Problem 1:

The conditional normal distributions for Thetas are given by the mean and the standard deviation . As functions of the other parameters, these are:

While the conditional gamma distributions for Rhos are given by the shape and scale parameters and . As functions of the other parameters, these are:

Solving for these using the priors gives the following conditional distributions, where the first two parameters are for a normal distribution, and the following two for a gamma distribution:

|  |  |  |
| --- | --- | --- |
|  | **Surface** | **Bottom** |
|  | 4.837 | 5.845 |
|  | 0.198 | 0.314 |
|  | 9.5 | 9.5 |
|  | 0.054 | 0.072 |

Code used for Problem 1:

wolf.surface=c(3.74, 4.61, 4.00, 4.67, 4.87, 5.12, 4.52, 5.29, 5.74, 5.48)

wolf.bottom=c(5.44, 6.88, 5.37, 5.44, 5.03, 6.48, 3.89, 5.85, 6.85, 7.16)

s\_n <- length(wolf.surface)

b\_n <- length(wolf.bottom)

s\_xbar <- mean(wolf.surface)

b\_xbar <- mean(wolf.bottom)

#Theta Parameters

mu0 <- 6

tau0 <- 1.5

#Rho parameters

alpha0 <- 4.5

beta0 <- 0.19

# Standard deviations of input data

b\_sigma0 <- sd(wolf.bottom)

s\_sigma0 <- sd(wolf.surface)

# Describing the precision as the inverse of the variance

b\_rho0 <- 1/b\_sigma0^2

s\_rho0 <- 1/s\_sigma0^2

b\_lambda0 <- 1/tau0^2

s\_lambda0 <- 1/tau0^2

# Conditional Distribution for Theta:

s\_mu1 <- ((mu0/tau0^2)+(sum(wolf.surface)/s\_sigma0)) /

((1/tau0^2)+(length(wolf.surface)/s\_sigma0))

b\_mu1 <- ((mu0/tau0^2)+(sum(wolf.bottom)/b\_sigma0)) /

((1/tau0^2)+(length(wolf.bottom)/b\_sigma0))

s\_tau1 <- (1/tau0^2 + length(wolf.surface)\*s\_rho0)^-.5

b\_tau1 <- (1/tau0^2 + length(wolf.bottom)\*b\_rho0)^-.5

# Conditional Distribution for Rho

b\_var\_theta <- sum((b\_n-b\_mu1)^2)

s\_var\_theta <- sum((s\_n-s\_mu1)^2)

b\_alpha1 <- alpha0 + b\_n/2

s\_alpha1 <- alpha0 + s\_n/2

b\_beta1 <- 1/(1/beta0 + b\_var\_theta/2)

s\_beta1 <- 1/(1/beta0 + s\_var\_theta/2)

Problem 2:

Drawing 10,000 Gibbs Samples from these distributions iteratively and taking the 90% credible interval provides the following quantile data:

|  |  |  |
| --- | --- | --- |
|  | **0.05** | **0.95** |
|  | 4.376 | 5.324 |
|  | 5.292 | 6.410 |
|  | 0.699 | 1.212 |
|  | 0.828 | 1.450 |
|  | 0.255 | 1.738 |

Drawing density plots to spot check these values yields the following:

![A picture containing screenshot

Description automatically generated]()![A picture containing map, screenshot

Description automatically generated]()

![A close up of a mans face

Description automatically generated]()

The quantiles look reasonable given these, and haven’t diverged unexpectedly far from the initial priors. We see that the 90% credible intervals for Thetas have very little overlap, while those for Sigmas are significantly overlapped. This is reflected in the density of the differences; it is largely distributed around 1, with only a small portion less than 0, indicating that in most trials Theta Bottom > Theta Surface.

Code used for Problem 2:

numSim <- 10000

# Initial guess for precision

s\_rGprev<-rgamma(1,shape=s\_alpha1,scale=s\_beta1)

b\_rGprev<-rgamma(1,shape=b\_alpha1,scale=b\_beta1)

for (k in 1:numSim) {

# reduces to xbar when tau = Inf

s\_muG = (mu0/tau0^2 + s\_n\*s\_rGprev\*s\_xbar) /

(1/tau0^2 + s\_n\*s\_rGprev)

b\_muG = (mu0/tau0^2 + b\_n\*b\_rGprev\*b\_xbar) /

(1/tau0^2 + b\_n\*b\_rGprev)

# reduces to sigmaGibbs/sqrt(n) when tau = Inf

s\_tauG = 1/sqrt(1/tau0^2 + s\_n\*s\_rGprev)

b\_tauG = 1/sqrt(1/tau0^2 + b\_n\*b\_rGprev)

# simulate new value for theta

s\_thetaGibbs[k] <- rnorm(1,mean=s\_muG,sd=s\_tauG)

b\_thetaGibbs[k] <- rnorm(1,mean=b\_muG,sd=b\_tauG)

# update scale given current theta

s\_betaG<-1/(1/beta0 + 0.5\*sum((wolf.surface-s\_thetaGibbs[k])^2))

b\_betaG<-1/(1/beta0 + 0.5\*sum((wolf.bottom-b\_thetaGibbs[k])^2))

# sample new value for rho

s\_rhoGibbs[k]<-rgamma(1,shape=s\_alpha1,scale=s\_betaG)

b\_rhoGibbs[k]<-rgamma(1,shape=b\_alpha1,scale=b\_betaG)

# calculate new value of sigma

s\_sigmaGibbs[k]<-1/sqrt(s\_rhoGibbs[k])

b\_sigmaGibbs[k]<-1/sqrt(b\_rhoGibbs[k])

# previous value of rho

s\_rGprev = s\_rhoGibbs[k]

b\_rGprev = b\_rhoGibbs[k]

}

# Calculate Quantiles

s\_thetaGibbs\_Q <- quantile(s\_thetaGibbs,c(0.05,0.95))

b\_thetaGibbs\_Q <- quantile(b\_thetaGibbs,c(0.05,0.95))

s\_sigmaGibbs\_Q <- quantile(1/sqrt(s\_rhoGibbs),c(0.05,0.95))

b\_sigmaGibbs\_Q <- quantile(1/sqrt(b\_rhoGibbs),c(0.05,0.95))

theta\_diff <- b\_thetaGibbs-s\_thetaGibbs

theta\_diff\_Q <- quantile(theta\_diff,c(0.05,0.95))

# print quantiles for report out

s\_thetaGibbs\_Q

b\_thetaGibbs\_Q

s\_sigmaGibbs\_Q

b\_sigmaGibbs\_Q

theta\_diff\_Q

# Plots to verify

plot(density(s\_thetaGibbs),col='blue',type='l',xlim=c(3.5,7.5),main='Θ Density')

lines(density(b\_thetaGibbs),col="green")

legend(6.2,1.4,c("Θ Surface ","Θ Bottom"),col=c("blue","green"),lty=c(1,1))

plot(density(1/sqrt(s\_rhoGibbs)),col='blue',type='l',xlim=c(0.5,2),ylim=c(0,3),main='Σ Density')

lines(density(1/sqrt(b\_rhoGibbs)),col="green")

legend(1.5,3,c("Σ Surface ","Σ Bottom"),col=c("blue","green"),lty=c(1,1))

plot(density(theta\_diff),col='blue',type='l',xlim=c(-1,3.5),main='Θ Differences')

Problem 3:

Checking for the simulation getting stuck in a local minima or maxima yields the chart below. It does not indicate any issues..

![A screenshot of a cell phone

Description automatically generated]()

Checking for autocorrelation in (Theta Bottom – Theta Surface) yields the chart below. In this case, there is very little autocorrelation among samples drawn, indicating chain convergence.

![A screenshot of a cell phone

Description automatically generated]()

The effective sample size for the differences in Thetas is 9,684. This indicates that it is a robust sample size, but perhaps not so large that we are wasting computational efforts.

Code used for Problem 3:

library(coda)

traceplot(as.mcmc(theta\_diff), main="Traceplot for Gibbs Sampling Theta\_B-Theta\_S")

acf(theta\_diff)

effectiveSize(theta\_diff)

Problem 4:

Repeating from Problem 3, the lack of autocorrelation indicates the chains have converged in the Gibbs Sampling simulation. The traceplot indicates there were no issues being stuck in localized areas of the space, and the effective sample size remains good for the difference between thetas.

As previously show, we just obtained the following values:

|  |  |  |
| --- | --- | --- |
|  | **0.05** | **0.95** |
|  | 4.376 | 5.324 |
|  | 5.292 | 6.410 |
|  | 0.699 | 1.212 |
|  | 0.828 | 1.450 |
|  | 0.255 | 1.738 |

From this, we can comfortably say that the mean bottom pollution of HCB in Wolf River is higher than the mean HCB pollution at the surface.

To compare with prior results from assignment six, let us remind ourselves of what we obtained previously:

|  |  |  |
| --- | --- | --- |
|  | **0.05** | **0.95** |
|  | 4.44 | 5.16 |
|  | 5.26 | 6.43 |
|  | 0.930 | 4.67 |
|  | 0.355 | 1.83 |

Converting the current data to Rho from Sigma yields:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0.05 (6)** | **0.05 (8)** | **0.95 (6)** | **0.95 (8)** |
|  | 4.44 | 4.38 | 5.16 | 5.32 |
|  | 5.26 | 5.29 | 6.43 | 6.41 |
|  | 0.930 | 0.681 | 4.67 | 2.048 |
|  | 0.355 | 0.476 | 1.83 | 1.459 |

Converting in the opposite direction to match the previous data to the current output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0.05 (6)** | **0.05 (8)** | **0.95 (6)** | **0.95 (8)** |
|  | 4.44 | 4.376 | 5.16 | 5.324 |
|  | 5.26 | 5.292 | 6.43 | 6.410 |
|  | 0.458 | 0.699 | 1.034 | 1.212 |
|  | 0.734 | 0.828 | 1.678 | 1.450 |

The Gibbs Sampling MCMC has nearly the same means, while the standard deviations are a bit higher, or the precisions are a bit lower. This is because it is sampling from a conditional distribution rather than a single posterior distribution.