

Exercise 7.3

(a)

```
#### Exercise 7.3
# data
bluecrab =
as.matrix(read.table(url("http://www2.stat.duke.edu/~pdh10/FCBS/Exercise
s/bluecrab.dat")))
orangecrab =
as.matrix(read.table(url("http://www2.stat.duke.edu/~pdh10/FCBS/Exercise
s/orangecrab.dat")))

# a
# blue crab
n = nrow(bluecrab)
ybar = colMeans(bluecrab)
Mu0 = c(ybar)
Sigma = cov(bluecrab)
S0 = Lambda0 = Sigma
nu0=4

# Gibbs Sampler
inv=solve
S = 10000
MU = matrix(NA, nrow=S, ncol=2)
SIGMA = matrix(NA, nrow = S, ncol = 4)

for(s in 1:S){
  # update MU
  Lambdan = inv(inv(Lambda0) + n*inv(Sigma))
  Mun = Lambdan %*% (inv(Lambda0) %*% Mu0 + n*inv(Sigma) %*% ybar)
  Mu = MASS::mvrnorm(n=1, Mun, Lambdan)

  # update Sigma
  Sn = S0 + (t(Y) - c(Mu)) %*% t(t(Y) - c(Mu))
  # notation -> Sn = S0 = Smu
  # Smu = sum(yi-mu)(y-mu)T
  Sigma = inv(rWishart(1, nu0+n, inv(Sn))[, , 1])

  MU[s,] = Mu
```

```

    SIGMA[s,] = c(Sigma)
  }

# orange crab

n = nrow(orange crab)
ybar = colMeans(orange crab)
Mu0 = c(ybar)
Sigma = cov(orange crab)
S0 = Lambda0 = Sigma
nu0=4

# Gibbs Sampler
inv=solve
S = 10000
MU1 = matrix(NA, nrow=S, ncol=2)
SIGMA1 = matrix(NA, nrow = S, ncol = 4)

for(s in 1:S){
  # update MU
  Lambdan = inv(inv(Lambda0) + n*inv(Sigma))
  Mun = Lambdan %*% (inv(Lambda0) %*% Mu0 + n*inv(Sigma) %*% ybar)
  Mu = MASS::mvrnorm(n=1, Mun, Lambdan)

  # update Sigma
  Sn = S0 + (t(Y) - c(Mu)) %*% t(t(Y) - c(Mu))
  # notation -> Sn = S0 = Smu
  # Smu = sum(yi-mu)(y-mu)T
  Sigma = inv(rWishart(1, nu0+n, inv(Sn))[, , 1])

  MU1[s,] = Mu
  SIGMA1[s,] = c(Sigma)
}

```

blue crab의 theta는 MU에, orange crab의 theta는 MU1에 저장됨!

(b)

```

disp = tail(1:S, S/2) # disp의 역할은 뭘까? - 5000번째부터 10000번째것까지만 사용
수렴하는 데 시간이 걸리기 때문!
title1 = "Posterior draws of blue crab size"
p1 = data.frame(btheta1 = MU[disp,1], btheta2 = MU[disp,2]) %>%
  ggplot(aes(x=btheta1, y=btheta2)) + geom_point(size = 0.5, color =
"steelblue")+

```

```

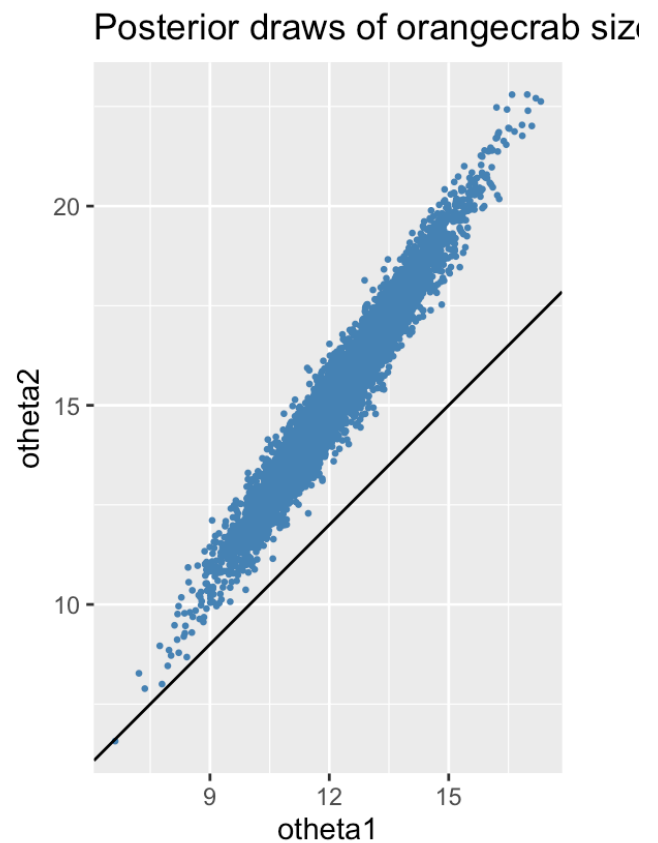
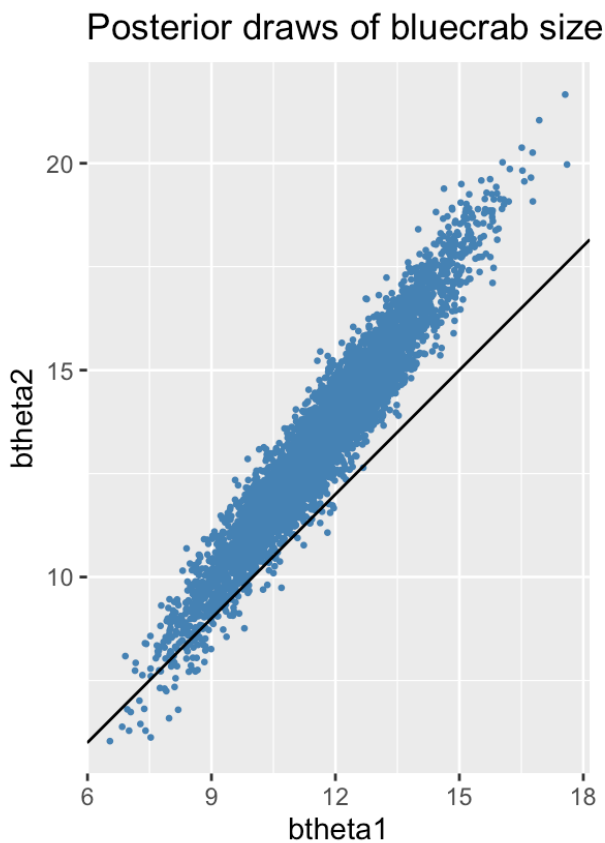
geom_abline(slope=1, intercept= 0)+
coord_fixed(ratio=1)+
labs(title=title1)

title2 = "Posterior draws of orangecrab size"
p2 = data.frame(otheta1 = MU1[disp,1], otheta2 = MU1[disp, 2]) %>%
  ggplot(aes(x=otheta1, y=otheta2))+ geom_point(size = 0.5, color =
"steelblue")+
  geom_abline(sloe =1, intercept = 0)+
  coord_fixed(ratio=1)+
  labs(title = title2)

ggarrange(p1, p2)

```

Blue crab과 Orange crab의 posterior theta 그래프



```

title3 = "Posterior draws of theta1 difference"
theta1_diff = MU[disp, 1] - MU1[disp, 1]
p3 = data.frame(theta1_diff = theta1_diff) %>%
  ggplot(aes(x=theta1_diff))+
  geom_histogram(color = "white", fill = "steelblue", bins = 30)+
  geom_vline(xintercept = 0)+
  labs(title=title3)

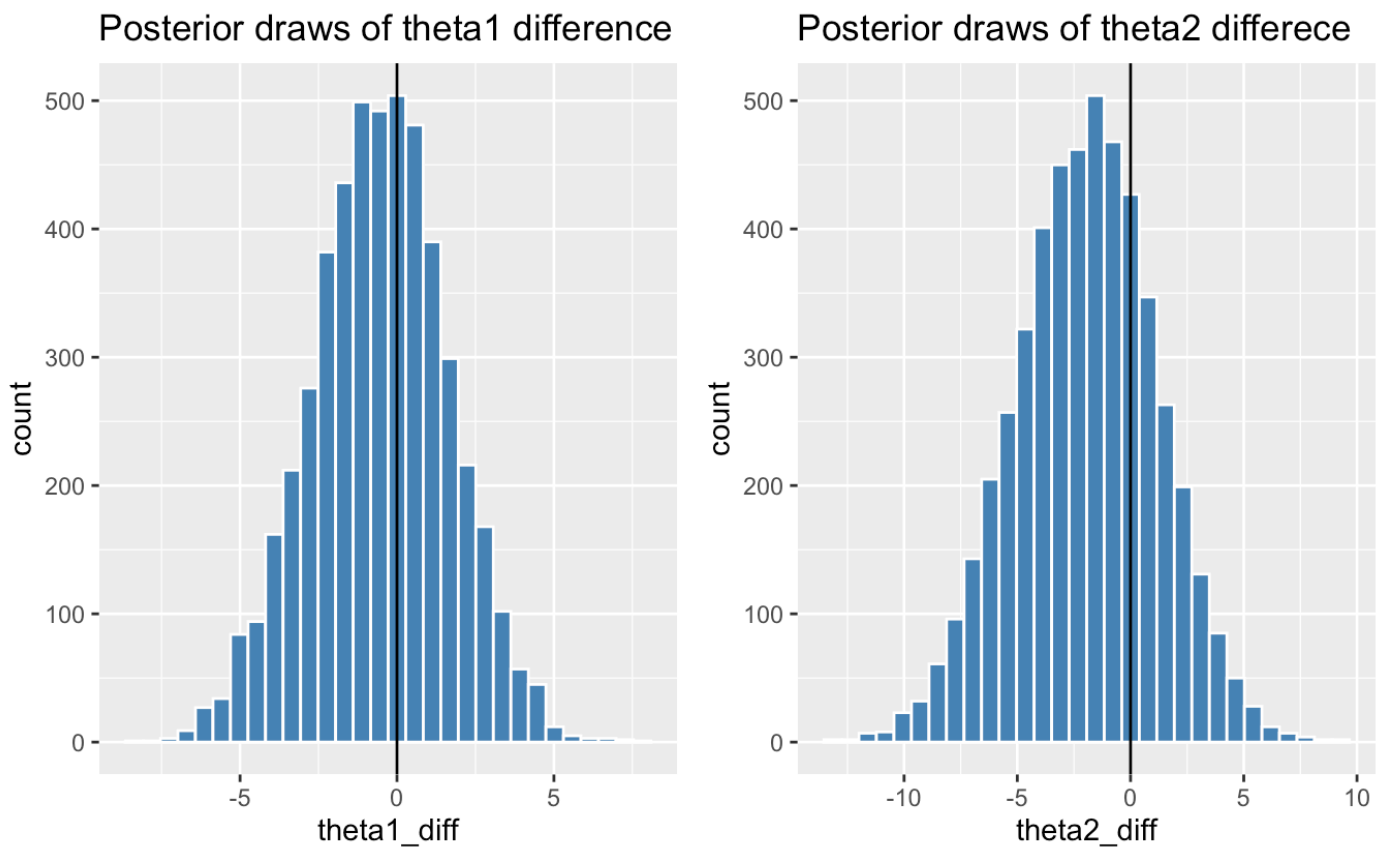
```

```

title4 = "Posterior draws of theta2 differece"
theta2_diff = MU[disp, 2] - MU1[disp, 2]
p4 = data.frame(theta2_diff = theta2_diff) %>%
  ggplot(aes(x=theta2_diff))+
    geom_histogram(color = "white", fill = "steelblue", bins = 30)+
    geom_vline(xintercept = 0)+
    labs(title=title4)

ggarrange(p3, p4)

```



살짝 0보다 왼쪽으로 치우쳐져 있는 것을 확인할 수 있다. 이는 orange crab의 theta가 더 큰 sample 이 많이 나왔음을 의미

```

mean(MU[disp, 1] > MU1[disp, 1])
[1] 0.4106
mean(MU[disp, 2] > MU1[disp, 2])
[1] 0.267

```

theta1, theta2 모두 orange crab이 더 큰 수가 많다.

(c)

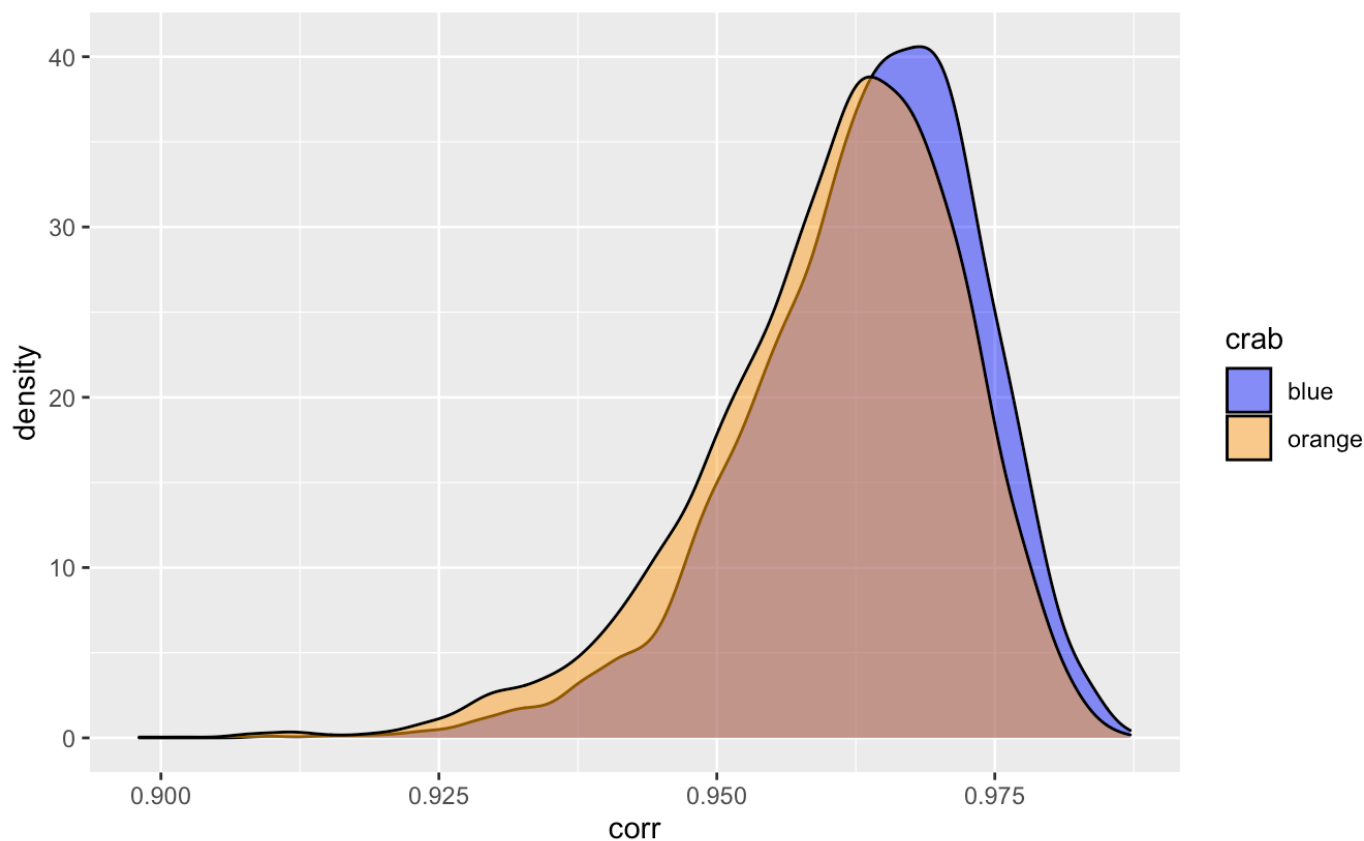
```
# c
bcorr = apply(SIGMA, MARGIN = 1, FUN = function(SIGMA){
  SIGMA[2] / sqrt(SIGMA[1]*SIGMA[4])
})
# apply(X, MARGIN = , FUN) MARGIN 1이면 row별 적용, 2이면 col별 적용

ocorr = apply(SIGMA1, MARGIN = 1, FUN = function(SIGMA){
  SIGMA[2] / sqrt(SIGMA[1]*SIGMA[4])
})

p5 = data.frame(crab = c(rep('blue', length(bcorr)/2), rep('orange',
length(ocorr)/2)),
               corr = c(bcorr[disp], ocorr[disp])) %>%
  ggplot(aes(x = corr, fill = crab))+
  geom_density(alpha = 0.5)+
  scale_fill_manual(values = c('blue', 'orange'))

ggarrange(p5)

mean(bcorr<ocorr)
[1] 0.439
```



blue crab의 경우가 orange crab 보다 correlation이 높은 sample이 더 많이 나왔다.

$P(\text{bcorr} < \text{ocorr} \mid y_b, y_o)$ 의 경우도 0.439로 그래프의 결과를 뒷받침한다. 이는 과제 예시 학술부장의 결과와 다르다.

흠...