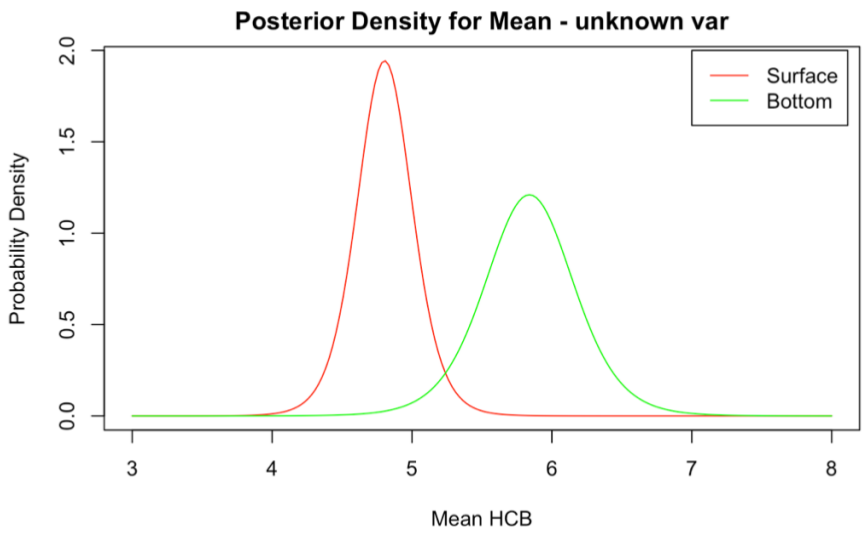
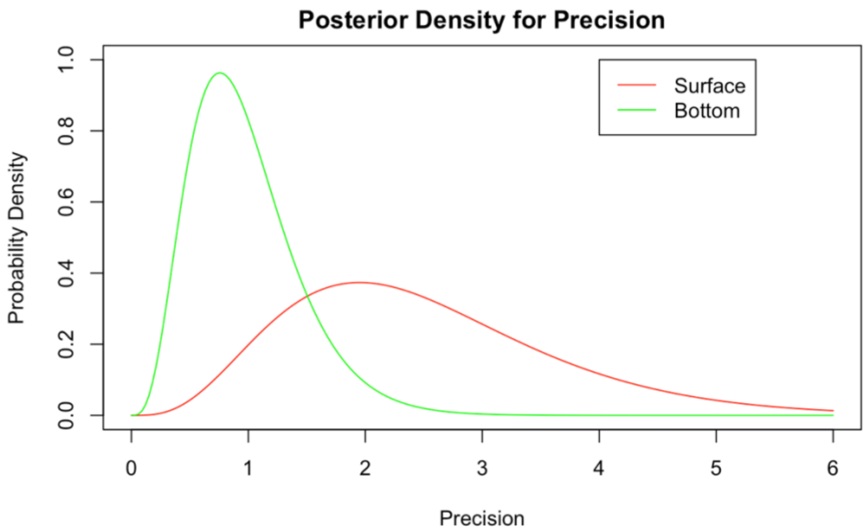
CSI-674 Jericho McLeod
Assignment 6 G00986513

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





90% credible intervals:

90% CI	0.05	0.95
Θ_s	4.437977	5.170023
Θ_b	5.251218	6.426782
P_s	0.9266696	4.7151187
P_b	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
Θ_s	4.437977	4.428835	5.170023	5.172087
Θ_b	5.251218	5.231000	6.426782	6.425441
P_s	0.9266696	0.8909986	4.7151187	4.7744099
P_b	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 Θ_b is larger than Θ_s in the direct Monte Carlo is 0.9884. The probability that P_b is larger than P_s is 0.0934; however, we are being asked if Σ_b is larger than Σ_s. The relationship between P and Σ is $P = 1/\Sigma_2$, thus the probability $\Sigma_b > \Sigma_s = P_b <= P_s$, or (1-0.0934), which is 0.9066.

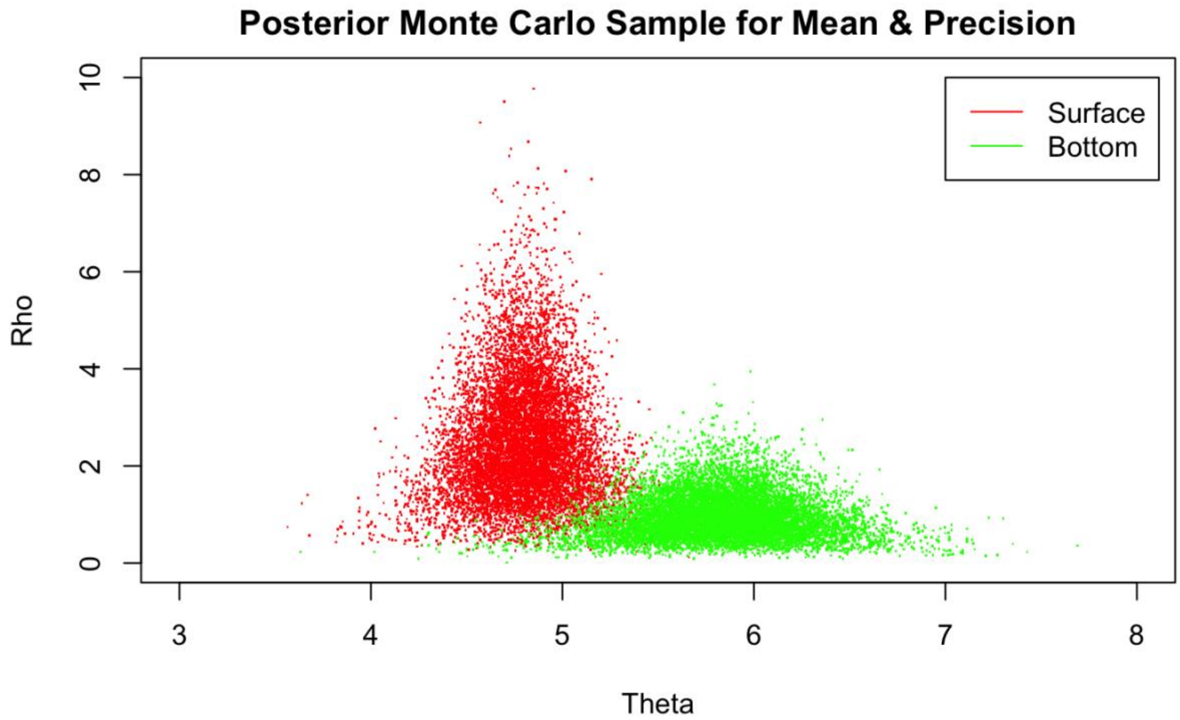
Code used for Problem 3

```
# 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)
```

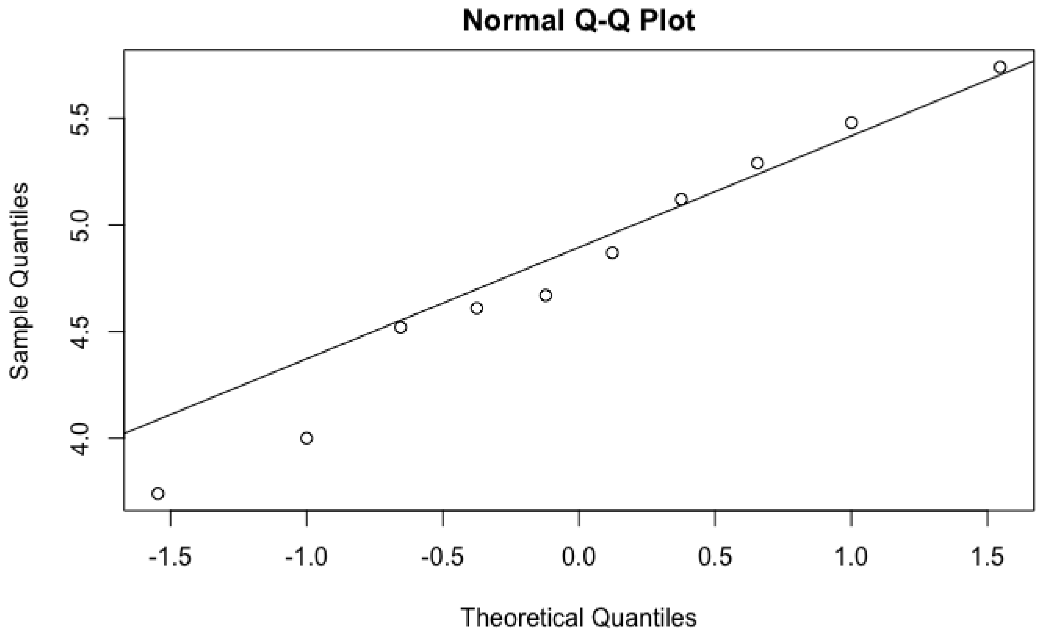
Problem 4

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.

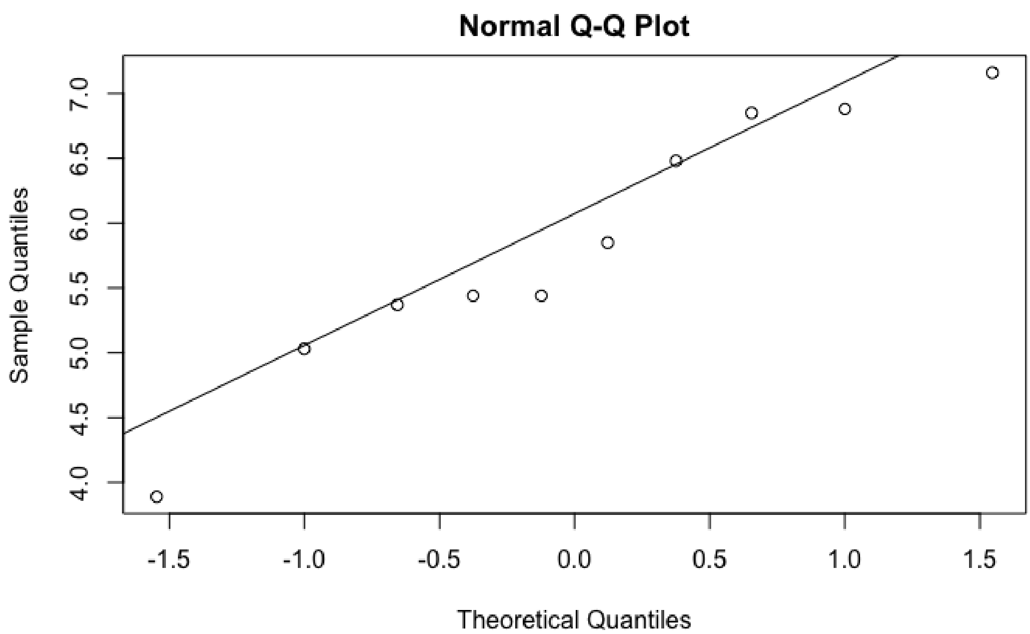
In conclusion; the means and standard deviations are different for the surface and the bottom of the Wolf River in Tennessee for measurements of HCB and aldrin per liter.



Surface Data:



Bottom Data:



Code used for Problem 4:

```
qqnorm(surface)
qqline(surface)
qqnorm(bottom)
qqline(bottom)
```