Stat 1651 HW 4

Rory Quinlan

2024-02-23

```
# Read Data
s1 <- scan("school1.dat")
s2 <- scan("school2.dat")
s3 <- scan("school3.dat")
```

Question 5.1 A.)

Posterior Means

```
# Using monte carlo simulation for all three schools (same as this (s1) from example code ch 5 p
g 76-78)

# Mu0, var0, k0, v0 given
mu0 <- 5
var0 <- 4
k0 <- 1
v0 <- 2

#posterior mean for s1
n1 <- length(s1)
y_bar1 <- mean(s1)
var1 <- var(s1)

mu_1 <- (k0 * mu0 + n1 * y_bar1)/(k0 + n1)
cat("Posterior mean for school 1 is:", mu_1)
```

```
## Posterior mean for school 1 is: 9.292308
```

```
#posterior mean for s2
n2 <- length(s2)
y_bar2 <- mean(s2)
var2 <- var(s2)

mu_2 <- (k0 * mu0 + n2 * y_bar2)/(k0 + n2)
cat("Posterior mean for school 2 is:", mu_2)
```

```
## Posterior mean for school 2 is: 6.94875
```

```
#posterior mean for s3
n3 <- length(s3)
y_bar3 <- mean(s3)
var3 <- var(s3)

mu_3 <- (k0 * mu0 + n3 * y_bar3)/(k0 + n3)
cat("Posterior mean for school 3 is:", mu_3)
```

```
## Posterior mean for school 3 is: 7.812381
```

95% CI

```
# 95% CI for s1 theta



set.seed(100)

vn1 <- v0 + n1
kn1 <- k0 + n1

s1n1 <- (1/vn1) * (v0 * var0 + (n1-1) * var1 + ((k0 * n1 )/kn1) * (y_bar1 - mu0)^2 )


s1_postsample <- 1/rgamma(10000, vn1/2, vn1 * s1n1 / 2)
theta1_postsample <- rnorm(10000, mu_1, sqrt(s1_postsample/(n1 + k0)))

cat("95% Confidence interval for s1 mean is",quantile(theta1_postsample,.025),"to",quantile(thet
a1_postsample,.975) )
```

```
## 95% Confidence interval for s1 mean is 7.760372 to 10.81625
```

```
# 95% CI for s1 sd

sd1_postsample <- sqrt(s1_postsample)

cat("95% Confidence interval for s1 sd is",quantile(sd1_postsample,.025),"to",quantile(sd1_posts
ample,.975) )
```

```
## 95% Confidence interval for s1 sd is 2.998103 to 5.155915
```

```r
# 95% CI for s2 theta

set.seed(100)

vn2 <- v0 + n2
kn2 <- k0 + n2

s2n2 <- (1/vn2) * (v0 * var0 + (n2-1) * var2 + ((k0 * n2 )/kn2) * (y_bar2 - mu0)^2 )


s2_postsample <- 1/rgamma(10000, vn2/2, vn2 * s2n2 / 2)
theta2_postsample <- rnorm(10000, mu_2, sqrt(s2_postsample/(n2 + k0)))

cat("95% Confidence interval for s2 mean is",quantile(theta2_postsample,.025),"to",quantile(thet
a2_postsample,.975) )
```

```
## 95% Confidence interval for s2 mean is 5.15876 to 8.71323
```

```r
# 95% CI for s2 sd

sd2_postsample <- sqrt(s2_postsample)

cat("95% Confidence interval for s2 sd is",quantile(sd2_postsample,.025),"to",quantile(sd2_posts
ample,.975) )
```

```
## 95% Confidence interval for s2 sd is 3.335093 to 5.873498
```

```r
# 95% CI for s3 theta

set.seed(100)

vn3 <- v0 + n3
kn3 <- k0 + n3

s3n3 <- (1/vn3) * (v0 * var0 + (n3-1) * var3 + ((k0 * n3 )/kn3) * (y_bar3 - mu0)^2 )


s3_postsample <- 1/rgamma(10000, vn3/2, vn3 * s3n3 / 2)
theta3_postsample <- rnorm(10000, mu_2, sqrt(s2_postsample/(n2 + k0)))

cat("95% Confidence interval for s3 mean is",quantile(theta3_postsample,.025),"to",quantile(thet
a3_postsample,.975) )
```

```
## 95% Confidence interval for s3 mean is 5.151665 to 8.753007
```

```r
# 95% CI for s3 sd

sd3_postsample <- sqrt(s3_postsample)

cat("95% Confidence interval for s3 sd is",quantile(sd3_postsample,.025),"to",quantile(sd3_posts
ample,.975) )
```

```
## 95% Confidence interval for s3 sd is 2.791296 to 5.11478
```

Question 5.1 B.)

```r
# Create parameter variables for function
small <- theta1_postsample
medium <- theta2_postsample
large <- theta3_postsample

# Create function that will find P(theta1 < theta2 < theta3)
# Using loop for each integer from 1 to 10000 check if medium is greater than small
# If it is then check if large is bigger than medium, if it is add it to sum variable
# then divide that amount by the total possible (10000) values
prob = function(small, medium, large){
  sum = 0
  for(i in 1:10000){
    if(medium[i] > small[i]){
      if(large[i] > medium[i]){
        sum = sum + 1
      }
    }
  }
  sum/10000
}



prob(small, medium, large)
```

```
## [1] 0.0018
```

```r
# Now that function is defined rearrange all possible iterations of small, medium, large which i
s 3! or 6 possibilities

#P(theta1 < theta3 < theta2)
prob(small, large, medium)
```

```
## [1] 0.0012
```

```
#P(theta2 < theta1 < theta3)
prob(medium, small, large)
```

```
## [1] 0.0215
```

```
#P(theta2 < theta3 < theta1)
prob(medium, large, small)
```

```
## [1] 0.4719
```

```
#P(theta3 < theta2 < theta1)
prob(large, medium, small)
```

```
## [1] 0.4814
```

```
#P(theta3 < theta1 < theta2)
prob(large, small, medium)
```

```
## [1] 0.0222
```

Question 5.1 C.)

```
set.seed(100)

# pred is is y tilde

sig1 <- sqrt(s1_postsample)
sig2 <- sqrt(s2_postsample)
sig3 <- sqrt(s3_postsample)

# generate 10000 possible values from normal distribution with mean theta and sd sig
pred1 <- rnorm(10000, small, sig1)
pred2 <- rnorm(10000, medium, sig2)
pred3 <- rnorm(10000, large, sig3)

#P(Y1 < Y2 < Y3)
prob(pred1, pred2, pred3)
```

```
## [1] 0.0881
```

```
#P(Y1 < Y3 < Y2)
prob(pred1, pred3, pred2)
```

```
## [1] 0.0962
```

```
#P(Y2 < Y3 < Y1)
prob(pred2, pred3, pred1)
```

```
## [1] 0.2706
```

```
#P(Y2 < Y1 < Y3)
prob(pred2, pred1, pred3)
```

```
## [1] 0.1449
```

```
#P(Y3 < Y1 < Y2)
prob(pred3, pred1, pred2)
```

```
## [1] 0.1577
```

```
#P(Y3 < Y2 < Y1)
prob(pred3, pred2, pred1)
```

```
## [1] 0.2425
```

Question 5.1 D.)

```
# take small vector and extract all values larger than medium and large
# take length to see how many values fit criteria
# Divide that by total possibilities (10000)
length(small[small>medium & small > large])/10000
```

```
## [1] 0.9533
```

```
length(pred1[pred1 > pred2 & pred1 > pred3])/10000
```

```
## [1] 0.5131
```

Question 5.2 .)

```
set.seed(100)

# variables given by problem
n <- 16
ya <- 75.2
sa <- 7.3

yb <- 77.5
sb <- 8.1

mu0 <-  75
s20 <- 100
k0 <- c(1,2,4,8,16,32,64)
v0 <- c(1,2,4,8,16,32,64)


kn <- k0 + n
vn <- v0 + n

data <- data.frame(k0 = k0, v0 = v0)

#calculate probabilites for each parameter value
# Create loop for each value in the length of k0
# Run a monte carlo simulation for group A (like from 5.1)
# Run a monte carlo simulation for group B (like from 5.1)
# Then take the mean of all times that theta_postsample_a is greater than theta_postsample_b
# Then concatenate with probs and add this to data from above
probs <- c()
for(i in 1:length(k0)){

#monte carlo simulation for A
  mu_na <- (k0[i] * mu0 + n * ya)/kn[i]
  var_na <- 1/vn[i] * (v0[i] * s20 + (n-1) * sa^2 + (k0[i] * n)/kn[i] * (ya - mu0)^2)
  s_postsample_a <- sqrt(1/rgamma(10000, vn[i]/2, var_na * vn[i]/2))
  theta_postsample_a <- rnorm(10000, mu_na, s_postsample_a/sqrt(kn[i]))

#monte carlo simulztion for B
  mu_nb <- (k0[i] * mu0 + n * yb)/kn[i]
  var_nb <- 1/vn[i] * (v0[i] * s20 + (n-1) * sb^2 + (k0[i] * n)/kn[i] * (yb - mu0)^2)
  s_postsample_b <- sqrt(1/rgamma(10000, vn[i]/2, var_nb * vn[i]/2))
  theta_postsample_b <- rnorm(10000, mu_nb, s_postsample_b/sqrt(kn[i]))

  #approximate probability theta_A < theta_B
  mean <- mean(theta_postsample_a < theta_postsample_b)
  probs <- c(probs, mean)
}
data$probabilites <- probs

# Plot findings
```

```
plot(x = v0, y = probs, xlab = "v0 = k0", ylab = "Probabilities", main = "P(theta_A < theta_B| y_A, y_B)", pch=19, lwd=2, col="darkgreen", type="b")
```