i_2,j}) - \mathbb{E}(y_{i_1,j})\mathbb{E}(y_{i_2,j})$

Because it is conditionally i.i.d,

$\mathbb{E}(y_{i_1,j}, y_{i_2,j}) - \mathbb{E}(y_{i_1,j})\mathbb{E}(y_{i_2,j}) = \mathbb{E}(y_{i_1,j})\mathbb{E}(y_{i_2,j}) - \mathbb{E}(y_{i_1,j})\mathbb{E}(y_{i_2,j}) = 0$

Next $Var[y_{i,j} \mid \mu, \tau^2] = Var[\mathbb{E}(y_{i,j} \mid \theta_j, \sigma^2) \mid \mu, \tau^2] + \mathbb{E}(Var[y_{i,j} \mid \theta_j, \sigma^2] \mid \mu, \tau^2)$, by the Law of Total Variance

$= Var(\theta_j \mid \mu, \tau^2) + \mathbb{E}(\sigma^2 \mid \mu, \tau^2)$

Which we can easily obtain $\tau^2 + \sigma^2 > 0$

$Var[\bar{y}., j \mid \mu, \tau^2] = Var[\mathbb{E}(\bar{y}., j \mid \theta_j, \sigma^2) \mid \mu, \tau^2] + \mathbb{E}(Var[\bar{y}., j \mid \theta_j, \sigma^2] \mid \mu, \tau^2)$, by the Law of Total Variance

$= Var(\theta_j \mid \mu, \tau^2) + \mathbb{E}(\sigma^2/n_j \mid \mu, \tau^2)$ by sample group mean

$= \tau^2 + \frac{\sigma^2}{n_j} > 0$

$Cov[y_{i_1,j}, y_{i_2,j} \mid \mu, \tau^2] = \mathbb{E}(Cov[y_{i_1,j}, y_{i-2,j} \mid \theta_j, \sigma^2] \mid \mu, \tau^2) + Cov[\mathbb{E}(y_{i_1,j} \mid \theta_j, \sigma^2), \mathbb{E}(y_{i_2,j} \mid \theta_j, \sigma^2)]$

by the Law of Total Covariance

$=\mathbb{E}(0 \mid \mu, \tau^2) + Cov[\mathbb{E}(y_{i_1,j} \mid \theta_j, \sigma^2), \mathbb{E}(y_{i_2,j} \mid \theta_j, \sigma^2)]$

$= Cov(\theta_j, \theta_j) = Var(\theta_j)$

$= \tau^2 > 0$

We can see that assumption from a),b) is correct after our computation.

d)

$p(\mu \mid \theta_1, \dots, \theta_m, \sigma^2, \tau^2, \mathbf{y}_1, \dots, \mathbf{y}_m)$

$= \frac{p(\mu,\theta_1,\ldots,\theta_m,\sigma^2,\tau^2,\mathbf{y}_1,\ldots,\mathbf{y}_m)}{\int p(\mu,\theta_1,\ldots,\theta_m,\sigma^2,\tau^2,\mathbf{y}_1,\ldots,\mathbf{y}_m)d\mu}$

$= \frac{p(\mu)p(\mathbf{y}_1,\ldots,\mathbf{y}_m|\theta_1,\ldots,\theta_m,\sigma^2)p(\theta_1,\ldots,\theta_m|\mu,\tau^2)p(\tau^2)p(\sigma^2)}{\int p(\mu)p(\mathbf{y}_1,\ldots,\mathbf{y}_m|\theta_1,\ldots,\theta_m,\sigma^2)p(\theta_1,\ldots,\theta_m|\mu,\tau^2)p(\tau^2)p(\sigma^2)d\mu}$

Simplifying,

$= \frac{p(\mu)p(\theta_1,\ldots,\theta_m|\mu,\tau^2)}{\int p(\mu)p(\theta_1,\ldots,\theta_m|\mu,\tau^2)d\mu}$

$= p(\mu|\theta_1,\ldots,\theta_m,\tau^2)$ by the bayes' theorem.

In words, we can say that μ is not $\propto \mathbf{y}_1,\ldots,\mathbf{y}_m,\sigma^2$, if the θ's are known.

Problem 8.3

```
library(dplyr)
```

```
## Warning:    'dplyr'  R   4.2.2

##
##          : '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)
```

```
## Warning:    'tidyr'  R   4.2.2
```

```
set.seed(32)
school1 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school1.dat'))
school2 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school2.dat'))
school3 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school3.dat'))
school4 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school4.dat'))
school5 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school5.dat'))
school6 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school6.dat'))
school7 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school7.dat'))
school8 <- scan(url('https://www2.stat.duke.edu/courses/Fall09/sta290/datasets/Hoffdata/school8.dat'))
school1 <- data.frame(school = 1, hours = school1)
school2 <- data.frame(school = 2, hours = school2)
school3 <- data.frame(school = 3, hours = school3)
school4 <- data.frame(school = 4, hours = school4)
school5 <- data.frame(school = 5, hours = school5)
school6 <- data.frame(school = 6, hours = school6)
school7 <- data.frame(school = 7, hours = school7)
school8 <- data.frame(school = 8, hours = school8)
Y <- rbind(school1,school2,school3,school4,school5,school6,school7,school8)
```

```
mu_0 = 7
gamma2_0 = 5
tau2_0 = 10
eta_0 = 2
sigma2_0 = 15
nu_0 = 2
```

a)

```
number_of_school = 8
n = rep(NA, 8)
sv = rep(NA, 8)
ybar = rep(NA, 8)
for (j in 1:8) {
  Y_j = Y[Y[, 1] == j, 2]
  ybar[j] = mean(Y_j)
  sv[j] = var(Y_j)
  n[j] = length(Y_j)
}

theta = ybar
sigma2 = mean(sv)
mu = mean(theta)
tau2 = var(theta)
```

```
set.seed(32)
theta_matrix = matrix(nrow = 1500, ncol = 8)

sig_mu_tau = matrix(nrow = 1500, ncol = 3)
colnames(sig_mu_tau) = c('sigma2', 'mu', 'tau2')
for (s in 1:1500) {
  for (j in 1:8) {
    vtheta = 1 / (n[j] / sigma2 + 1 / tau2)
    etheta = vtheta * (ybar[j] * n[j] / sigma2 + mu / tau2)
    theta[j] = rnorm(1, etheta, sqrt(vtheta))
  }

  nu_n = nu_0 + sum(n)
  s_s = nu_0 * sigma2_0
  for (i in 1:8) {
    s_s = s_s + sum((Y[Y[, 1] == i, 2] - theta[i])^2)
  }
  sigma2 = 1 / rgamma(1, nu_n / 2, s_s / 2)

  vmu = 1 / (8 / tau2 + 1 /gamma2_0)
  emu = vmu * (8 * mean(theta) / tau2 + mu_0 / gamma2_0)
  mu = rnorm(1, emu, sqrt(vmu))

  eta_8 = eta_0 + 8
  s_s = eta_0 * tau2_0 + sum((theta - mu)^2)
  tau2 = 1 / rgamma(1, eta_8 / 2, s_s / 2)

  theta_matrix[s, ] = theta
```

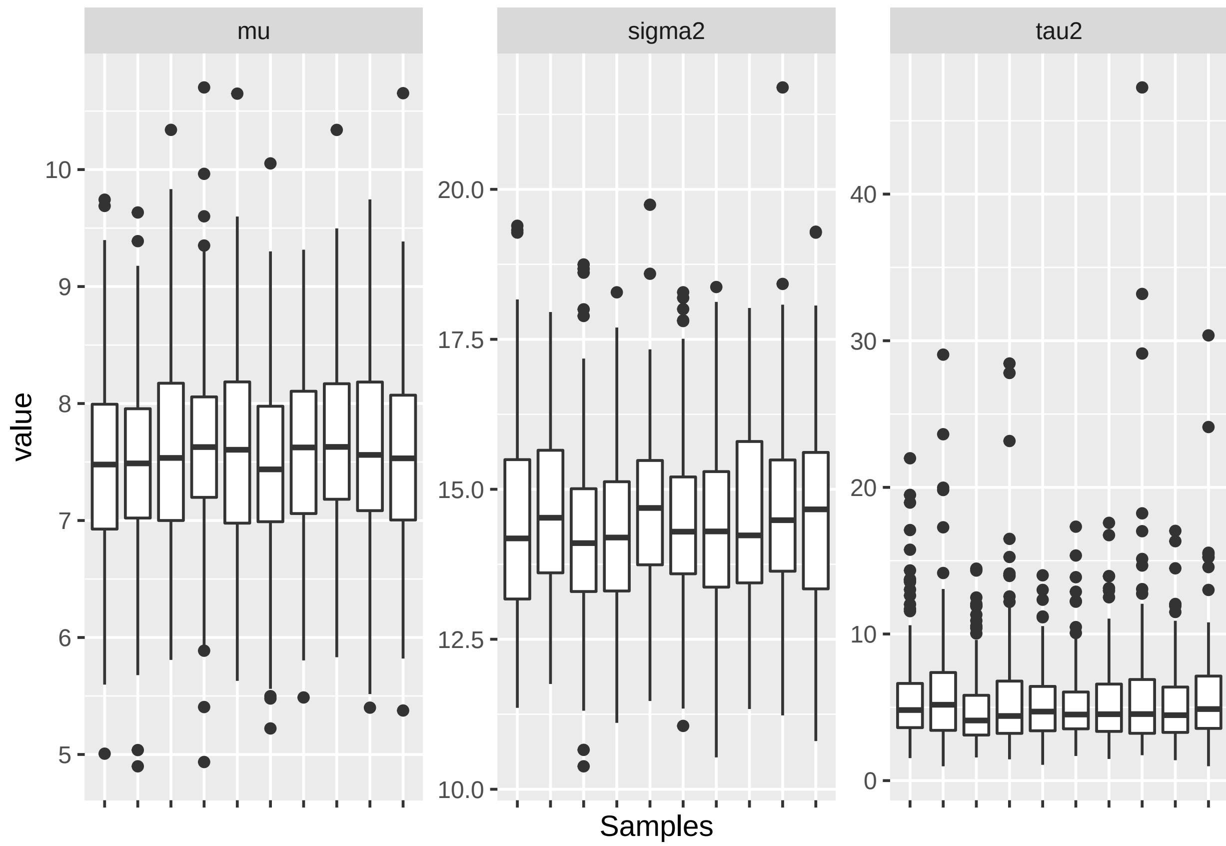
```
  sig_mu_tau[s, ] = c(sigma2, mu, tau2)
}
```

Boxplot of σ^2, μ, τ^2:

```
library(ggplot2)
sig_mu_tau_df = data.frame(sig_mu_tau)
colnames(sig_mu_tau_df) = c('sigma2', 'mu', 'tau2')
sig_mu_tau_df$s = 1:1500
cut_size = 10
sig_mu_tau_df = sig_mu_tau_df %>%
  tbl_df %>%
  mutate(s_cut = cut(s, breaks = cut_size)) %>%
  gather('variable', 'value', sigma2:tau2)
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## i Please use `tibble::as_tibble()` instead.
```

```
ggplot(sig_mu_tau_df, aes(x = s_cut, y = value)) +
  facet_wrap(~ variable, scales = 'free_y') +
  geom_boxplot() +
  theme(axis.text.x = element_blank()) +
  xlab('Samples')
```



Effective sample sizes of σ^2, μ, τ^2:

```r
library(coda)
```

```
## Warning:    'coda'  R   4.2.2
```

```r
effectiveSize(sig_mu_tau[, 1])
```

```
## var1
## 1500
```

```r
effectiveSize(sig_mu_tau[, 2])
```

```
##     var1
## 1320.758
```

```r
effectiveSize(sig_mu_tau[, 3])
```

```
##     var1
## 1102.411
```

As we can see they are 1500, 1321, 1102 when they are rounded to whole number.

b)

The posterior means and 95% confidence regions are following:

```r
t(apply(sig_mu_tau, MARGIN = 2, FUN = quantile, probs = c(0.025, 0.5, 0.975)))
```

```
##              2.5%       50%     97.5%
## sigma2 11.700481 14.347298 17.810885
## mu      5.951327  7.549100  9.175179
## tau2    1.884821  4.611138 14.331560
```

Comparing posterior to prior:

```r
set.seed(32)
library(MASS)
```

```
##
##          : 'MASS'
```

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

```r
library(MCMCpack)
```

```
## Warning:    'MCMCpack'  R   4.2.2
```

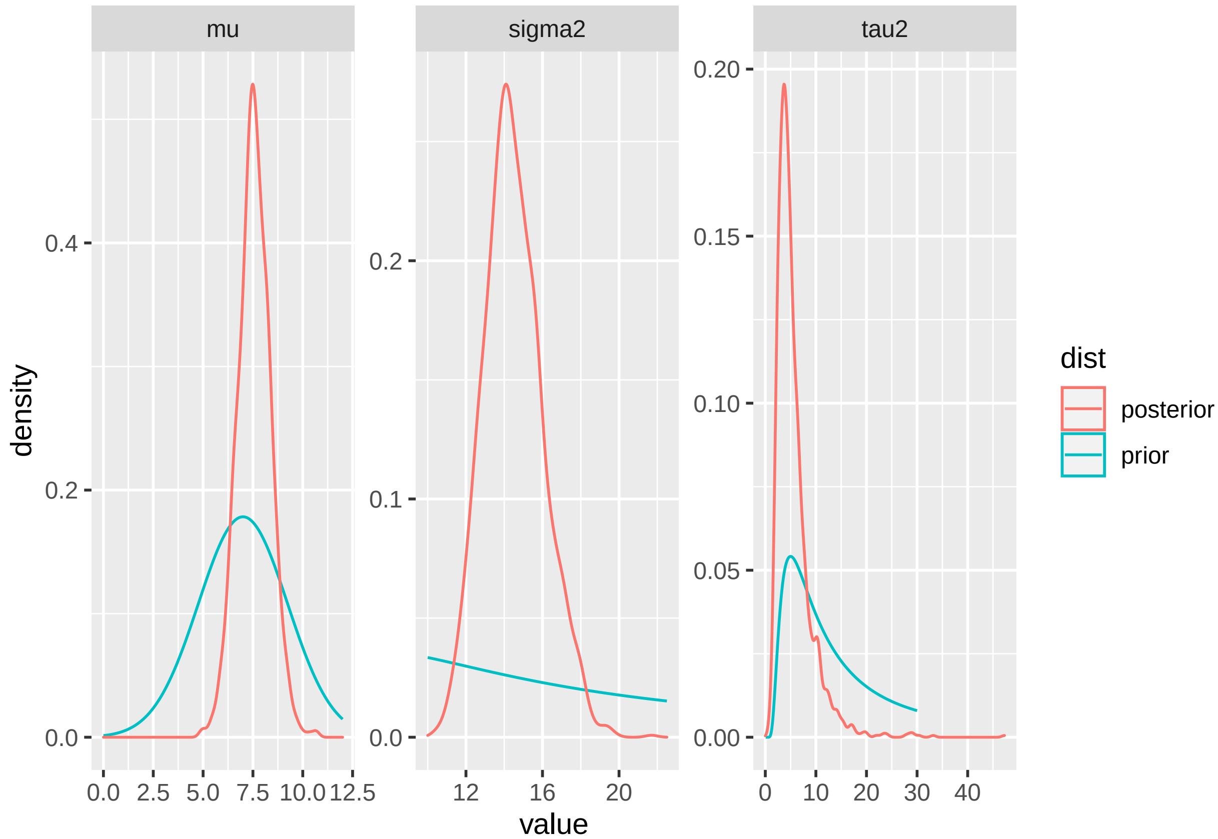
```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2022 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)
## ##
```

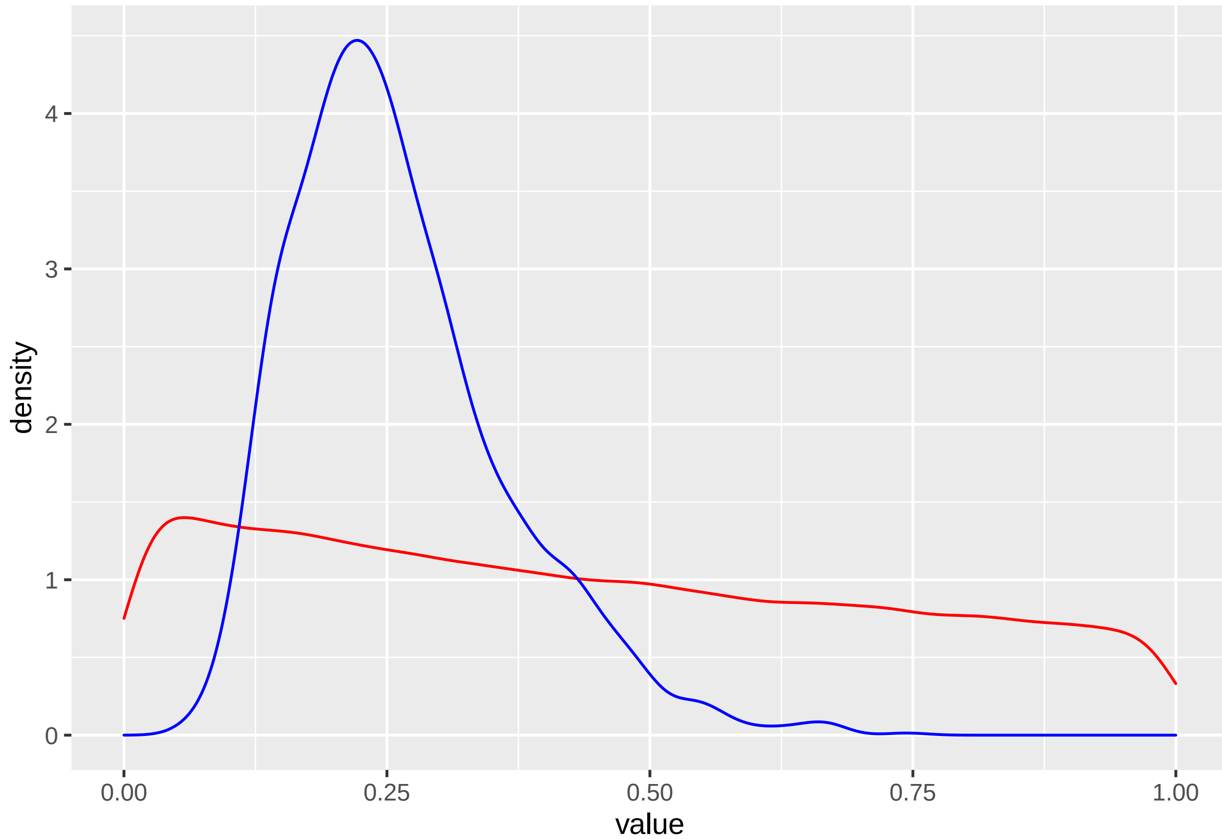
```
sigma2_prior = data.frame(
  value = seq(10, 22.5, by = 0.1),
  density = dinvgamma(seq(10, 22.5, by = 0.1), nu_0 / 2, nu_0 * sigma2_0 / 2),
  variable = 'sigma2'
)
tau2_prior = data.frame(
  value = seq(0, 30, by = 0.1),
  density = dinvgamma(seq(0, 30, by = 0.1), eta_0 / 2, eta_0 * tau2_0 / 2),
  variable = 'tau2'
)
mu_prior = data.frame(
  value = seq(0, 12, by = 0.1),
  density = dnorm(seq(0, 12, by = 0.1), mu_0, sqrt(gamma2_0)),
  variable = 'mu'
)
priors = rbind(sigma2_prior, tau2_prior, mu_prior)
priors$dist = 'prior'
sig_mu_tau_df$dist = 'posterior'
ggplot(priors, aes(x = value, y = density, color = dist)) +
  geom_line() +
  geom_density(data = sig_mu_tau_df, mapping = aes(x = value, y = ..density..)) +
  facet_wrap(~ variable, scales = 'free')
```



We can see that μ, τ^2 had similar prior and posterior, but σ^2 is much different. Using the fact we have good estimates of average hours of studying and variability between different schools in those. But the variability among students is different.

c)

```
set.seed(32)
tau2_0_prior = (1 / rgamma(100000, eta_0 / 2, eta_0 * tau2_0 / 2))
sigma2_0_prior = (1 / rgamma(100000, nu_0 / 2, nu_0 * sigma2_0 / 2))
prior_df = data.frame(
  value = (tau2_0_prior) / (tau2_0_prior + sigma2_0_prior),
  dist = 'prior'
)
posterior_df = data.frame(
  value = sig_mu_tau[, 'tau2'] / (sig_mu_tau[, 'tau2'] + sig_mu_tau[, 'sigma2']),
  dist = 'posterior'
)
ggplot(prior_df, aes(x = value, y = ..density..)) +
  geom_density(data = prior_df, color = 'red') +
  geom_density(data = posterior_df, color = 'blue')
```



The mean of posterior is

```
mean(posterior_df$value)
```

```
## [1] 0.2611494
```

d)

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

```
## [1] 0.524
```

```
theta7= (theta_matrix[, 7] < theta_matrix[, -7]) %>%
  apply(MARGIN = 1, FUN = all)
mean(theta7)
```

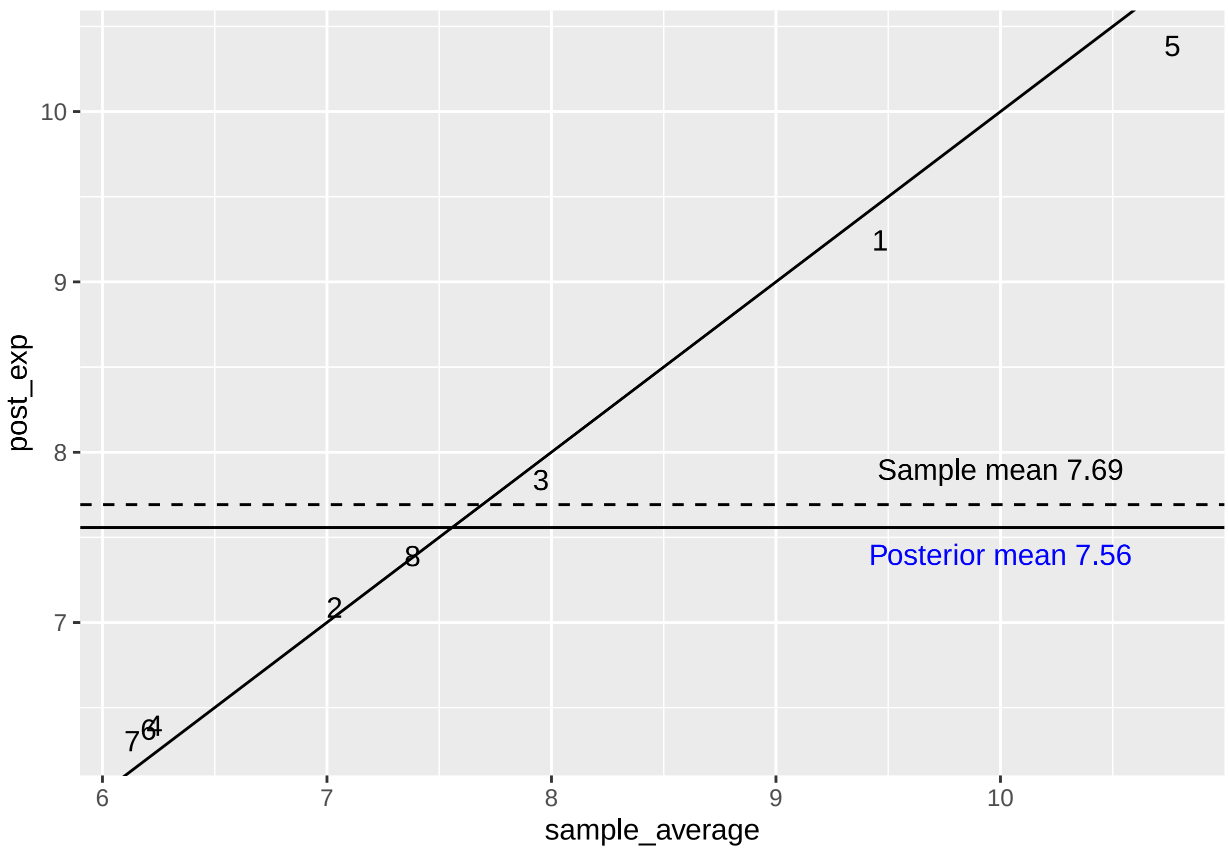
```
## [1] 0.322
```

The probability of θ_7 smaller than θ_6 is 0.524.

The probability of θ_7 is the smallest of all θ's is 0.322.

e)

```
relationship = data.frame(
  sample_average = ybar,
  post_exp = colMeans(theta_matrix),
  school = 1:length(ybar)
)
ggplot(relationship, aes(x = sample_average, y = post_exp, label = school)) +
  geom_text() +
  geom_abline(slope = 1, intercept = 0) +
  geom_hline(yintercept = mean(Y[, 'hours']), lty = 2) +
  annotate('text', x = 10, y = 7.9, label = paste0("Sample mean ", round(mean(Y[, 'hours']), 2))) +
  geom_hline(yintercept = mean(sig_mu_tau[, 'mu'])) +
  annotate('text', x = 10, y = 7.4, label = paste0("Posterior mean ", round(mean(sig_mu_tau[, 'mu']), 2
```



We can see that posterior mean and mean of pooled sample are pretty similar.