**Problem 1**

The posterior density for mean is a T distribution, and the precision density is a gamma distribution. The set of parameters for mu, kappa, alpha, and beta, with an additional parameter, spread, being used as a function of kappa, alpha, and beta for the dispersion parameter in the T distribution. The hyperparameters and plots are shown below.

|  |  |  |
| --- | --- | --- |
| **Hyper Parameters** | **Surface** | **Bottom** |
| **µ\*** | 4.804 | 5.839 |
| **κ\*** | 10 | 10 |
| **α\*** | 4.5 | 4.5 |
| **β\*** | 0.557 | 0.216 |
| **Spread** | 0.19967 | 0.32065 |

A screenshot of a map

Description automatically generated

A picture containing screenshot

Description automatically generated

90% credible intervals:

|  |  |  |
| --- | --- | --- |
| **90% CI** | **0.05** | **0.95** |
| **Θs** | 4.437977 | 5.170023 |
| **Θb** | 5.251218 | 6.426782 |
| **Ρs** | 0.9266696 | 4.7151187 |
| **Ρb** | 0.3593438 | 1.8284283 |

The means appear to be different, whereas the precision for bottom and top appears to have significant overlap, when visually inspecting plots of these distributions and when examining just the confidence intervals.

**Code used for Problem 1**

#input the data

surface <- c(3.74,4.61,4.00,4.67,4.87,5.12,4.52,5.29,5.74,5.48)

bottom <- c(5.44,6.88,5.37,5.44,5.03,6.48,3.89,5.85,6.85,7.16)

# Create the hyperparameters for Surface

s\_alpha <- -1/2

s\_beta <- Inf

s\_mu <- 0

s\_k <- 0

s\_n <- length(surface)

surface\_ss <- 0

for (i in surface) {

surface\_ss <- surface\_ss + (i - mean(surface))^2

}

length(surface)

# Update the hyperparameters for Suface

s\_alpha\_star <- s\_alpha + s\_n/2

s\_beta\_star <- (1/s\_beta + 0.5\*surface\_ss + (s\_n\*s\_k)/(2\*(s\_n+s\_k)))^-1

s\_mu\_star <- (s\_k\*s\_mu + s\_n\*mean(surface)) / (s\_k+s\_n)

s\_k\_star <- s\_k + s\_n

s\_spread <- sqrt(1/(s\_k\_star\*s\_alpha\_star\*s\_beta\_star))

# Create the hyperparameters for Bottom

b\_alpha <- -1/2

b\_beta <- Inf

b\_mu <- 0

b\_k <- 0

b\_n <- length(bottom)

bottom\_ss <- 0

for (i in bottom) {

bottom\_ss <- bottom\_ss + (i - mean(bottom))^2

}

length(bottom)

# Update the hyperparameters for Bottom

b\_alpha\_star <- b\_alpha + b\_n/2

b\_beta\_star <- (1/b\_beta + 0.5\*bottom\_ss + (b\_n\*b\_k)/(2\*(b\_n+b\_k)))^-1

b\_mu\_star <- (b\_k\*b\_mu + b\_n\*mean(bottom)) / (b\_k+b\_n)

b\_k\_star <- b\_k + b\_n

b\_spread <- sqrt(1/(b\_k\_star\*b\_alpha\_star\*b\_beta\_star))

# Create the T distribution for Thetas

theta=seq(from=3, to=8, length=200)

s\_std = (theta - s\_mu\_star)/s\_spread # Transform to center 0 spread 1

s\_t\_dens = dt(s\_std,df=2\*s\_alpha\_star)/s\_spread

b\_std = (theta - b\_mu\_star)/b\_spread # Transform to center 0 spread 1

b\_t\_dens = dt(b\_std,df=2\*b\_alpha\_star)/b\_spread

# Plotting the T distribution

plot(theta, s\_t\_dens, type="l",col="red",

#ylim=c(0,0.33),

main="Posterior Density for Mean - unknown var",

xlab="Mean HCB", ylab="Probability Density")

lines(theta, b\_t\_dens, type="l",col="green")

legend(7,2,

c("Surface","Bottom"),col=c("red","green"),

lty=c(1,1))

# Create the Gamma Distribution for Rho

rho=seq(from=.0, to=6, length=200)

s\_g\_dens = dgamma(rho,shape=s\_alpha\_star,scale=s\_beta\_star)

b\_g\_dens = dgamma(rho,shape=b\_alpha\_star,scale=b\_beta\_star)

# P lot the Gamma distribution for Rho

plot(rho, s\_g\_dens, type="l",col="red",

ylim = c(0,1),

main="Posterior Density for Precision",

xlab = "Precision",ylab="Probability Density")

lines(rho, b\_g\_dens, type="l",col="green")

legend(4,1,

c("Surface","Bottom"),col=c("red","green"),

lty=c(1,1))

# Find the credible intervals

s\_quantile <- qt(0.95,2\*s\_alpha\_star)

b\_quantile <- qt(0.95,2\*b\_alpha\_star)

# Mean credible intervals:

s\_t\_ci <- c(s\_mu\_star-s\_quantile\*s\_spread, s\_mu\_star+s\_quantile\*s\_spread)

b\_t\_ci <- c(b\_mu\_star-b\_quantile\*b\_spread, b\_mu\_star+b\_quantile\*b\_spread)

# Precision credible intervals:

s\_g\_ci <- qgamma(c(0.05,0.95), shape=s\_alpha\_star, scale=s\_beta\_star)

b\_g\_ci <- qgamma(c(0.05,0.95), shape=b\_alpha\_star, scale=b\_beta\_star)

**Problem 2**

Sampling from the distributions in Problem 1 for a total of 10,000 examples yields a relatively good approximation of the original functions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **90% CI** | **0.05\_Original** | **0.05\_dMC** | **0.95\_Original** | **0.95\_dMC** |
| **Θs** | 4.437977 | 4.428835 | 5.170023 | 5.172087 |
| **Θb** | 5.251218 | 5.231000 | 6.426782 | 6.425441 |
| **Ρs** | 0.9266696 | 0.8909986 | 4.7151187 | 4.7744099 |
| **Ρb** | 0.3593438 | 0.3573785 | 1.8284283 | 1.8285048 |

**Code used for Problem 2**

# Direct Monte Carlo simulation

numSim <- 10000

rhoMC\_surface <- rgamma(numSim,shape=s\_alpha\_star,scale=s\_beta\_star)

thetaMC\_surface <- rnorm(numSim,mean=s\_mu\_star,sd=1/sqrt(s\_k\_star\*rhoMC\_surface))

rhoMC\_bottom <- rgamma(numSim,shape=b\_alpha\_star,scale=b\_beta\_star)

thetaMC\_bottom <- rnorm(numSim,mean=b\_mu\_star,sd=1/sqrt(b\_k\_star\*rhoMC\_bottom))

# Plotting the results

plot(thetaMC\_surface,rhoMC\_surface,col="red",pch=".",

xlim=c(3,8),

ylim=c(0,10),

main="Posterior Monte Carlo Sample for Mean & Precision",

xlab="Theta",

ylab="Rho")

points(thetaMC\_bottom,rhoMC\_bottom,col="green",pch=".")

legend(7,10,c("Surface","Bottom"),col=c("red","green"),lty=c(1,1))

# Estimating 90% Credible Intervals

rhoMC\_surface\_sorted <- sort(rhoMC\_surface)

thetaMC\_surface\_sorted <- sort(thetaMC\_surface)

rhoMC\_bottom\_sorted <- sort(rhoMC\_bottom)

thetaMC\_bottom\_sorted <- sort(thetaMC\_bottom)

# mean CIs from MC

thetaMC\_s\_ci <- c(thetaMC\_surface\_sorted[10000\*0.05],thetaMC\_surface\_sorted[10000\*0.95])

thetaMC\_b\_ci <- c(thetaMC\_bottom\_sorted[10000\*0.05],thetaMC\_bottom\_sorted[10000\*0.95])

# Precision CIs from MC

rhoMC\_s\_ci <- c(rhoMC\_surface\_sorted[10000\*0.05],rhoMC\_surface\_sorted[10000\*0.95])

rhoMC\_b\_ci <- c(rhoMC\_bottom\_sorted[10000\*0.05],rhoMC\_bottom\_sorted[10000\*0.95])

print(thetaMC\_s\_ci)

print(thetaMC\_b\_ci)

print(rhoMC\_s\_ci)

print(rhoMC\_b\_ci)

**Problem 3**

The probability that Θb is larger than Θs in the direct Monte Carlo is 0.9884. The probability that Ρb is larger than Ρs is 0.0934; however, we are being asked if Σb is larger than Σs. The relationship between Ρ and Σ is *Ρ = 1/Σ2*, thus the probability Σb > Σs = Ρb <= Ρs, or (1-0.0934), which is 0.9066.

**Code used for Problem 3**

# SD higher

rhoDiff <- rhoMC\_bottom-rhoMC\_surface

prob\_bottom\_rho\_higher <- sum(rhoDiff**>=**0)/length(rhoDiff)

# Means higher

thetaDiff <- thetaMC\_bottom-thetaMC\_surface

prob\_bottom\_theta\_higher <- sum(thetaDiff**>**0)/length(thetaDiff)

print(prob\_bottom\_theta\_higher)

print(1-prob\_bottom\_rho\_higher)

**Problem 4**

The results match the logical connections of the Monte Carlo to the original input data descriptive statistics; the means and standard deviations appeared to be different and the same result was obtained by further analysis. This is also reflected in the visualization below. The likelihood that the concentration of aldrin and hexachlorobenzene per liter at the bottom of the of the Wolf River in Tennessee are higher than those at the surface is 98.84%, and the likelihood that the standard deviation among observations from the bottom being higher than those at the surface is 90.66%. The assumption of normality is reasonable due to the large number of samples drawn from the binomial distribution of whether or not one is greater than the other with respect to the Monte Carlo data. For the original input data, the QQ plots below indicate normality.

In conclusion; the means and standard deviations are different for the surface and the bottom of the Wolf River in Tennessee for measurements of HCB and aldrin per liter.

A screenshot of a cell phone

Description automatically generated

Surface Data:

A close up of a map

Description automatically generated

Bottom Data:

A close up of a map

Description automatically generated

Code used for Problem 4:

qqnorm(surface)

qqline(surface)

qqnorm(bottom)

qqline(bottom)