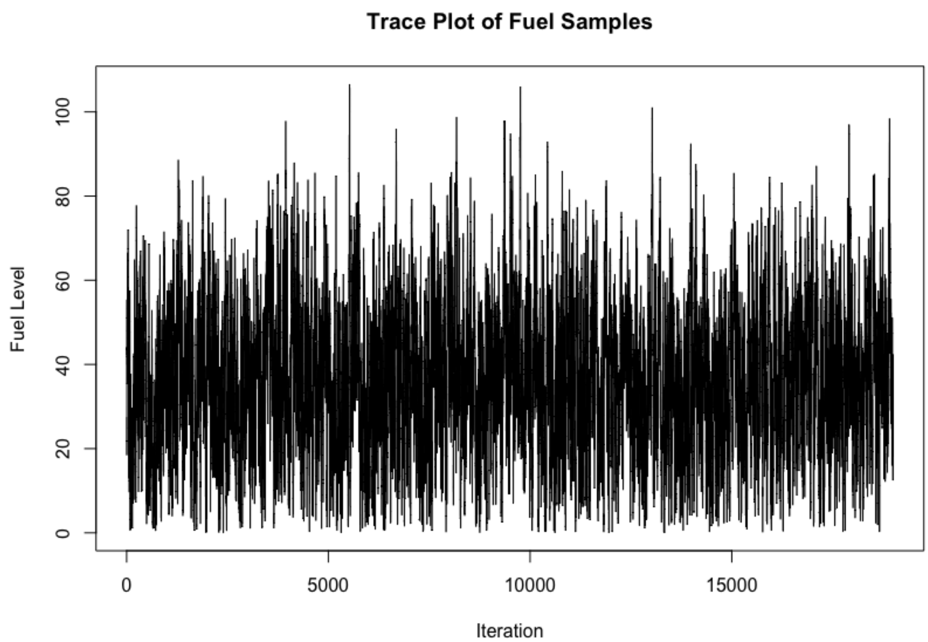
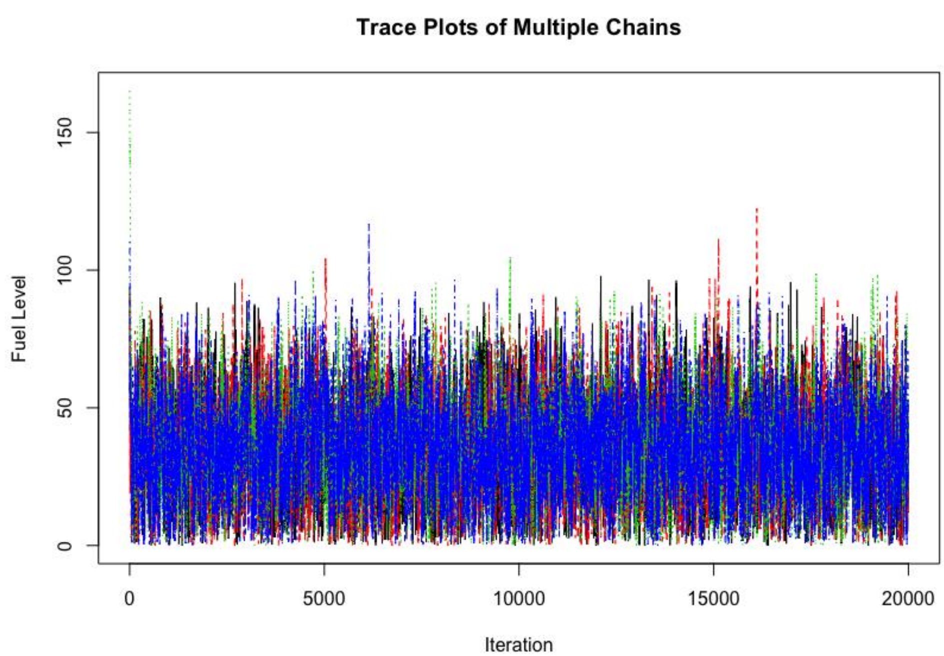
Questions

1. a. The numerical mode about the inference converges because as seen from the diagram below, the trace plot doesn't show any specific trend such as upward or downward pattern. When I ran multiple chains, I could discern that the chains appeared to converge to the same distribution. This is a significant sign of convergence.





b. I used Effective Sample Size as a numerical approach to assess the accuracy of the method from the previous question. Given 15,000 number of samples, I obtained an ESS of 1024. This represents 6.8% of the number of samples. This may not be sufficient as 10% is typically considered the reasonable minimum. The lower ESS may have been due to factors such as autocorrelation, a phenomenon whereby samples in the Hastings algorithm are correlated. This coincides with the error from the positive control. Using a positive control value of 44 liters, I obtained an error of 21.3 which is high for a fuel range of 0 to 180 liters.

c. Comparing the Hastings algorithm with the Bayes Monte Carlo(BMC), I observed a significant disparity in terms of error. I noticed that the BMC had a superior error of 15.42 while the Hastings had error of 21.3 for the same number of samples 15,000. This indicates that the BMC may require fewer runs to reach convergence as compared to Hastings.

Documentation

b. I decided to use both the Effective Sample Size and the positive control in order to obtain a bigger picture about the status of convergence for the Hastings algorithm. Ultimately, I could have used the positive control approach.

c. I believe my analysis is reproducible because I provided the seed for my random samplings which can be used to replicate my results.

e. I would like to reference sampling techniques employed from the Confidence Interval Estimation chapter of Verzani's "Simple R"

f. The code utilizes the GNU license. I assign copyright of this work to myself.

Appendix

```
library(ggplot2)
library(coda)

#PSet 4
tfuel <- 182
s_mu <- 34
s_sigma <- 20
userate <- 18
use_sd <- 2



#prior
prior <- function(fuel) {
  ifelse(fuel >= 0 & fuel <= tfuel, 1/tfuel, 0)
}

#likelihood
likely <- function(fuel, measurement) {
  dnorm(measurement, mean = fuel, sd = s_sigma)
}

#Metropolis Hastings
metropolis_hastings <- function(sim, initial_fuel, measurement) {
  samples <- numeric(sim)
  current_fuel <- initial_fuel
  samples[1] <- current_fuel

  for (i in 2:sim) {

    proposed_fuel <- rnorm(1, mean = current_fuel, sd = 10)
```

```
    if (proposed_fuel >= 0 && proposed_fuel <= tfuel) {
      acceptance_ratio <- (likely(proposed_fuel, measurement) * prior(proposed_fuel)) /
        (likely(current_fuel, measurement) * prior(current_fuel))

      if (runif(1) < min(1, acceptance_ratio)) {
        current_fuel <- proposed_fuel
      }
    }
    samples[i] <- current_fuel
  }
  return(samples)
}


#Implementing Hastings algorithm
sim = 15000
initial_fuel <- runif(1, 0, tfuel)
measurement <- s_mu
post_hastings <- metropolis_hastings(sim, initial_fuel, measurement)
#removing burn-in
burn_in = 1000  # Remove the first 1000 samples as burn-in
post_hastings <- post_hastings[(burn_in + 1):sim]



#assessing convergence visually
plot(post_hastings, type = "l", main = "Trace Plot of Fuel Samples", xlab = "Iteration", ylab =
"Fuel Level")

#assessing accuracy of convergence with effective sample size approach
```

```
effectiveSize(post_hastings)

#Assessing accuracy using positive control
true_fuel <- 44 #using sensor reading as positive control
simulated_measurement <- rnorm(1, mean = true_fuel, sd = s_sigma)

# Running Hastings with the simulated measurement
fuel_positive <- metropolis_hastings(sim, initial_fuel, simulated_measurement)
fuel_positivex <- fuel_positive[(burn_in + 1):sim]

# Estimating mode
d_positive <- density(fuel_positivex)
est_positive <- d_positive$x[which.max(d_positive$y)]

# Calculating error
error <- abs(est_positive - true_fuel)
cat("Error (Positive Control):", error, "\n")

#determining number of runs for convergence
chains = 4
fuel_chains <- matrix(NA, nrow = sim, ncol = chains)
for (i in 1:chains) {
  initial_fuel <- runif(1, 0, tfuel)
  fuel_chains[, i] <- metropolis_hastings(sim, initial_fuel, measurement)
}

#plotting
matplot(fuel_chains, type = "l", main = "Trace Plots of Multiple Chains", xlab = "Iteration", ylab
= "Fuel Level", col = 1:chains)

#comparison with BMC Samples
```

```r
set.seed(42)
sim = 15000
fuel_samples = runif(sim, 0, tfuel)
sensor_noise = rnorm(sim, mean = 0, sd = s_sigma)
measured_fuel = fuel_samples + sensor_noise

#Likelihood
likely_mc <- dnorm(measured_fuel, mean = s_mu, sd = s_sigma)
weights_mc <- likely_mc / sum(likely_mc)
post_mc <- sample(fuel_samples, size = sim, replace = TRUE, prob = weights_mc)

#Estimating mode and error
density_bmc <- density(post_mc)
est_bmc <- density_bmc$x[which.max(density_bmc$y)]
#results_bmc$estimated_mode[i] <- estimated_mode_bmc

# Check convergence criterion (example: tolerance on mode)
error_bmc <- abs(est_bmc - true_fuel) # true_fuel from your positive control
cat("BMC Error =", error_bmc, "\n")
```