Homework 7

Jake Whalen

April 2, 2018

Table of Contents

[Problem 1 1](#_Toc510993603)

[Problem 2 3](#_Toc510993604)

[Problem 3 5](#_Toc510993605)

[Problem 4 7](#_Toc510993606)

suppressMessages(library(tidyverse))  
suppressMessages(library(coda))

# Problem 1

This problem concerns the HCB data from Assignment 6. For this problem, assume the observations are normally distributed with unknown depth-specific means and and common but unknown standard deviation .

#URL of data given in assignment  
www <- "http://www.biostat.umn.edu/~lynn/iid/wolf.river.dat"  
  
#Read in the data  
pollutants <- read.table(www, header=TRUE, stringsAsFactors = F)  
  
pollutants %<>%  
 mutate(id = rep(seq(1, 10, by = 1), 3)) %>%  
 select(-Aldrin) %>%  
 spread(Depth, HCB) %>%  
 select(-id, -Middepth)  
  
knitr::kable(pollutants)

|  |  |
| --- | --- |
| Bottom | Surface |
| 5.44 | 3.74 |
| 6.88 | 4.61 |
| 5.37 | 4.00 |
| 5.44 | 4.67 |
| 5.03 | 4.87 |
| 6.48 | 5.12 |
| 3.89 | 4.52 |
| 5.85 | 5.29 |
| 6.85 | 5.74 |
| 7.16 | 5.48 |

Assume that experts have provided the following prior information based on previous studies.

* The unknown means and are independent and normally distributed with mean and standard deviation . The unknown precision is independent of and and has a gamma distribution with shape and scale .
* Experts specified a 95% prior credible interval of [3, 9] for and . A good fit to this credible interval is obtained by setting the prior mean to and the prior standard deviation to .
* A 95% prior credible interval of [0.75, 2.0] is given for the unknown standard deviation . This translates to a credible interval of [0.25, 1.8] for . A good fit to this credible interval is obtained by setting the prior shape to and the prior scale to .

Find the following conditional distributions:

* The conditional distribution for given , P , and the observations.
  + The values are given for , and the observations ().
  + The equation below shows the given values being plugged in to solve for .
  + Then I solve for using the given values.
  + Thus the conditional distribution for given , P , and the observations is Normal(, ).
  + Since is unknown the parameters are dependent on the value of . Thus the exact distribution is not known.
* The conditional distribution for given , P, and the observations.
  + This distribution is solved similar to the surface theta above but replaces the observation values.
  + Then I solve for using the given values.
  + Thus the conditional distribution for given , P , and the observations is Normal(, ).
  + Since is unknown the parameters are dependent on the value of . Thus the exact distribution is not known.
* The conditional distribution for P given , , and the observations.
  + When finding the distribution for P we are given and .
  + These two values in addition to and the observations will help calculate the parameters for the Gama distribution of P.
  + First the parameter is calculated.
  + Since there is no P in the equation for this value is a constant and will remain the same during Gibbs sampling.
  + Then the parameter is calculated.
  + is dependent on P thus the value for it depends on the value of P.
  + The resulting distribution for P given , , and the observations is Gamma(14.5, ).

# Problem 2

Using the distributions you found in Part 1, draw 10,000 Gibbs samples of .

First I must define some of the variables that will be used to calculate the samples. I create two n values to represent the number of observation for both Surface (s) and Bottom (b).

nS <- length(pollutants$Surface)  
nB <- length(pollutants$Bottom)

Then I create variables to store the given values from Problem 1. This consists of the prior information the experts have provided.

#Theta  
mu.0 <- 6  
tau.0 <- 1.5  
#P  
alpha.0 <- 4.5  
beta.0 <- 0.19

Since I know is a constant from Problem 1 I define a variable for to be used later.

alpha.star <- alpha.0 + (nB + nS) / 2

Next I need to set some initial values for , and P. For both of the values I use the mean of the respective observations. For I take the standard deviation of all the observations combined. P has a known relationship with thus I use what I found for to initialize P.

thetaS <- c(mean(pollutants$Surface))  
thetaB <- c(mean(pollutants$Bottom))  
sigma <- c(sd(c(pollutants$Surface, pollutants$Bottom)))  
rho <- c(1 / sigma[1]^2)

The next step is to create vectors in R that will start with the prior values and receive updated values with each new Gibbs sample.

muS <- c(mu.0)  
muB <- c(mu.0)  
tauS <- c(tau.0)  
tauB <- c(tau.0)  
Beta <- c(beta.0)

I am now at the point where I will implement a loop to perform Gibbs sampling. First I define a variable that stores the number of simulations that I will run.

numSim <- 10000

Then I write a loop that calculates the parameters for each (, ) and passes them into the function rnorm(). The result is a single random value for . I add both the parameters and sample value for to the next spot in the respective vectors. Next I calculate a value for using the samples I just found. This new value and the constant are used in the rgamma() function to determine a new sample value for P. Then the value for P is used to calculate and both are stored in their respective vectors.

for(j in 2:numSim){  
 muS[j] <- ((mu.0/tau.0^2) + (rho[j-1] \* sum(pollutants$Surface))) / (tau.0^-2 + nS \* rho[j-1])  
 tauS[j] <- (tau.0^-2 + nS \* rho[j-1]) ^ (-0.5)  
 thetaS[j] <- rnorm(1, mean = muS[j], sd = tauS[j])  
   
 muB[j] <- ((mu.0/tau.0^2) + (rho[j-1] \* sum(pollutants$Bottom))) / (tau.0^-2 + nB \* rho[j-1])  
 tauB[j] <- (tau.0^-2 + nB \* rho[j-1]) ^ (-0.5)  
 thetaB[j] <- rnorm(1, mean = muB[j], sd = tauB[j])  
   
 Beta[j] <- (1/beta.0 + 1/2 \* sum((pollutants$Surface - thetaS[j])^2 + (pollutants$Bottom - thetaB[j])^2)) ^ -1  
 rho[j] <- rgamma(1, shape=alpha.star, scale=Beta[j])  
 sigma[j]<-1/sqrt(rho[j])  
}

Once the loop completes the Gibbs sampling step is complete. The next part of the question asks to estimate 90% credible intervals for , and .

thetaS.quant <- sprintf("[%s, %s]", round(quantile(thetaS, .05), 3), round(quantile(thetaS, 0.95), 3))  
thetaB.quant <- sprintf("[%s, %s]", round(quantile(thetaB, .05), 3), round(quantile(thetaB, 0.95), 3))  
sigma.quant <- sprintf("[%s, %s]", round(quantile(sigma, .05), 3), round(quantile(sigma, 0.95), 3))  
  
thetaBS <- thetaB - thetaS  
thetaBS.quant <- sprintf("[%s, %s]", round(quantile(thetaBS, .05), 3), round(quantile(thetaBS, 0.95), 3))

The table below shows the results for the calculations of these credible intervals.

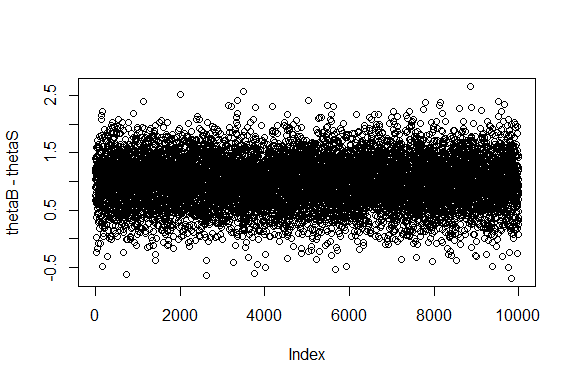
|  |  |
| --- | --- |
| Distribution | 90% Credible Interval |
| Theta S | [4.366, 5.343] |
| Theta B | [5.357, 6.327] |
| Sigma | [0.763, 1.193] |
| Theta(Bottom - Surface) | [0.29, 1.667] |

# 

# Problem 3

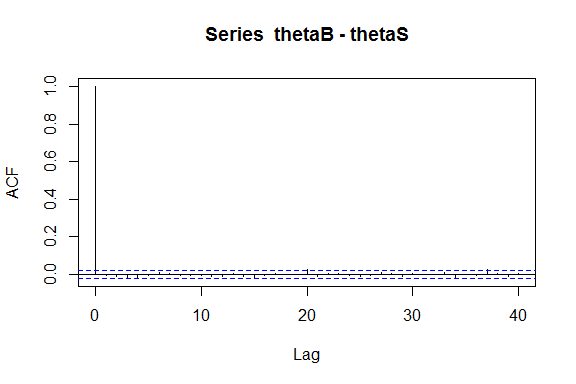
Do a traceplot of .

plot(thetaB - thetaS)



Find the autocorrelation function of .

acf(thetaB - thetaS)



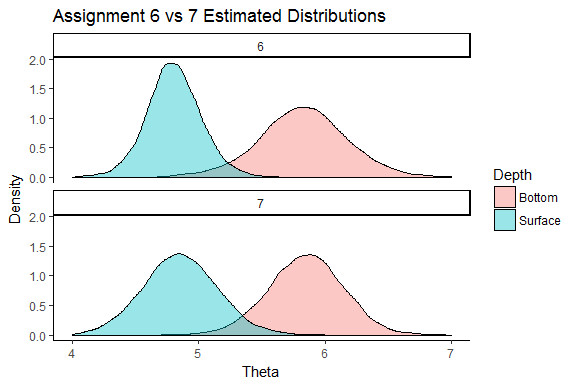
Find the effective sample size for your Monte Carlo sample for .

effectiveSize(thetaB - thetaS)

## var1   
## 10000

# Problem 4

The distributions found when using Gibbs sampling look like what I would have expected. The distributions are centered close to where they were in assignment 6 which used Monte Carlo to estimate the values. The spread of the two distributions are similar which was expected since the P value was used to estimate both the bottom and surface ’s with Gibbs.



Comparing with the Assignment 6 results above the results from Gibbs sampling in 7 match the Monte Carlo well. The only noticeable difference is the spread. Since the P value was based on all the observations together, the narrower spread for the Surface grew a little in the Gibbs sample. Likewise the spread for the bottom concentrations became slightly narrower because of the influence of the tighter Surface distribution. The code below shows the Monte Carlo credible intervals from assignment 6 being calculated.

thetaS.quant6 <- sprintf("[%s, %s]", round(quantile(p2\_mc('Surface')$thetaVals, .05), 3), round(quantile(p2\_mc('Surface')$thetaVals, 0.95), 3))  
thetaB.quant6 <- sprintf("[%s, %s]", round(quantile(p2\_mc('Bottom')$thetaVals, .05), 3), round(quantile(p2\_mc('Bottom')$thetaVals, 0.95), 3))  
sigma.quant6.S <- sprintf("[%s, %s]", round(quantile(p2\_mc('Surface')$sigmaVals, .05), 3), round(quantile(p2\_mc('Surface')$sigmaVals, 0.95), 3))  
sigma.quant6.B <- sprintf("[%s, %s]", round(quantile(p2\_mc('Bottom')$sigmaVals, .05), 3), round(quantile(p2\_mc('Bottom')$sigmaVals, 0.95), 3))  
sigma.quant6 <- sprintf("S %s, B %s", sigma.quant6.S, sigma.quant6.B)  
  
thetaBS6 <- p2\_mc('Bottom')$thetaVals - p2\_mc('Surface')$thetaVals  
thetaBS.quant6 <- sprintf("[%s, %s]", round(quantile(thetaBS6, .05), 3), round(quantile(thetaBS6, 0.95), 3))

The table below shows the 90% credible interval values for the results from assignment 6 (Monte Carlo) and 7 (Gibbs).

|  |  |  |
| --- | --- | --- |
| Distribution | Gibbs | Monte.Carlo |
| Theta S | [4.366, 5.343] | [4.45, 5.164] |
| Theta B | [5.357, 6.327] | [5.248, 6.428] |
| Sigma | [0.763, 1.193] | S [0.457, 1.039], B [0.742, 1.662] |
| Theta(Bottom - Surface) | [0.29, 1.667] | [0.355, 1.738] |

There is overall agreement between the two methods in regard to the values. The sigma credible intervals for monte Carlo surface and bottom concentrations fall slightly above and below the Gibbs sigma as expected.