HW4

Andrew Liang

9/28/2020

5.1

```r
#data
school <- list()
school[1] <- read.table("school1.dat")
school[2] <- read.table("school2.dat")
school[3] <- read.table("school3.dat")

#prior info
mu0 <- 5
s20 <- 4
k0 <- 1
v0 <- 2

n <- sapply(school, length)
ybar <- sapply(school, mean)
s2 <- sapply(school, var)
```

a)

```r
#posterior info
kn <- k0 + n
vn <- v0 + n
mun <- (k0*mu0 + n*ybar)/kn
s2n <- (v0*s20 + (n-1)*s2 + k0*n*(ybar-mu0)^2/kn)/vn

s.postsample <- s2.postsample <- theta.postsample <- matrix(0, 10000,
      3, dimnames = list(NULL, c("school1","school2","school3")))


#calculating theta and var values for each school
set.seed(1651)

for(i in 1:3){
  s2.postsample[,i] <- 1/rgamma(10000, vn[i]/2, s2n[i] * vn[i]/2)
  s.postsample[,i] <- sqrt(s2.postsample[,i])
  theta.postsample[,i] <- rnorm(10000, mun[i], s.postsample[,i]/sqrt(kn[i]))
}
```

```r
#posterior theta (means) and 95% CI for each school
colMeans(theta.postsample)
```

```
##  school1  school2  school3
## 9.290937 6.945155 7.818714
```

```r
apply(theta.postsample, 2, function(x){
  quantile(x, c(.025,.975))
})
```

```
##        school1  school2  school3
## 2.5%   7.78158 5.150979 6.188196
## 97.5% 10.81893 8.773321 9.413126
```

```r
#posterior sd and its 95% for each school
colMeans(s.postsample)
```

```
##  school1  school2  school3
## 3.911122 4.393559 3.751240
```

```r
apply(s.postsample, 2, function(x){
  quantile(x, c(.025,.975))
})
```

```
##        school1  school2  school3
## 2.5%  2.992594 3.346446 2.792604
## 97.5% 5.151195 5.890116 5.135844
```

The outputs above show the mean and the 95% confidence interval of the means and standard deviations for schools 1, 2, and 3.

b)

```r
library(combinat) #for permutations
```

```
##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##     combn
```

```r
#determine ranks for thetas
theta.ranks <- t(apply(theta.postsample, 1, rank))
rank.probs <- list()
for(p in permn(3)){
  index <- apply(theta.ranks, 1, function(row){
```

```r
    all(row == p)
  })
  rank.probs[[paste(p, collapse = ",")]] <- length(theta.ranks
                                                   [index,1])/10000
  #probability of permutations
}
```

In the probability matrix, the position indicates the school index, while the value indicates the theta rank(with 1 being the smallest, 3 being the largest); thus, it is not consistent with the subscript indexing int he problem. The listed probabilities are as follows:

$\theta_1 < \theta_2 < \theta_3$

```r
rank.probs[["1,2,3"]]
```

```
## [1] 0.0059
```

$\theta_1 < \theta_3 < \theta_2$

```r
rank.probs[["1,3,2"]]
```

```
## [1] 0.0033
```

$\theta_2 < \theta_3 < \theta_1$

```r
rank.probs[["3,1,2"]]
```

```
## [1] 0.6758
```

$\theta_3 < \theta_2 < \theta_1$

```r
rank.probs[["3,2,1"]]
```

```
## [1] 0.2183
```

$\theta_2 < \theta_1 < \theta_3$

```r
rank.probs[["2,1,3"]]
```

```
## [1] 0.0808
```

$\theta_3 < \theta_1 < \theta_2$

```r
rank.probs[["2,3,1"]]
```

```
## [1] 0.0159
```

c)

```r
#posterior prediction distribution
set.seed(1651)

pred.postsample <- matrix(0, 10000, 3, dimnames = list(NULL, c("school1","school2","school3")))
for(i in 1:3){
  pred.postsample[,i] <- rnorm(10000,mun[i],sqrt(s2.postsample[,i] * (1 + 1/kn[i])))
}

#determine ranks for predictions
pred.ranks <- t(apply(pred.postsample, 1, rank))
pred.probs <- list()
for (p in permn(3)) {
    index <- apply(pred.ranks, 1, function(row) {
        all(row == p)
    })
    pred.probs[[paste(p, collapse = ",")]] <- length(pred.ranks[index, 1])/10000
}
```

$\tilde{Y}_1 < \tilde{Y}_2 < \tilde{Y}_3$

```r
pred.probs[["1,2,3"]]
```

```
## [1] 0.1054
```

$\tilde{Y}_1 < \tilde{Y}_3 < \tilde{Y}_2$

```r
pred.probs[["1,3,2"]]
```

```
## [1] 0.1036
```

$\tilde{Y}_2 < \tilde{Y}_3 < \tilde{Y}_1$

```r
pred.probs[["3,1,2"]]
```

```
## [1] 0.271
```

$\tilde{Y}_3 < \tilde{Y}_2 < \tilde{Y}_1$

```r
pred.probs[["3,2,1"]]
```

```
## [1] 0.2004
```

$\tilde{Y}_2 < \tilde{Y}_1 < \tilde{Y}_3$

```r
pred.probs[["2,1,3"]]
```

```
## [1] 0.1795
```

$\tilde{Y}_3 < \tilde{Y}_1 < \tilde{Y}_2$

```r
pred.probs[["2,3,1"]]
```

```
## [1] 0.1401
```

d)

```r
sum(unlist(rank.probs[c("3,1,2","3,2,1")]))
```

```
## [1] 0.8941
```

Probability that $\theta_1 > \theta_2$ and θ_3 is listed above.

```r
sum(unlist(pred.probs[c("3,1,2","3,2,1")]))
```

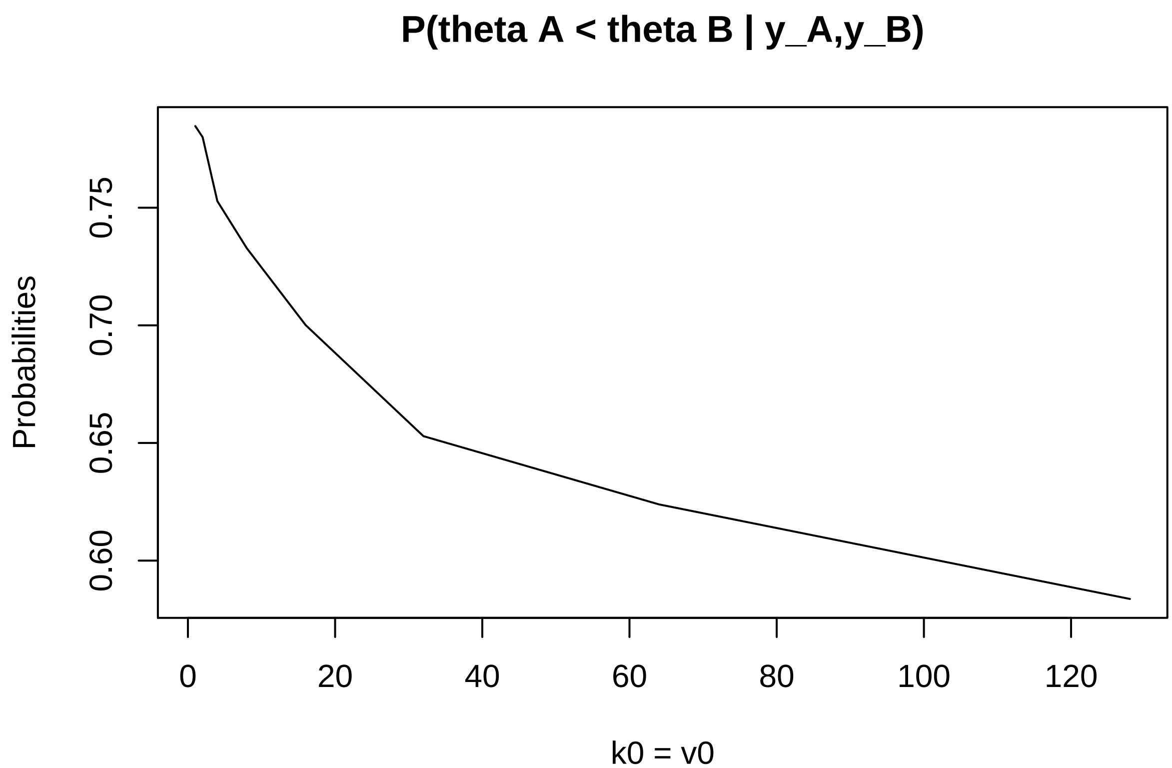
```
## [1] 0.4714
```

Probability that $\tilde{Y}_1 > \tilde{Y}_2$ and $\tilde{Y}_3$ is listed above.

5.2

```r
#prior info
mu0 <- 75
s20 <- 100
k0 <- c(1,2,4,8,16,32,64,128)
v0 <- c(1,2,4,8,16,32,64,128)

#sampling info
na <- nb <- n <- 16

ybara <- 75.2
sa <- 7.3

ybarb <- 77.5
sb <- 8.1

#posterior info
kn <- k0 + n
vn <- v0 + n

set.seed(1651)
postprob <- c()

for(i in 1:length(k0)){
  #MC sampling for A
  mu.a <- (k0[i]*mu0 + na*ybara)/kn[i]
  s2.a <- (v0[i]*s20 + (n-1)*(sa^2) + k0[i]*n*(ybara-mu0)^2/kn[i])/vn[i]
```

```
  s.postsample.a <- sqrt(1/rgamma(10000, vn[i]/2, s2.a * vn[i]/2))
  theta.postsample.a <- rnorm(10000, mu.a, s.postsample.a/sqrt(kn[i]))

  #MC sampling for B
  mu.b <- (k0[i]*mu0 + nb*ybarb)/kn[i]
  s2.b <- (v0[i]*s20 + (n-1)*(sb^2) + k0[i]*n*(ybara-mu0)^2/kn[i])/vn[i]
  s.postsample.b <- sqrt(1/rgamma(10000, vn[i]/2, s2.b * vn[i]/2))
  theta.postsample.b <- rnorm(10000, mu.b, s.postsample.b/sqrt(kn[i]))

  postprob[i] <- mean(theta.postsample.a < theta.postsample.b)
}

plot(k0, postprob, main = "P(theta A < theta B | y_A,y_B)",
     type = "l", xlab = "k0 = v0", ylab = "Probabilities")
```



From the plot above, we can see that as $\kappa_0 = \nu_0$ increases, the probability of $\theta_A < \theta_B | y_A, y_B$ decreases. We can think of κ_0, ν_0 as the prior sample size. Thus, if one has a very large/stronger sample size, they are less certain that $\theta_A < \theta_B | y_A, y_B$ is true, where as someone with a smaller/weaker prior sample size would be more certain of $\theta_A < \theta_B | y_A, y_B$