

S626

HW3

John Koo

```
library(ggplot2)
import::from(magrittr, `%>%`)

theme_set(theme_bw())
set.seed(626)
doMC::registerDoMC(8)
```

4.2

a

From problem 3.3, we know:

- $\theta_A \mid y_A \sim \text{Gamma}(120 + \sum y_{A,i}, 10 + n_A)$
- $\theta_B \mid y_B \sim \text{Gamma}(12 + \sum y_{B,i}, 1 + n_B)$

```
# data
y.a <- c(12, 9, 12, 14, 13, 13, 15, 8, 15, 6)
y.b <- c(11, 11, 10, 9, 9, 8, 7, 10, 6, 8, 8, 9, 7)

# statistics
n.a <- length(y.a)
n.b <- length(y.b)
sum.y.a <- sum(y.a)
sum.y.b <- sum(y.b)

# prior parameters
a.a <- 120
a.b <- 10
b.a <- 12
b.b <- 1

# sample size
n <- 2 ** 10

# generate sample and compute probability
theta.a <- rgamma(n, a.a + sum(y.a), a.b + n.a)
theta.b <- rgamma(n, b.a + sum(y.b), b.b + n.b)
mean(theta.b < theta.a)
```

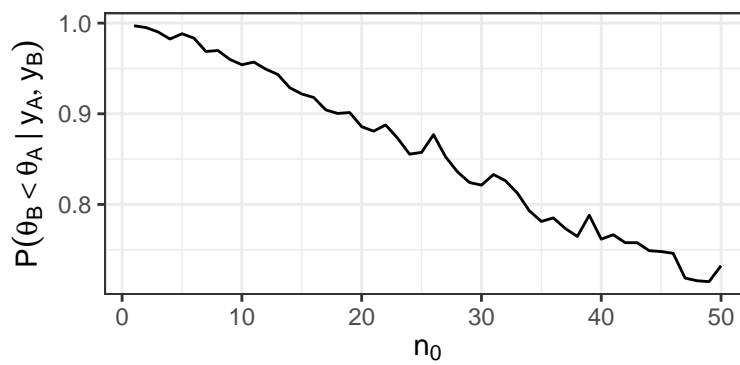
```
[1] 0.9941406
```

b

```
# values of n_0 to try
n.0.vector <- seq(50)

sensitivity.df <- plyr::ldply(n.0.vector, function(n.0) {
  theta.a <- rgamma(n, a.a + sum(y.a), a.b + n.a)
  theta.b <- rgamma(n, 12 * n.0 + sum(y.b), n.0 + n.b)
  p <- mean(theta.b < theta.a)
  dplyr::data_frame(n.0 = n.0, p = p)
}, .parallel = TRUE)

ggplot(sensitivity.df) +
  geom_line(aes(x = n.0, y = p)) +
  labs(x = expression(n[0]),
       y = expression(P(theta[B] < theta[A] ~'|'~y[A], y[B])))
```



As n_0 increases, we are more sure that $\theta_B = E[Y_B] = 12$, and since $\bar{Y}_A < 12$, even though $\bar{Y}_B < \bar{Y}_A$, our posterior estimate for θ_B approaches 12, so the probability decreases.

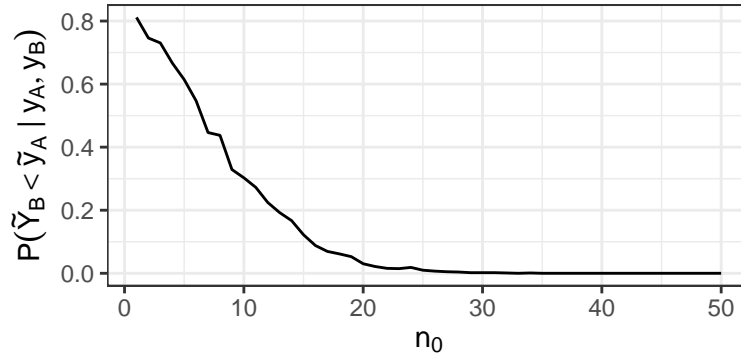
c

From class, we know:

- $\tilde{Y}_A | y_A \sim NB(120 + \sum y_{A,i}, 10 + n_A)$
- $\tilde{Y}_B | y_B \sim NB(12n_0 + \sum y_{B,i}, n_0 + n_B)$

```
pred.posterior.df <- plyr::ldply(n.0.vector, function(n.0) {
  y.a.tilde <- rnbino(n, a.a + sum(y.a), mu = a.b + n.a)
  y.b.tilde <- rnbino(n, 12 * n.0 + sum(y.b), mu = n.0 + n.b)
  dplyr::data_frame(n.0 = n.0, p = mean(y.b.tilde < y.a.tilde))
}, .parallel = TRUE)

ggplot(pred.posterior.df) +
  geom_line(aes(x = n.0, y = p)) +
  labs(x = expression(n[0]),
       y = expression(P(tilde(Y)[B] < tilde(y)[A] ~'|'~y[A], y[B])))
```



Similar to above, the probability goes to 0 as we become more and more sure that $\theta_B > \theta_A$.

4.5

a

Given

- $Y_i | \theta, x_i \sim \text{Poisson}(\theta x_i)$
- $\theta \sim \text{Gamma}(a, b)$

Then

$$\begin{aligned}
 & \bullet \quad p(\theta | x, y) = p(\theta) \prod p(y_i | \theta, x_i) \\
 & \propto \theta^{a-1} e^{-b\theta} \theta^{\sum y_i} e^{-\theta \sum x_i} \\
 & = \theta^{\sum y_i + a - 1} e^{-(b + \sum x_i)\theta} \\
 & \implies \theta | x, y \sim \text{Gamma}(\sum y_i + a, \sum x_i + b)
 \end{aligned}$$

b

```

react.df <- readr::read_table(
  'http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/cancer_react.dat'
)
noreact.df <- readr::read_table(
  'http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/cancer_noreact.dat'
)

apply(react.df, 2, sum)

```

```

      x      y
95 256

```

```

apply(noreact.df, 2, sum)

```

```

      x      y
1037 2285

```

So $\theta_1 | x, y \sim \text{Gamma}(256 + a_1, 85 + b_1)$
 and $\theta_2 | x, y \sim \text{Gamma}(2285 + a_2, 1037 + b_2)$

c

For the calculations of $P(\theta_2 > \theta_1 | x, y)$, I will use Monte-Carlo estimation instead of computing the integrals. The plots will be done last for better comparison.

Before we compute any posterior probabilities or estimates, it's worth computing the MLEs as a comparison:

- $\hat{\theta}_1 = 2285/1037 \approx 2.203$
- $\hat{\theta}_2 = 256/95 \approx 2.695$

i

```
# statistics
sum.x2 <- sum(react.df$x)
sum.y2 <- sum(react.df$y)
sum.x1 <- sum(noreact.df$x)
sum.y1 <- sum(noreact.df$y)

# parameters
a1 <- a2 <- 2.2 * 100
b1 <- b2 <- 100

# expected values
(sum.y1 + a1) / (sum.x1 + b1)

[1] 2.203166
(sum.y2 + a2) / (sum.x2 + b2)

[1] 2.441026

# intervals
qgamma(c(.025, .975), sum.y1 + a1, sum.x1 + b1)

[1] 2.117726 2.290273
qgamma(c(.025, .975), sum.y2 + a2, sum.x2 + b2)

[1] 2.226633 2.665131

# p(theta_1 > theta_2 | x, y)
theta.1 <- rgamma(n, sum.y1 + a1, sum.x1 + b1)
theta.2 <- rgamma(n, sum.y2 + a2, sum.x2 + b2)
mean(theta.2 > theta.1)

[1] 0.9765625
```

ii

```
# parameters
a1 <- 2.2 * 100
b1 <- 100
a2 <- 2.2
b2 <- 1
```

```

# expected values
(sum.y1 + a1) / (sum.x1 + b1)

[1] 2.203166
(sum.y2 + a2) / (sum.x2 + b2)

[1] 2.689583
# intervals
qgamma(c(.025, .975), sum.y1 + a1, sum.x1 + b1)

[1] 2.117726 2.290273
qgamma(c(.025, .975), sum.y2 + a2, sum.x2 + b2)

[1] 2.371497 3.027397
# p(theta_1 > theta_2 | x, y)
theta.1 <- rgamma(n, sum.y1 + a1, sum.x1 + b1)
theta.2 <- rgamma(n, sum.y2 + a2, sum.x2 + b2)
mean(theta.2 > theta.1)

[1] 1

```

iii

```

# parameters
a1 <- a2 <- 2.2
b1 <- b2 <- 1

# expected values
(sum.y1 + a1) / (sum.x1 + b1)

[1] 2.203468
(sum.y2 + a2) / (sum.x2 + b2)

[1] 2.689583
# intervals
qgamma(c(.025, .975), sum.y1 + a1, sum.x1 + b1)

[1] 2.114081 2.294680
qgamma(c(.025, .975), sum.y2 + a2, sum.x2 + b2)

[1] 2.371497 3.027397
# p(theta_1 > theta_2 | x, y)
theta.1 <- rgamma(n, sum.y1 + a1, sum.x1 + b1)
theta.2 <- rgamma(n, sum.y2 + a2, sum.x2 + b2)
mean(theta.2 > theta.1)

[1] 0.9990234

```

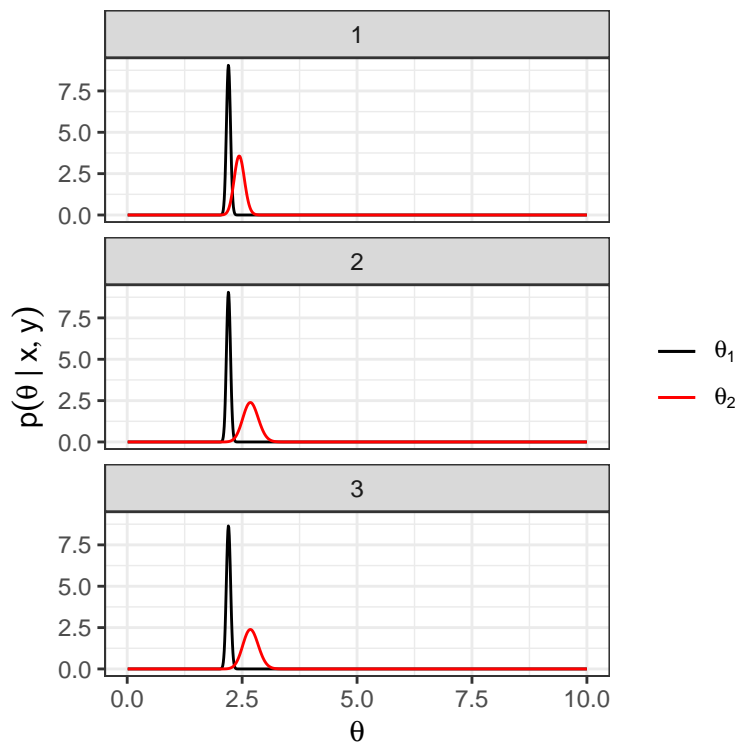
posterior plots and comparison

```
# support of theta
theta <- seq(.01, 10, .01)

# parameters
a1 <- c(2.2 * 100, 2.2 * 100, 2.2)
a2 <- c(2.2 * 100, 2.2, 2.2)
b1 <- c(100, 100, 1)
b2 <- c(100, 1, 1)

out.df <- plyr::ldply(seq(3), function(i) {
  p.1 <- dgamma(theta, a1[i] + sum.y1, b1[i] + sum.x1)
  p.2 <- dgamma(theta, a2[i] + sum.y2, b2[i] + sum.x2)
  dplyr::data_frame(theta = theta, p.1 = p.1, p.2 = p.2, scenario = i)
}, .parallel = TRUE)

ggplot(out.df) +
  facet_wrap(~ scenario, ncol = 1) +
  geom_line(aes(x = theta, y = p.1, colour = '1')) +
  geom_line(aes(x = theta, y = p.2, colour = '2')) +
  labs(x = expression(theta), y = expression(p(theta~'|'~x, y))) +
  scale_colour_manual(labels = c(expression(theta[1]),
                                expression(theta[2])),
                     values = seq(2),
                     name = NULL)
```



The prior for θ_1 is very close to its MLE, and we have a lot of data for θ_1 , so there isn't much change in our estimate as we shift the prior certainty. On the other hand, the prior for θ_2 is lower than its MLE, and we

don't have as much data for θ_2 , so as we make the prior weaker, there is a shift toward the MLE.

5.1

```
# data from https://www2.stat.duke.edu/~pdh10/FCBS/Exercises/  
# for some reason i couldn't extract the data directly  
school1 <- readr::read_lines('~dev/stats-hw/stat-s626/school1.dat') %>%  
  as.numeric()  
school2 <- readr::read_lines('~dev/stats-hw/stat-s626/school2.dat') %>%  
  as.numeric()  
school3 <- readr::read_lines('~dev/stats-hw/stat-s626/school3.dat') %>%  
  as.numeric()  
  
schools <- list(school1, school2, school3)
```

a

From class, we had:

- $\mu \mid \phi, y \sim \mathcal{N}(\mu_n, \frac{1}{\kappa_n \phi})$
- $\phi \mid y \sim \text{Gamma}(\nu_n/2, SS_n/2)$

We will use Monte Carlo estimation instead of evaluating integrals.

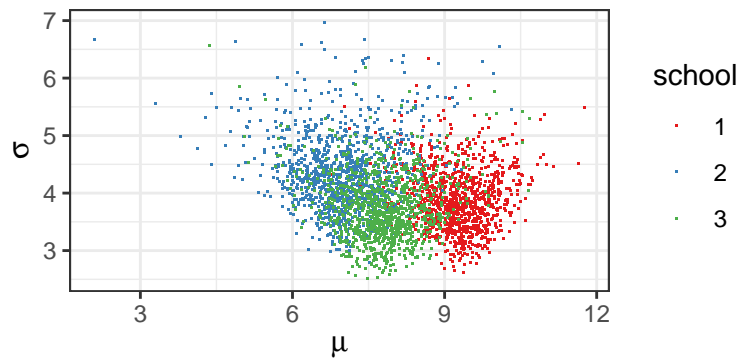
```
# MC params  
sample.size <- 2 ** 10  
  
# parameters  
mu.0 <- 5  
sigma2.0 <- 4  
kappa.0 <- 1  
nu.0 <- 2  
  
# transformed parameters  
ss.0 <- sigma2.0 * nu.0  
  
# summary statistics  
mc.df <- plyr::ldply(seq_along(schools), function(i) {  
  # posterior params  
  n <- length(schools[[i]])  
  kappa.n <- kappa.0 + n  
  mu.n <- (kappa.0 * mu.0 + sum(schools[[i]])) / kappa.n  
  nu.n <- nu.0 + n  
  y.bar <- mean(schools[[i]])  
  ss.n <-  
    ss.0 + sum((schools[[i]] - y.bar) ** 2) +  
    n * kappa.0 / kappa.n * (y.bar - mu.0) ** 2  
  
  # sample phi  
  phi <- rgamma(sample.size, nu.n / 2, ss.n / 2)  
  # sample mu  
  mu <- rnorm(sample.size, mu.n, 1 / kappa.n / phi)
```

```

dplyr::data_frame(school = i, mu = mu, phi = phi, i = seq(sample.size))
}, .parallel = TRUE)

ggplot(mc.df) +
  geom_point(aes(x = mu, y = phi ** -.5, colour = factor(school)),
             alpha = 1, shape = '.') +
  labs(x = expression(mu), y = expression(sigma), colour = 'school') +
  scale_colour_brewer(palette = 'Set1')

```



```

mc.df %>%
  dplyr::group_by(school) %>%
  dplyr::summarise(`E(\theta | y)` = mean(mu),
                   `E(\sigma | y)` = mean(phi ** -.5),
                   `lower(\theta | y)` = quantile(mu, .025),
                   `upper(\theta | y)` = quantile(mu, .975),
                   `lower(\sigma | y)` = quantile(phi ** -.5, .025),
                   `upper(\sigma | y)` = quantile(phi ** -.5, .975)) %>%
  dplyr::ungroup() %>%
  knitr::kable()

```

school	$E(\theta y)$	$E(\sigma y)$	$lower(\theta y)$	$upper(\theta y)$	$lower(\sigma y)$	$upper(\sigma y)$
1	9.277265	3.892939	8.014325	10.438970	2.962883	5.037124
2	6.957561	4.377755	5.212700	8.621927	3.280598	5.893623
3	7.793922	3.746218	6.212247	9.275824	2.818355	4.987352

b

```

# generate all six permutations
permutations <- gtools::permutations(3, 3, seq(3)) %>%
  {lapply(seq_len(nrow(.)), function(i) .[i, ])}

# compute probability for each permutation
plyr::ldply(permutations, function(permutation) {
  wide.df <- mc.df %>%
    dplyr::select(i, school, mu) %>%
    tidyr::spread(school, mu) %>%
    dplyr::select(-i)

```



```

p <- mean((wide.df[permutation[1]] < wide.df[permutation[2]]) &
          (wide.df[permutation[2]] < wide.df[permutation[3]]))

dplyr::data_frame(permutation = paste(permutation, collapse = ', '),
                  `~$P(\\theta_i < \\theta_j < \\theta_k | y)$` = p)
}, .parallel = TRUE) %>%
knitr::kable()

```

permutation	$P(\theta_i < \theta_j < \theta_k y)$
1, 2, 3	0.0039062
1, 3, 2	0.0019531
2, 1, 3	0.0419922
2, 3, 1	0.7412109
3, 1, 2	0.0107422
3, 2, 1	0.2001953

c

```

plyr::ldply(permutations, function(permutation) {
  wide.df <- mc.df %>%
    dplyr::group_by(school, i) %>%
    dplyr::mutate(y.tilde = rnorm(1, mu, phi ** -.5)) %>%
    dplyr::ungroup() %>%
    dplyr::select(i, school, y.tilde) %>%
    tidyr::spread(school, y.tilde) %>%
    dplyr::select(-i)

  p <- mean((wide.df[permutation[1]] < wide.df[permutation[2]]) &
            (wide.df[permutation[2]] < wide.df[permutation[3]]))

  dplyr::data_frame(permutation = paste(permutation, collapse = ', '),
                    `~$P(\\tilde{Y}_i < \\tilde{Y}_j < \\tilde{Y}_k | y)$` = p)
}, .parallel = TRUE) %>%
knitr::kable()

```

permutation	$P(\tilde{Y}_i < \tilde{Y}_j < \tilde{Y}_k y)$
1, 2, 3	0.1015625
1, 3, 2	0.0869141
2, 1, 3	0.1777344
2, 3, 1	0.2568359
3, 1, 2	0.1269531
3, 2, 1	0.2080078

d

```

posterior.df <- mc.df %>%
  dplyr::select(i, school, mu) %>%
  tidyr::spread(school, mu) %>%

```

```

dplyr::select(-i)

pred.posterior.df <- mc.df %>%
  dplyr::group_by(school, i) %>%
  dplyr::mutate(y.tilde = rnorm(1, mu, phi ** -.5)) %>%
  dplyr::ungroup() %>%
  dplyr::select(i, school, y.tilde) %>%
  tidyr::spread(school, y.tilde) %>%
  dplyr::select(-i)

p.posterior <- mean((posterior.df[1] > posterior.df[2]) &
  (posterior.df[1] > posterior.df[3]))
p.pred.posterior <- mean((pred.posterior.df[1] > pred.posterior.df[2]) &
  (pred.posterior.df[1] > pred.posterior.df[3]))

dplyr::data_frame(
  ` $P(\\theta_1 > \\theta_2, \\theta_1 > \\theta_3 | y)$` = p.posterior,
  ` $P(\\tilde{Y}_1 > \\tilde{Y}_2, \\tilde{Y}_1 > \\tilde{Y}_3 | y)$` =
    p.pred.posterior
) %>%
  knitr::kable()

```

$P(\theta_1 > \theta_2, \theta_1 > \theta_3 y)$	$P(\tilde{Y}_1 > \tilde{Y}_2, \tilde{Y}_1 > \tilde{Y}_3 y)$
0.9414062	0.4599609