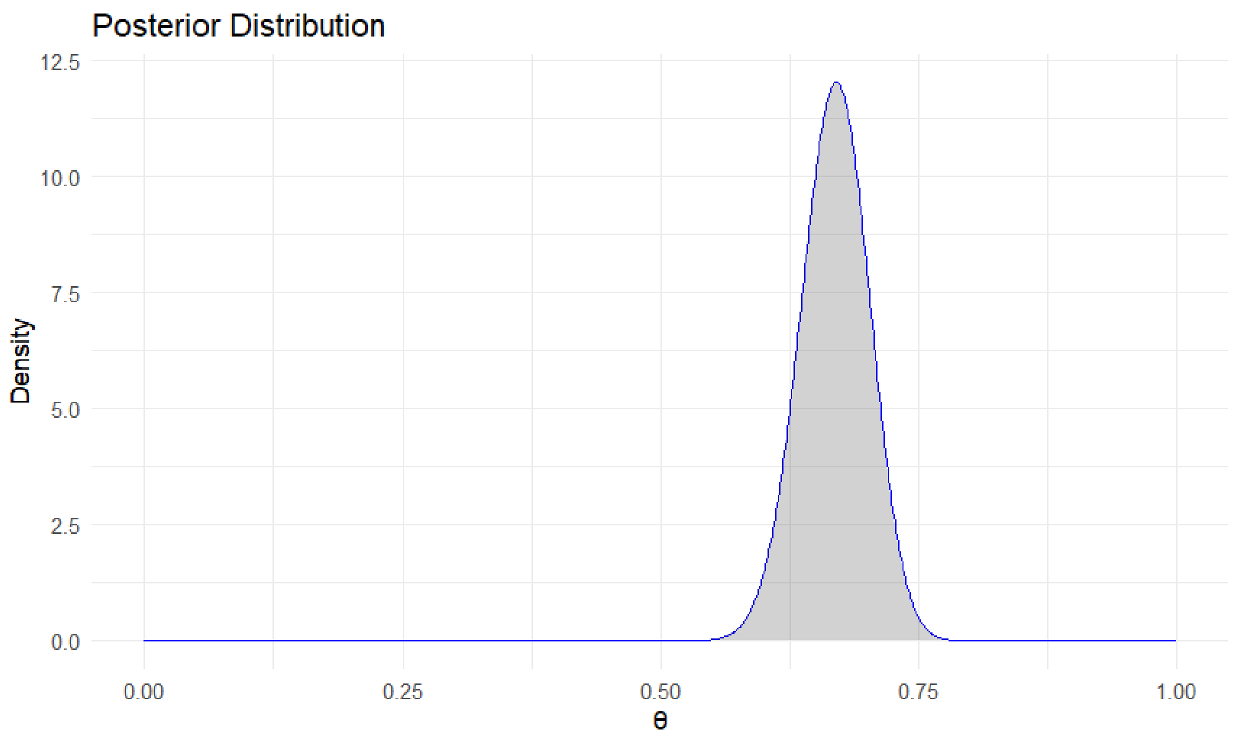
CGS698
Assignment-3

Anupam Chaudhary

(210170)

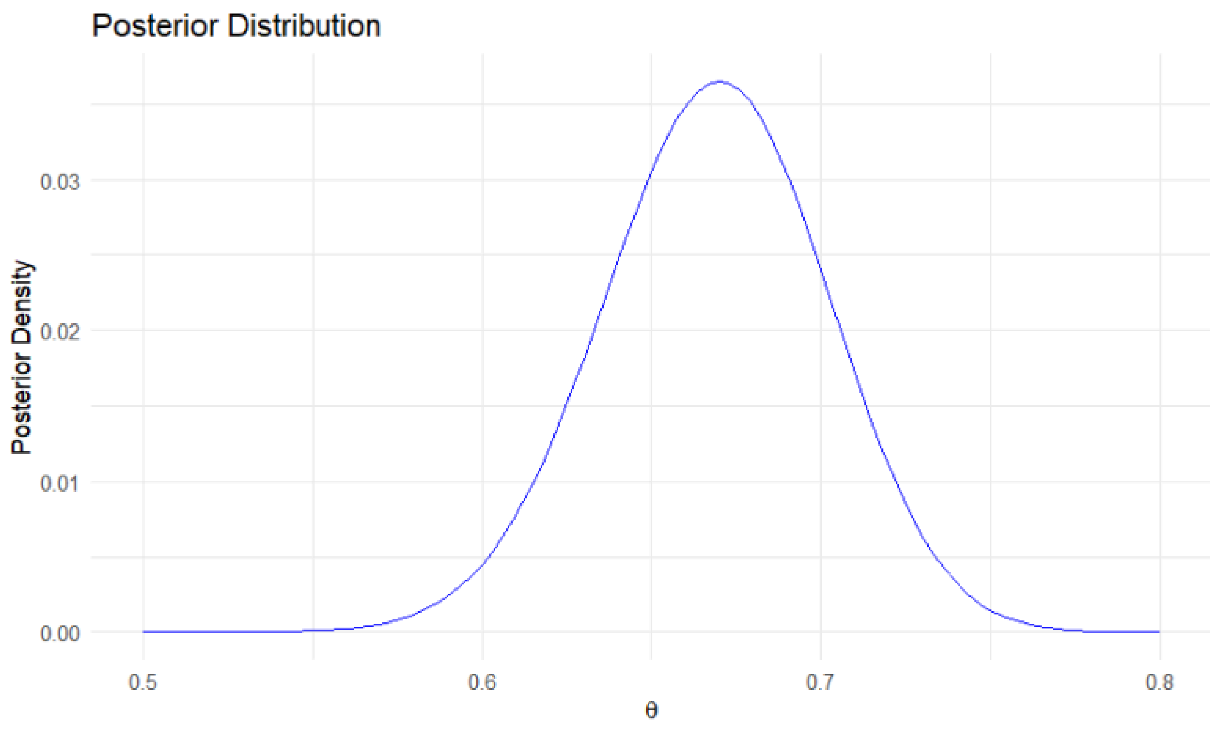
Part 1: Estimating the posterior distribution using different computational methods

1.
```
> library(ggplot2)
> alpha <- 135
> beta <- 67
> theta_values <- seq(0, 1, length.out = 1000)
> posterior_density <- dbeta(theta_values,alpha,beta)
> data <- data.frame(theta = theta_values, density = posterior_density)
> ggplot(data, aes(x = theta, y = density)) +
+  geom_line(color = 'blue') +
+  geom_area( alpha = 0.2) +
+  labs(title = 'Posterior Distribution ', x = 'θ', y = 'Density') +
+  theme_minimal()
```



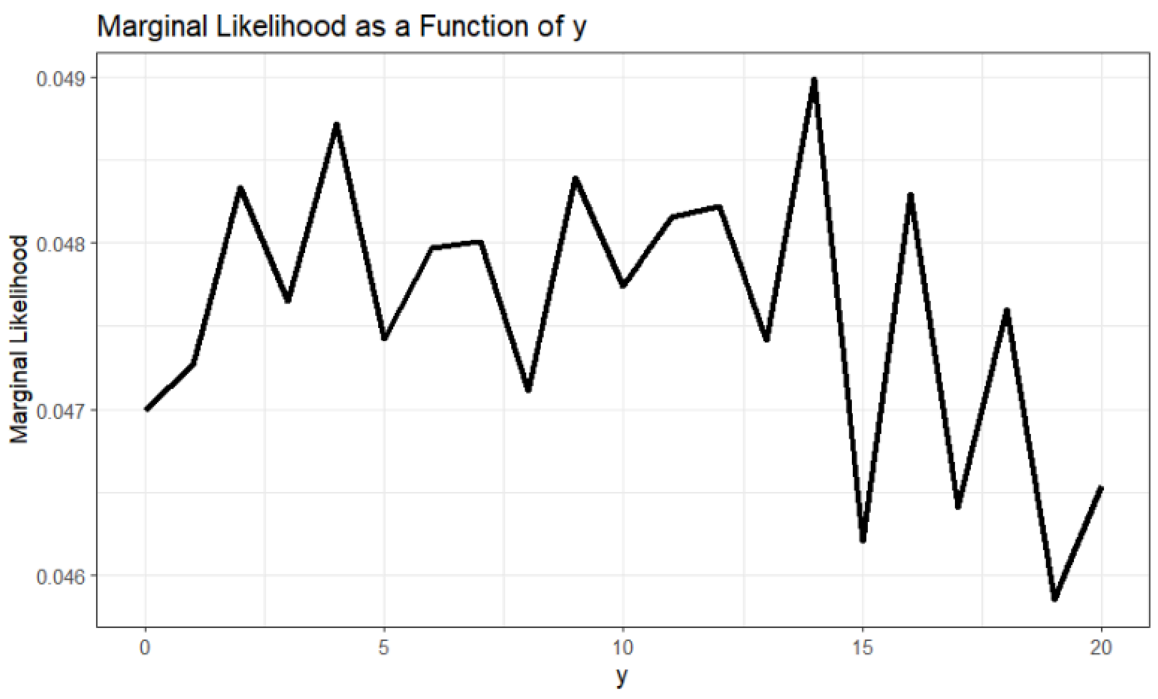
2.

```
> y <- c(10, 15, 15, 14, 14, 14, 13, 11, 12, 16)
> theta_grid <- seq(0.5, 0.8, length.out = 100)
> df.posterior <- data.frame(theta = theta_grid, likelihood = NA, prior = NA,
posterior =
+                              NA)
> for (i in 1:length(theta_grid)) {
+  likelihood <- prod(dbinom(y, size = 20, prob = theta_grid[i]))
+  prior <- dbeta(theta_grid[i], shape1 = 1, shape2 = 1)
+  df.posterior$likelihood[i] <- likelihood
+  df.posterior$prior[i] <- prior
+  df.posterior$posterior[i] <- likelihood * prior
+ }
> df.posterior$posterior <- df.posterior$posterior /
sum(df.posterior$posterior)
> ggplot(df.posterior, aes(x = theta, y = posterior)) +
+  geom_line(color = "blue") +
+  labs(title = "Posterior Distribution",
+       x = expression(theta),
+       y = "Posterior Density") +
+  theme_minimal()
```



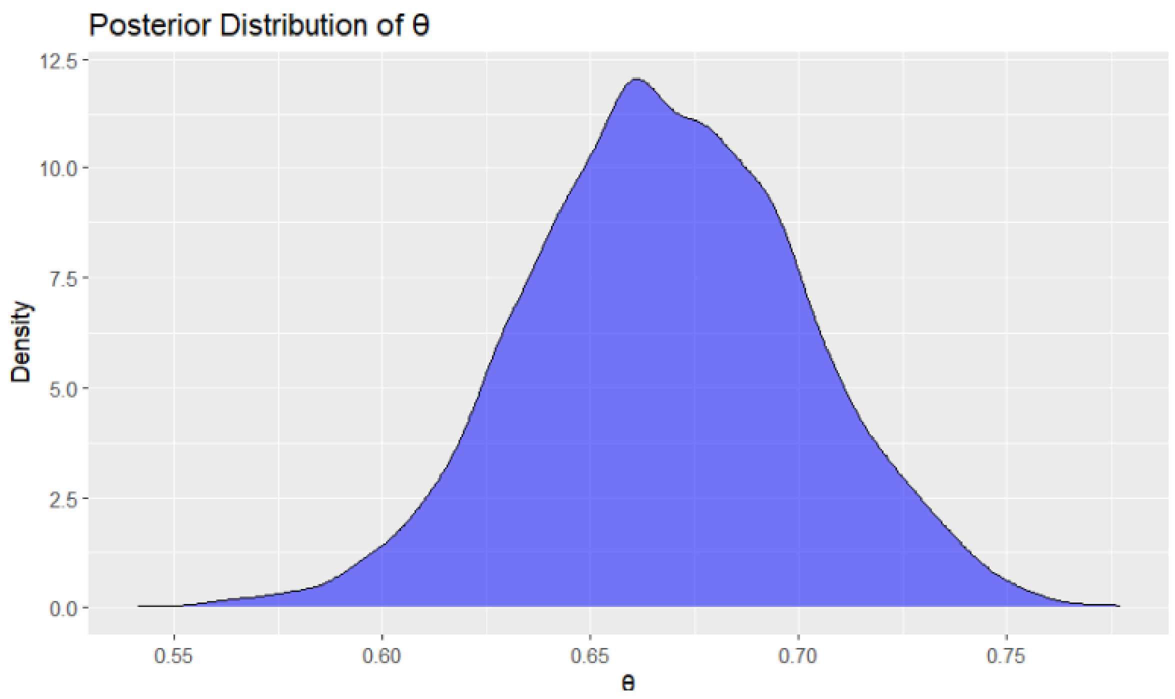
3.

```
> library(ggplot2)
> set.seed(123)
> y_values <- 0:20
> num_samples <- 10000
> estimate_marginal_likelihood <- function(y) {
+  df.estimate <- data.frame(matrix(ncol = 2, nrow = num_samples))
+  colnames(df.estimate) <- c("theta_sample", "likelihood")
+  for(i in 1:num_samples) {
+    theta_i <- rbeta(1, 1, 1)
+    likelihood <- dbinom(y, size = 20, prob = theta_i)
+    df.estimate[i, ] <- c(theta_i, likelihood)
+  }
+  ML <- mean(df.estimate$likelihood)
+
+  return(ML)
+ }
> ML_values <- sapply(y_values, estimate_marginal_likelihood)
> results <- data.frame(y = y_values, ML = ML_values)
> ggplot(results, aes(x = y, y = ML)) +
+  geom_line(size = 1.2) +
+  theme_bw() +
+  xlab("y") +
+  ylab("Marginal Likelihood") +
+  ggtitle("Marginal Likelihood as a Function of y")
```

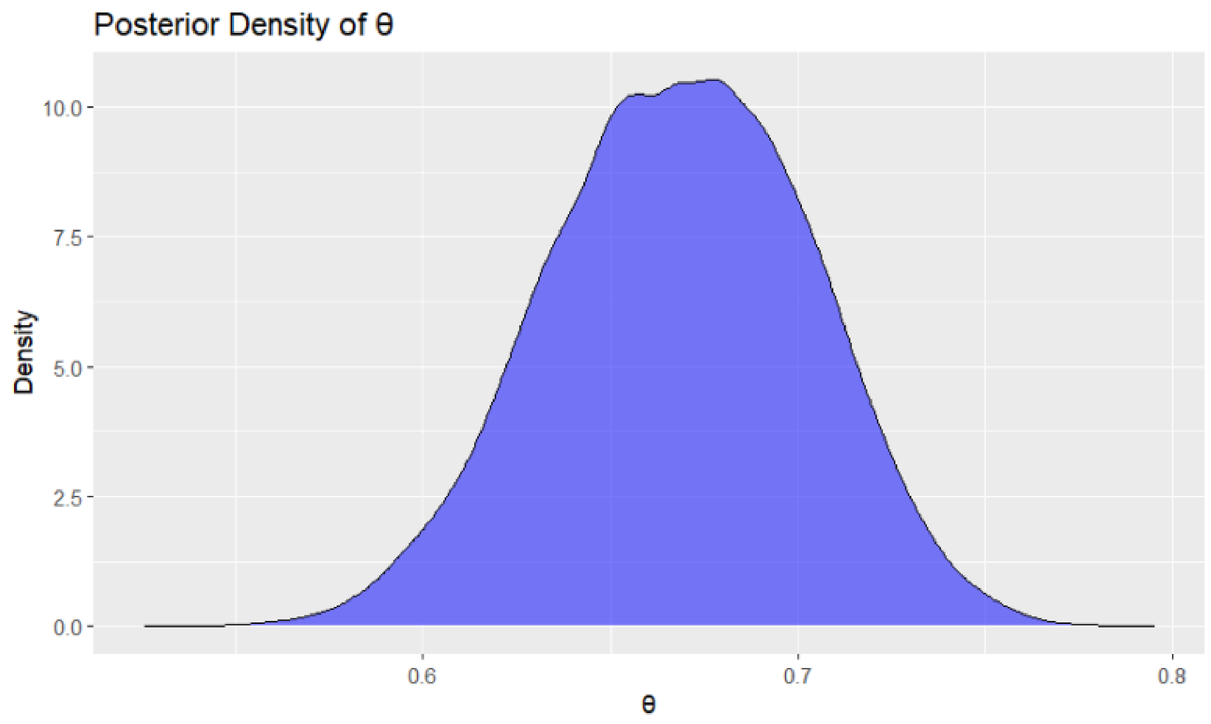


4.

```
> library(ggplot2)
> set.seed(123)
> y <- c(10, 15, 15, 14, 14, 14, 13, 11, 12, 16)
> n <- 20
> num_samples <- 10000
> df.estimate <- data.frame(matrix(ncol = 5, nrow = num_samples))
> colnames(df.estimate) <- c("theta_i", "likelihood", "prior", "proposal",
"posterior")
> for (i in 1:num_samples) {
+  while (theta_i < 0 || theta_i > 1) {
+    theta_i <- rnorm(1, 0.67, 0.2)
+  }
+  likelihood <- prod(dbinom(y, n, theta_i))
+  prior <- dbeta(theta_i, 1, 1)
+  posterior <- (likelihood * prior) / proposal
+  df.estimate[i, ] <- c(theta_i, likelihood, prior, proposal, posterior)
+ for (i in 1:num_samples) {
+  theta_i <- rnorm(1, 0.67, 0.2)
+  while (theta_i < 0 || theta_i > 1) {
+    theta_i <- rnorm(1, 0.67, 0.2)
+  }
+  likelihood <- prod(dbinom(y, n, theta_i))
+  prior <- dbeta(theta_i, 1, 1)
+  proposal <- dnorm(theta_i, 0.67, 0.2)
+  posterior <- (likelihood * prior) / proposal
+  df.estimate[i, ] <- c(theta_i, likelihood, prior, proposal, posterior)
+ }
+ df.estimate <- df.estimate[complete.cases(df.estimate), ]
+ df.estimate$posterior <- df.estimate$posterior / sum(df.estimate$posterior)
+ sample_size <- num_samples / 4
+ theta_samples <- sample(df.estimate$theta_i, size = sample_size, prob =
+                           df.estimate$posterior, replace = TRUE)
+ df.sample <- data.frame(theta_samples)
+ ggplot(df.sample, aes(x = theta_samples)) +
+  geom_density(fill = "blue", alpha = 0.5) +
+  labs(title = "Posterior Distribution of θ", x = "θ", y = "Density")
+ hist(theta_samples, breaks = 30, main = "Histogram of θ Samples", xlab = "θ",
col =
+       "blue", border = "black")
```



5.
```
> library(ggplot2)
> set.seed(123)
> y <- c(10, 15, 15, 14, 14, 14, 13, 11, 12, 16)
> n <- 20
> a <- 1
> b <- 1
> nsamp <- 50000
> theta_chain <- rep(NA, nsamp)
> theta_chain[1] <- rbeta(1, a, b)
> i <- 1
> step <- 0.08
> while (i < nsamp) {
+  proposal_theta <- rnorm(1, theta_chain[i], step)
+  if (proposal_theta > 0 & proposal_theta < 1) {
+    post_new <- sum(dbinom(y, n, proposal_theta, log = TRUE)) +
+      dbeta(proposal_theta, a, b, log = TRUE)
+    post_prev <- sum(dbinom(y, n, theta_chain[i], log = TRUE)) +
+      dbeta(theta_chain[i], a, b, log = TRUE)
+
+    Hastings_ratio <- exp(post_new + dnorm(theta_chain[i], proposal_theta,
step, log
+                                              = TRUE) -
+                             (post_prev + dnorm(proposal_theta, theta_chain[i],
step, log =
+                                                    TRUE)))
+
+     p_str <- min(Hastings_ratio, 1)
+
+     if (p_str > runif(1, 0, 1)) {
+       theta_chain[i + 1] <- proposal_theta
+       i <- i + 1
+     }
+   }
+ }
> theta_chain <- theta_chain[!is.na(theta_chain)]
> hist(theta_chain, breaks = 30, main = "Posterior Distribution of θ", xlab =
"θ", col =
+        "blue", border = "black")
> ggplot(data.frame(theta_chain), aes(x = theta_chain)) +
+   geom_density(fill = "blue", alpha = 0.5) +
+   labs(title = "Posterior Density of θ", x = "θ", y = "Density")
```



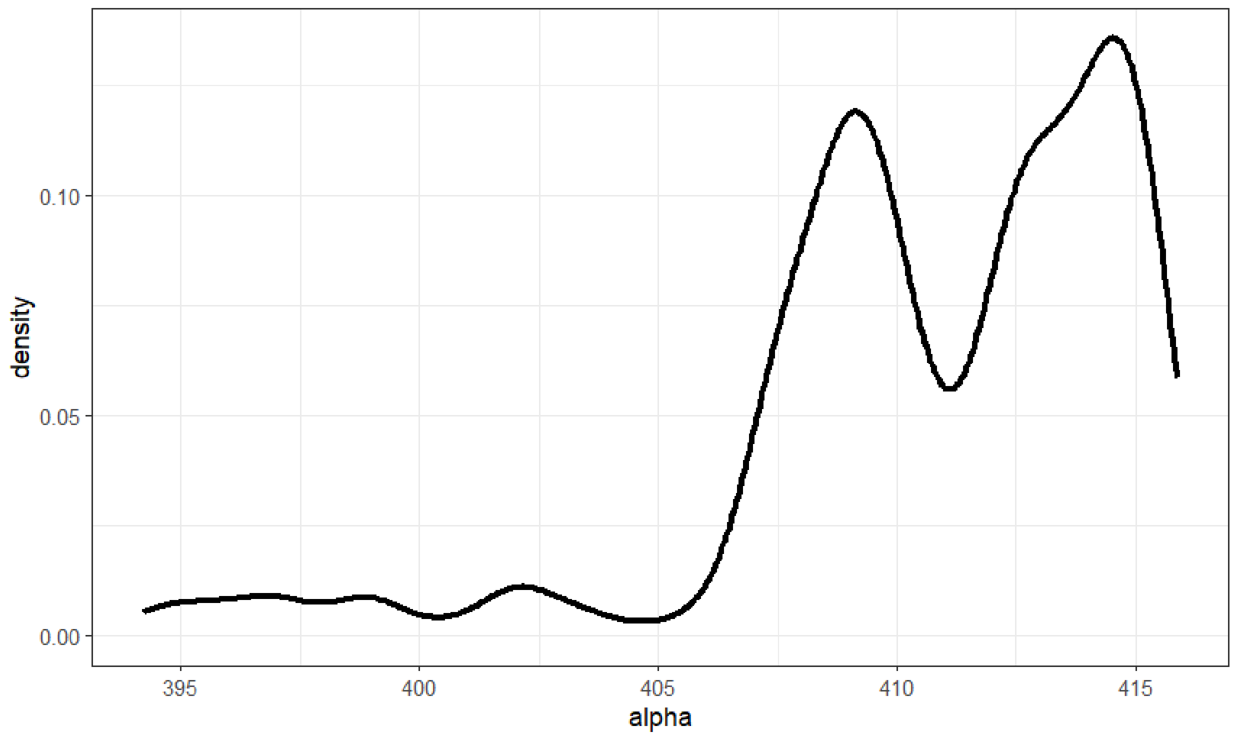
6.

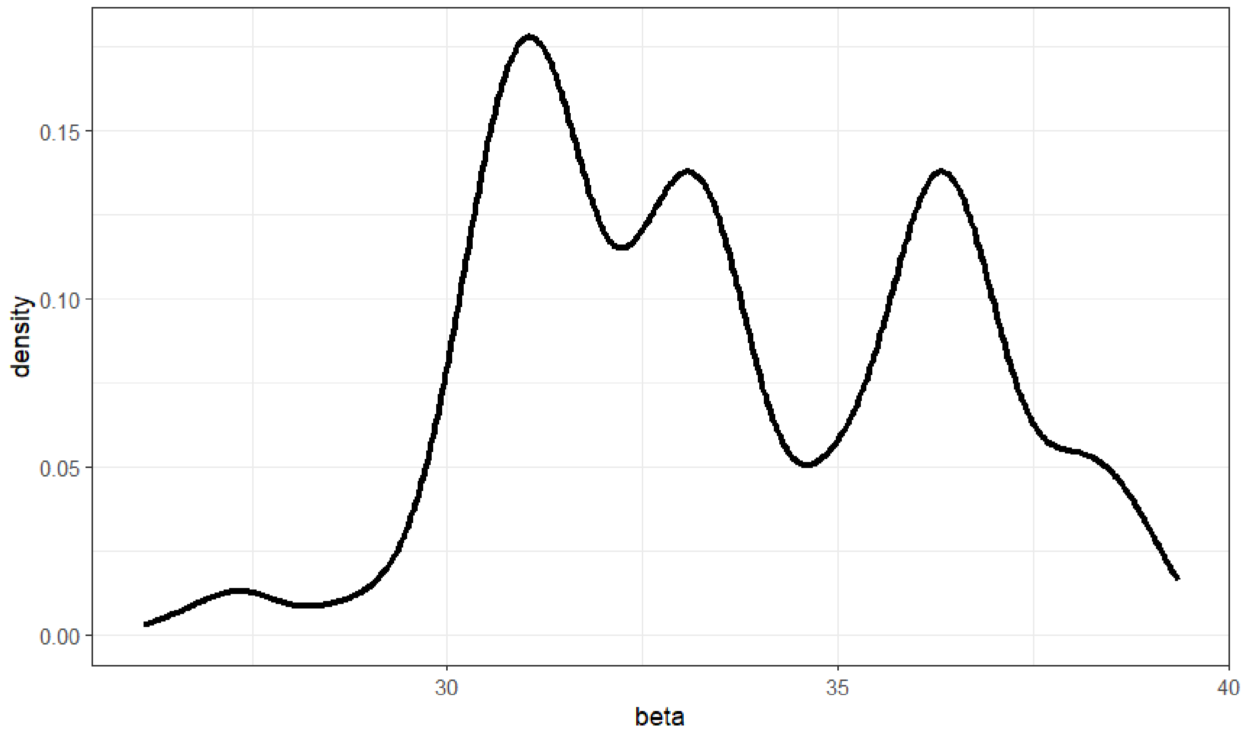
From above graphs,the posteriors from importance sampling and markov chain monte carlo have fatter tails as compared to analytical posterior

Part 2.5:

2.5.1:

```
>
> library(truncnorm)
> library(ggplot2)
> # Read the data from the corrected URL
> url <-
"https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main/notes/Data/wor
d-recognition-times.csv"
> dat <- read.table(url, sep = ",", header = TRUE)[, -1]
> # Parameters
> nsamp <- 4000
> sigma <- 30
> step <- 0.2
> # Initialize chains
> alpha_chain <- numeric(nsamp)
> beta_chain <- numeric(nsamp)
> type_chain <- integer(nsamp)
> # Initial values
> alpha_chain[1] <- rnorm(1, 400, 50)
> beta_chain[1] <- rtruncnorm(1, mean = 0, sd = 50, a = 0)
> type_chain[1] <- sample(0:1, 1)
> # Metropolis-Hastings Algorithm
> for (i in 2:nsamp) {
+  # Sample from proposal distributions
+  proposal_alpha <- rnorm(1, alpha_chain[i - 1], step)
+  proposal_beta <- rtruncnorm(1, mean = beta_chain[i - 1], sd = step, a = 0)
+  proposal_type <- sample(0:1, 1)
+
+  # Calculate mu
+  mu <- proposal_alpha + proposal_type * proposal_beta
+
+  # Calculate log-posterior for the new and previous states
+  log_posterior_new <- sum(dnorm(dat$RT, mu, sigma, log = TRUE)) +
+    dnorm(proposal_alpha, 400, 50, log = TRUE) +
+    proposal_type * log(dtruncnorm(proposal_beta, mean = 0, sd = 50, a = 0))
+
+  log_posterior_prev <- sum(dnorm(dat$RT, alpha_chain[i - 1] + type_chain[i -
1] * beta_chain[i - 1], sigma, log = TRUE)) +
+    dnorm(alpha_chain[i - 1], 400, 50, log = TRUE) +
+    type_chain[i - 1] * log(dtruncnorm(beta_chain[i - 1], mean = 0, sd = 50, a
= 0))
+
+  # Calculate Hastings ratio and acceptance probability
+  hastings_ratio <- exp(log_posterior_new - log_posterior_prev)
+  acceptance_prob <- min(1, hastings_ratio)
+
+  # Accept or reject the proposal
+  if (runif(1) < acceptance_prob) {
+    alpha_chain[i] <- proposal_alpha
+    beta_chain[i] <- proposal_beta
+    type_chain[i] <- proposal_type
+  } else {
+    alpha_chain[i] <- alpha_chain[i - 1]
+    beta_chain[i] <- beta_chain[i - 1]
+    type_chain[i] <- type_chain[i - 1]
+  }
+ }
> # Quantiles for alpha and beta
> print(quantile(alpha_chain, probs = c(0.025, 0.975)))
    2.5%    97.5%
397.0015 415.5064
> print(quantile(beta_chain, probs = c(0.025, 0.975)))
    2.5%    97.5%
28.67353 38.62055
> # Histogram for beta_chain
> hist(beta_chain, main = "Histogram of Beta Chain", xlab = "Beta")
> # Posterior distributions
> posteriors <- data.frame(alpha = alpha_chain, beta = beta_chain)
> # Density plot for alpha
> ggplot(posteriors, aes(x = alpha)) +
+  geom_density(size = 1.2) +
+  theme_bw() + xlab("alpha") +
+  theme(legend.title = element_blank(), legend.position = "top")
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.
> # Density plot for beta
> ggplot(posteriors, aes(x = beta)) +
+  geom_density(size = 1.2) +
+  theme_bw() + xlab("beta") +
+  theme(legend.title = element_blank(), legend.position = "top")
```





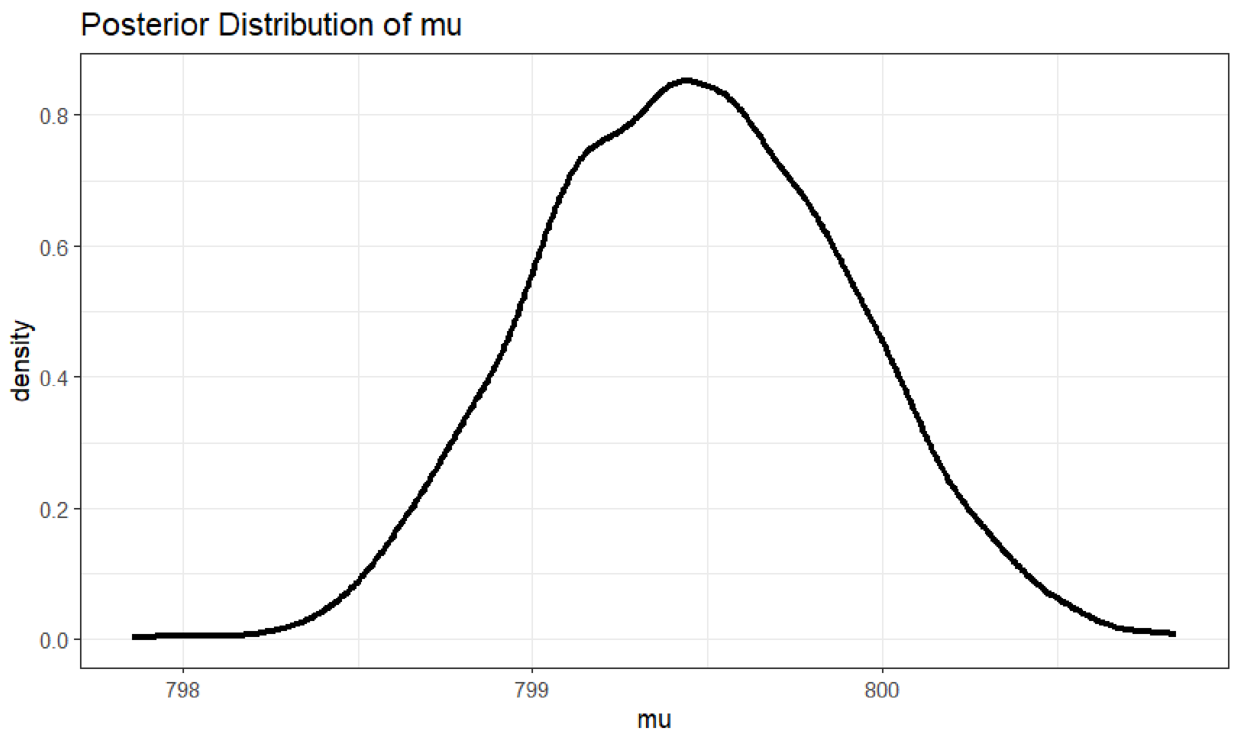
2.5.2:
For alpha,Quantile range is (395.2903 ,405.3474)
And for beta,Quantile range is (31.53942,45.00786)

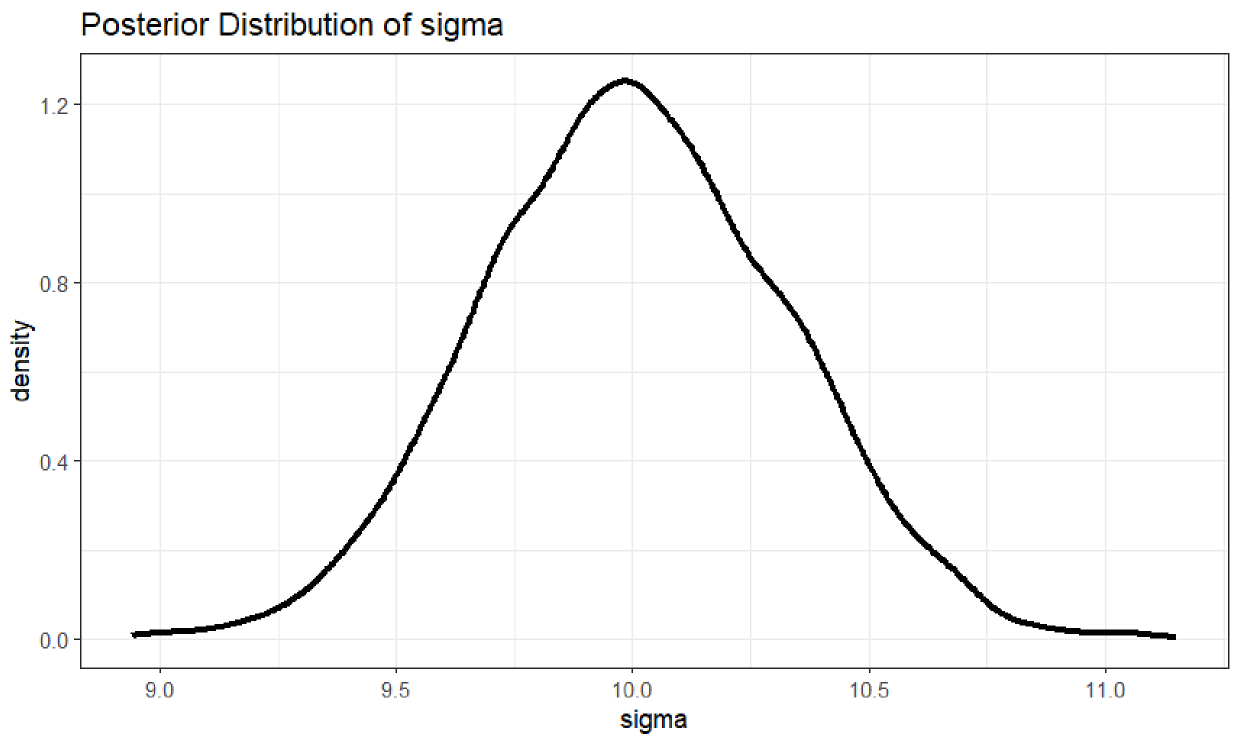
Part 3: Hamiltonian Monte Carlo sampler

For below graphs:

```
> true_mu <- 800
> true_var <- 100 #sigma^2
> y <- rnorm(500,mean=true_mu,sd=sqrt(true_var))
> hist(y)
> #Gradient functions
> gradient <- function(mu,sigma,y,n,m,s,a,b){
+  grad_mu <- (((n*mu)-sum(y))/(sigma^2))+((mu-m)/(s^2))
+  grad_sigma <- (n/sigma)-(sum((y-mu)^2)/(sigma^3))+((sigma-a)/(b^2))
+  return(c(grad_mu,grad_sigma))
+ }
> #Potential energy function
> V <- function(mu,sigma,y,n,m,s,a,b){
+  nlpd <-
-(sum(dnorm(y,mu,sigma,log=T))+dnorm(mu,m,s,log=T)+dnorm(sigma,a,b,log=T))
+  nlpd
+ }
> HMC <- function(y,n,m,s,a,b,step,L,initial_q,nsamp,nburn){
+  mu_chain <- rep(NA,nsamp)
+  sigma_chain <- rep(NA,nsamp)
+  reject <- 0
+  #Initialization of Markov chain
+  mu_chain[1] <- initial_q[1]
+  sigma_chain[1] <- initial_q[2]
+  #Evolution of Markov chain
+  i <- 1
+  while(i < nsamp){
+    q <- c(mu_chain[i],sigma_chain[i]) # Current position of the particle
+    p <- rnorm(length(q),0,1) # Generate random momentum at the current
position
+    current_q <- q
+    current_p <- p
+    current_V = V(current_q[1],current_q[2],y,n,m,s,a,b) # Current potential
energy
+    current_T = sum(current_p^2)/2 # Current kinetic energy
+    # Take L leapfrog steps
+    for(l in 1:L){
+      # Change in momentum in 'step/2' time
+      p <- p-((step/2)*gradient(q[1],q[2],y,n,m,s,a,b))
+      # Change in position in 'step' time
+      q <- q + step*p
+      # Change in momentum in 'step/2' time
+      p <- p-((step/2)*gradient(q[1],q[2],y,n,m,s,a,b))
+    }
+    proposed_q <- q
+    proposed_p <- p
+    proposed_V = V(proposed_q[1],proposed_q[2],y,n,m,s,a,b) # Proposed
potential energy
+    proposed_T = sum(proposed_p^2)/2 # Proposed kinetic energy
+    accept.prob <- min(1,exp(current_V+current_T-proposed_V-proposed_T))
+    # Accept/reject the proposed position q
+    if(accept.prob>runif(1,0,1)){
+      mu_chain[i+1] <- proposed_q[1]
+      sigma_chain[i+1] <- proposed_q[2]
+      i <- i+1
+    }else{
+      reject <- reject+1
+    }
+  }
+  posteriors <- data.frame(mu_chain,sigma_chain)[-(1:nburn),]
+  posteriors$sample_id <- 1:nrow(posteriors)
+  posteriors
+ }
> df.posterior <- HMC(y=y,n=length(y), # data
+                     m=1000,s=20,a=10,b=2, # priors
+                     step=0.02, # step-size
+                     L=12, # no. of leapfrog steps
+                     initial_q=c(1000,11), # Chain initialization
+                     nsamp=6000, # total number of samples
+                     nburn=2000) # number of burn-in samples
> ggplot(df.posterior, aes(x = mu_chain)) +
+  geom_density(size = 1.2) +
+  theme_bw() + xlab("mu") +
+  theme(legend.title = element_blank(),
+        legend.position = "top") +
+  ggtitle("Posterior Distribution of mu ")
> ggplot(df.posterior, aes(x = sigma_chain)) +
+  geom_density(size = 1.2) +
+  theme_bw() + xlab("sigma") +
+  theme(legend.title = element_blank(),
+        legend.position = "top") +
+  ggtitle("Posterior Distribution of sigma")
```

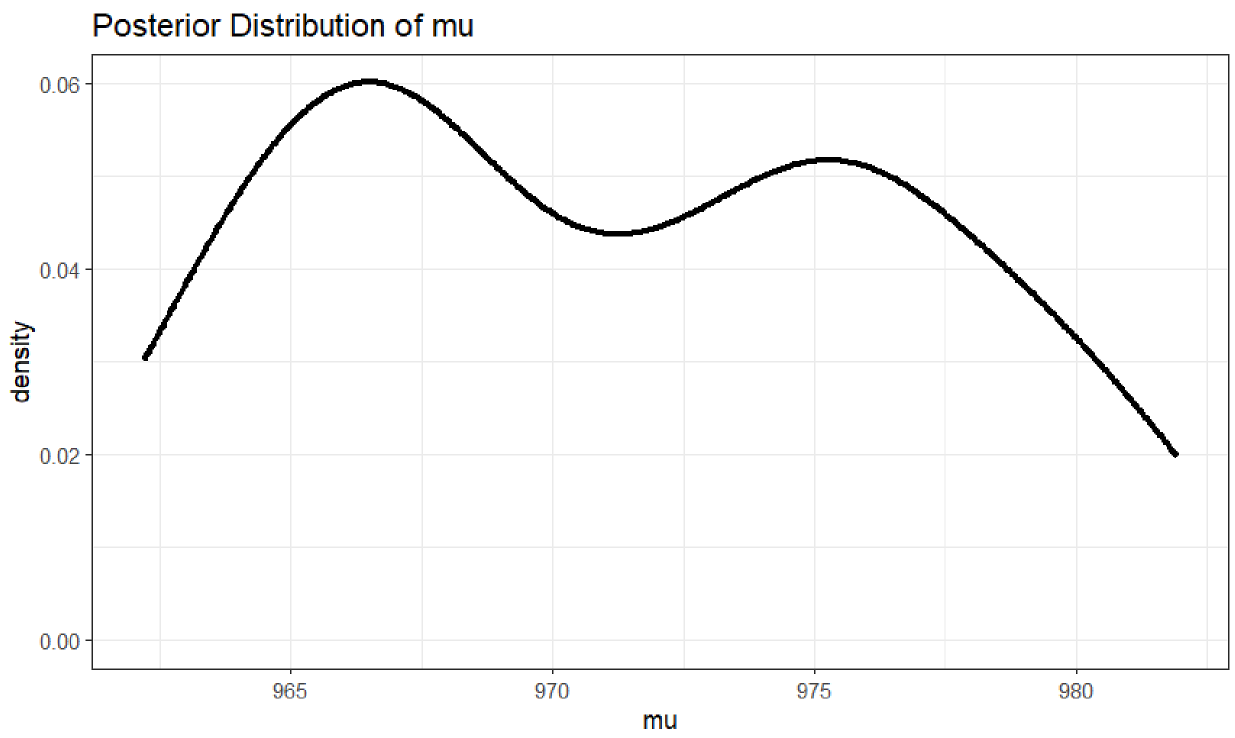
3.1:

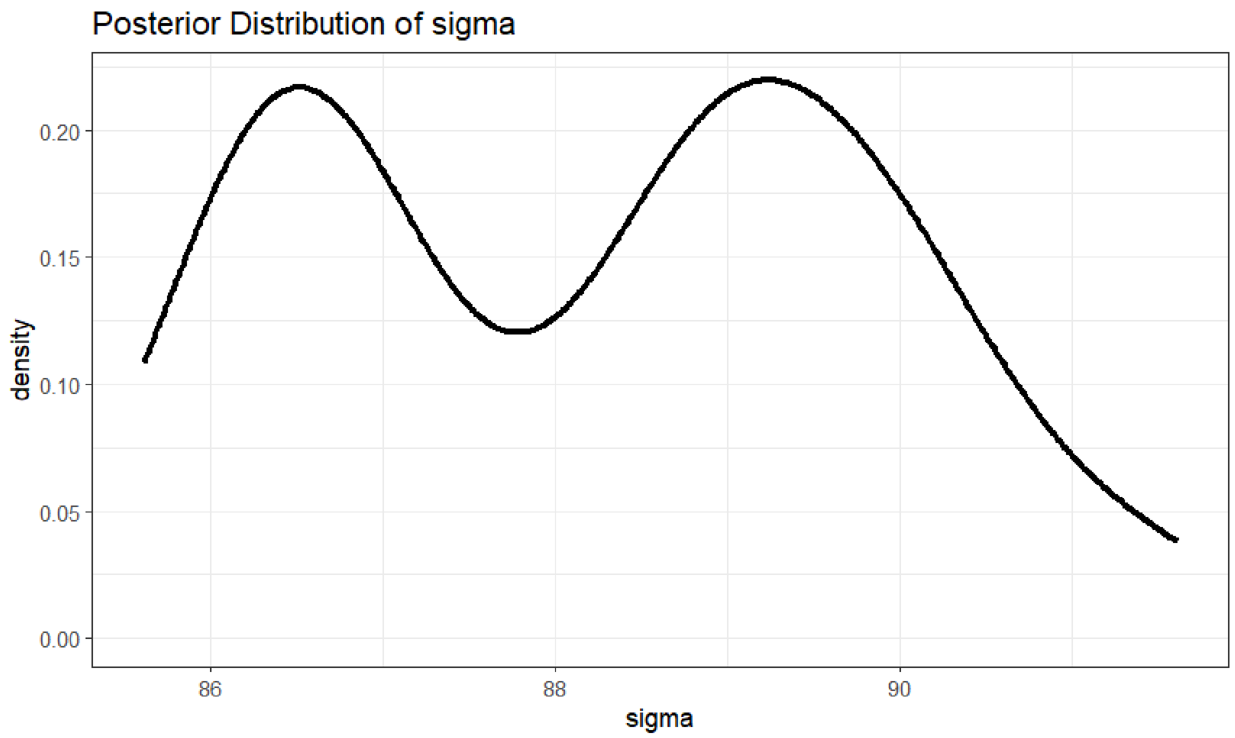

Posterior Distribution of mu
density
0.0
0.2
0.4
0.6
0.8
798
799
800
mu


Posterior Distribution of sigma
density
0.0
0.4
0.8
1.2
9.0
9.5
10.0
10.5
11.0
sigma

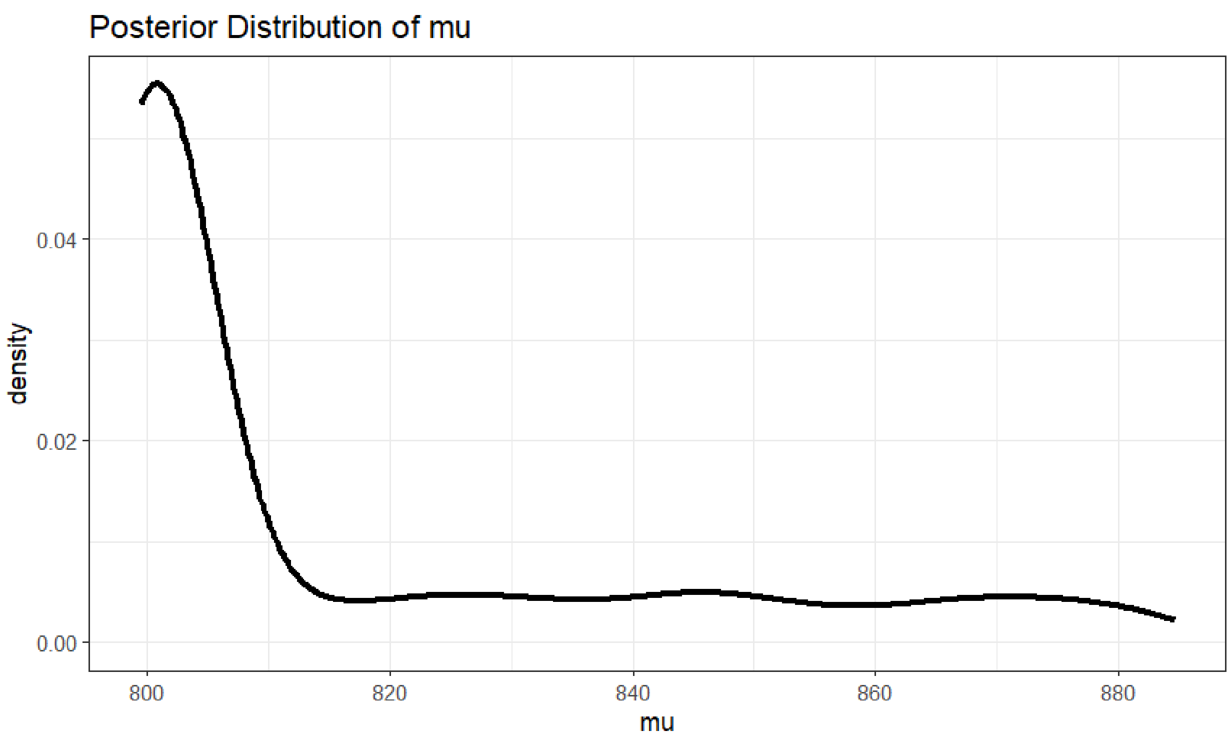
3.2:

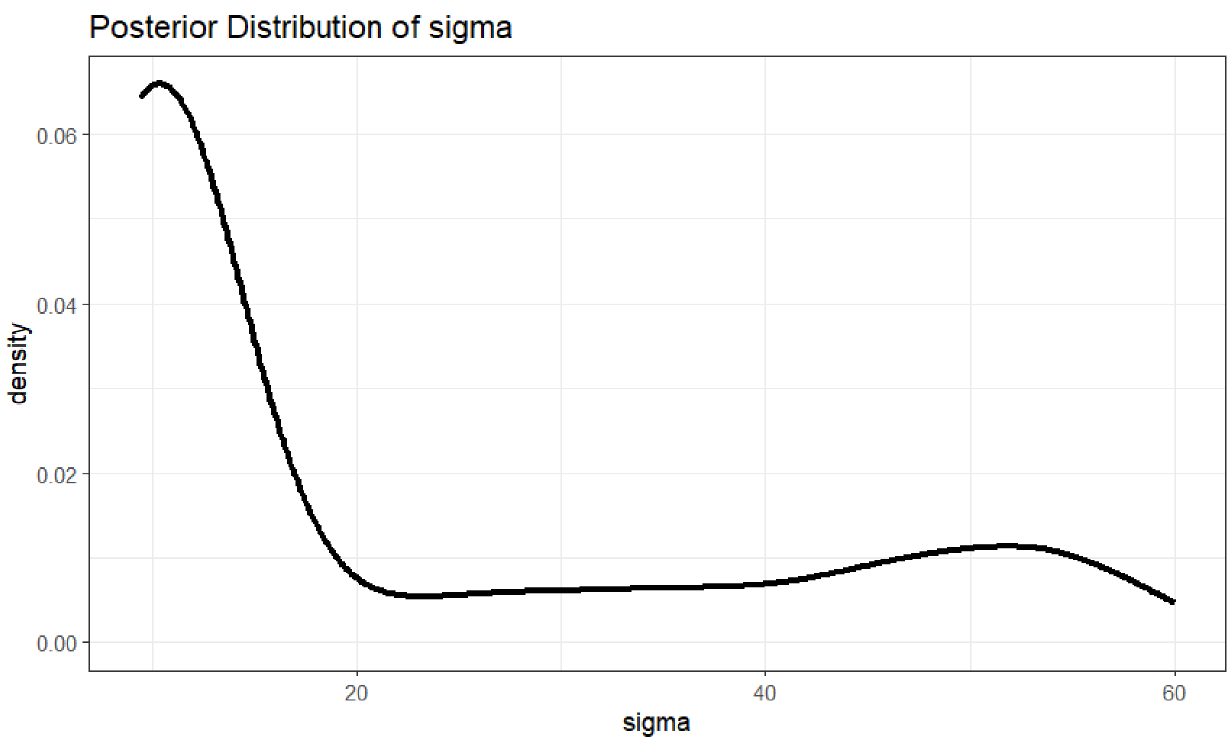
nsamp=100,nburn=100/3 = 33



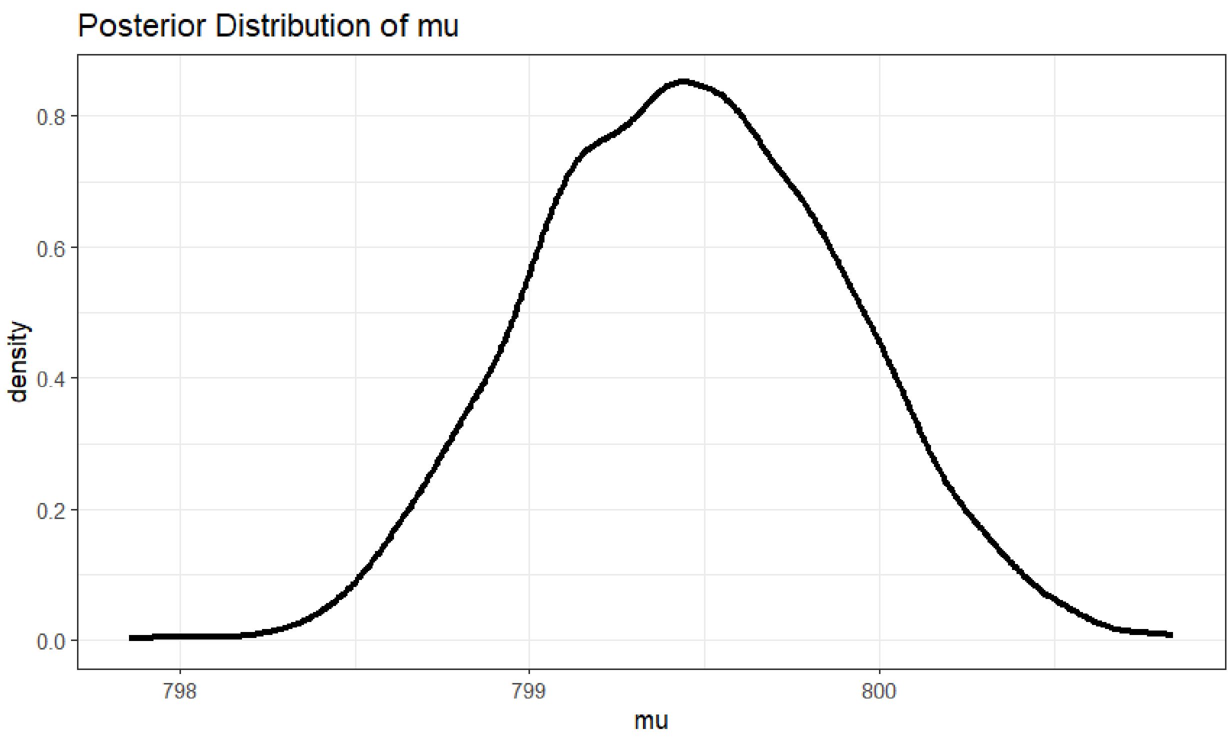


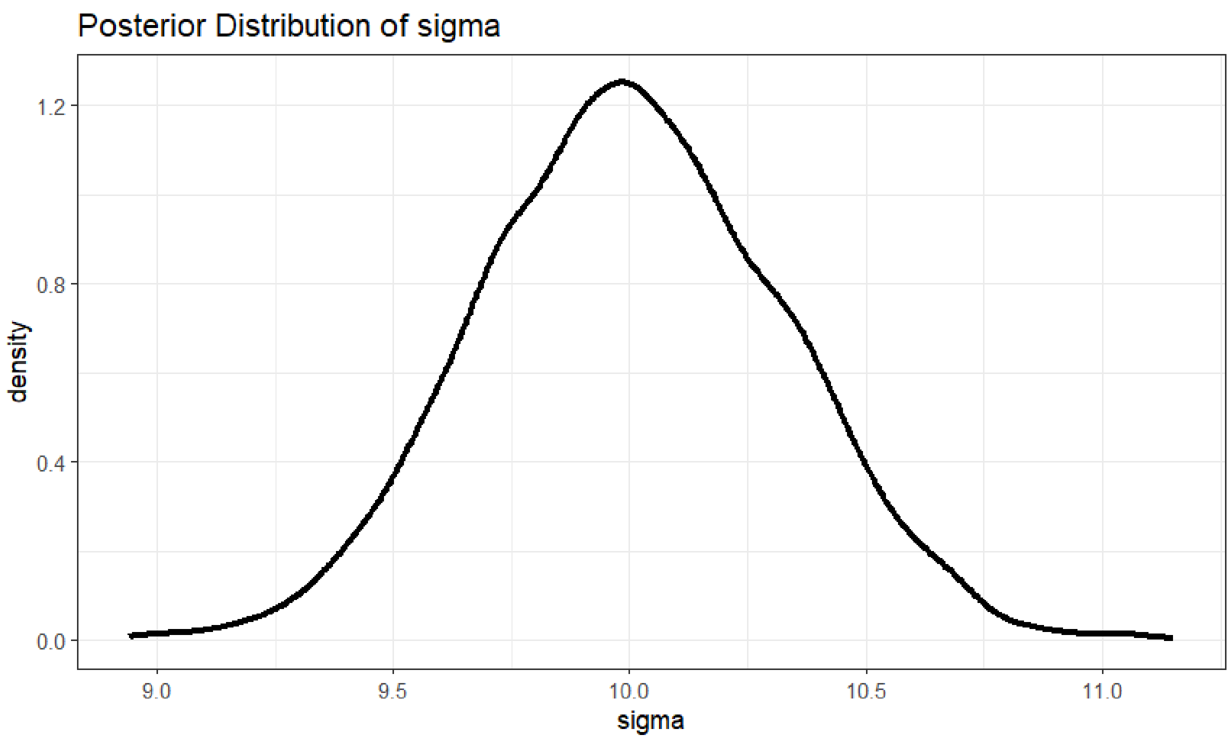
nsamp=1000,nburn=1000/3 = 333





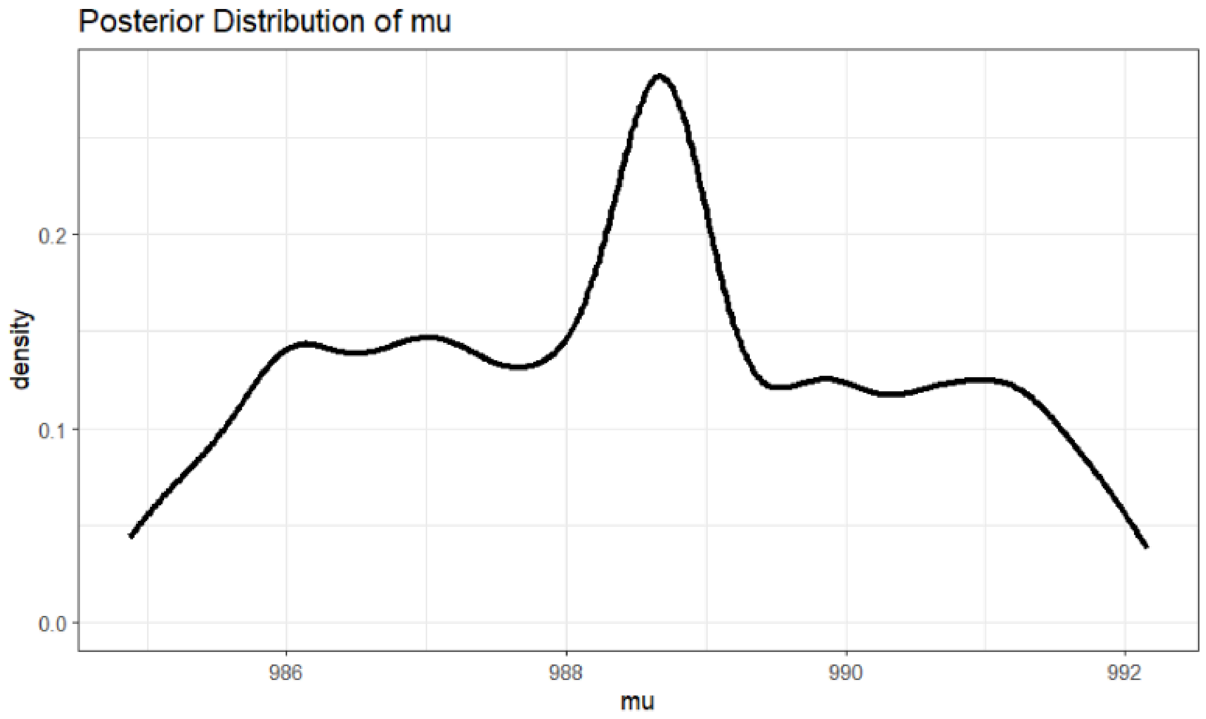
nsamp=6000,nburn=6000/3=2000

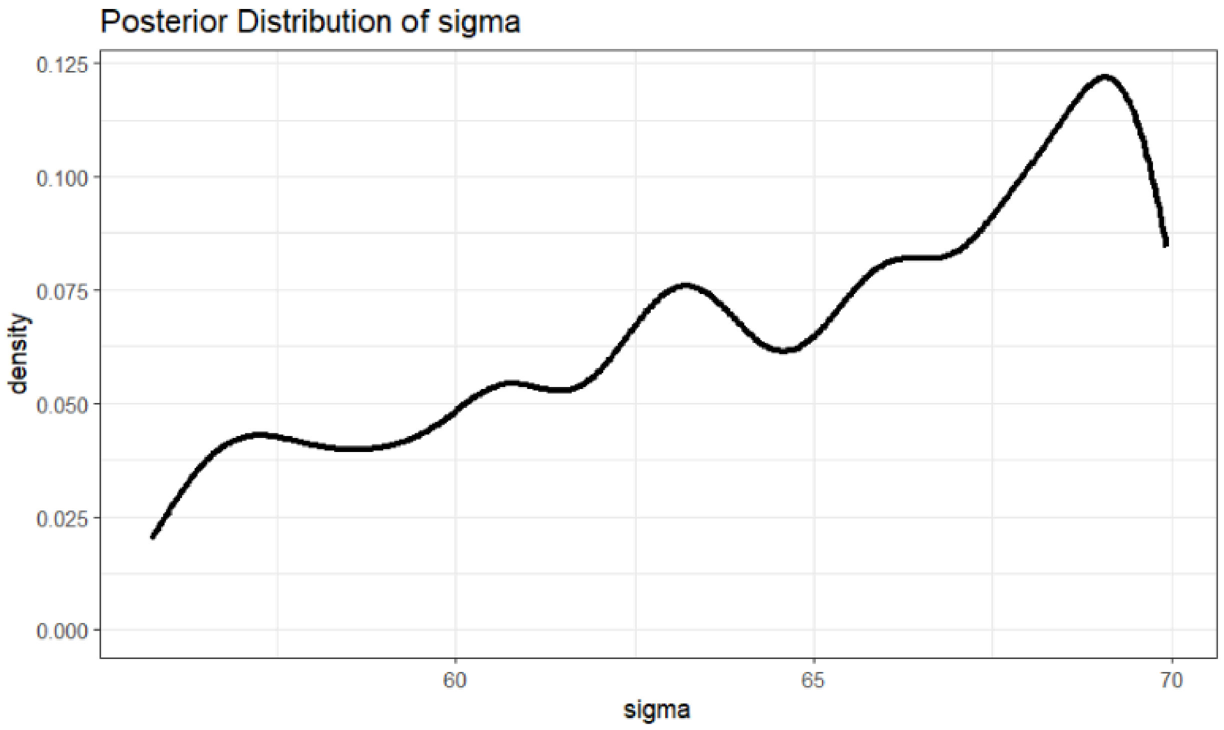




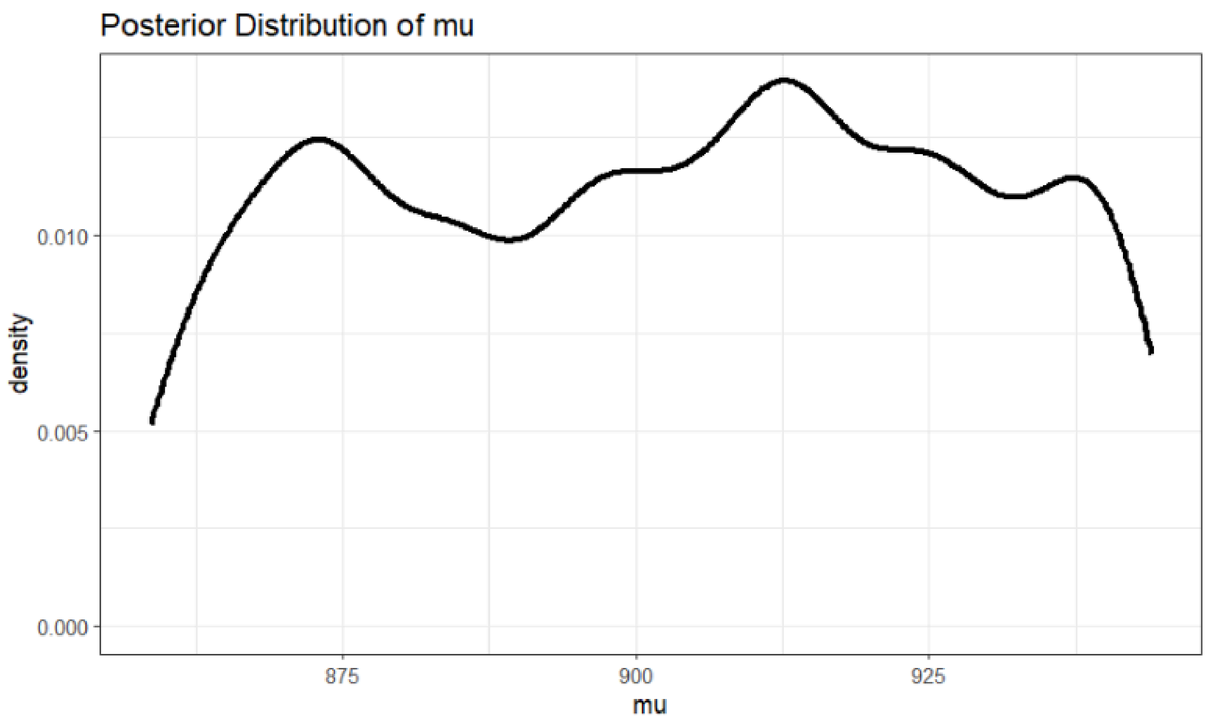
3.3:

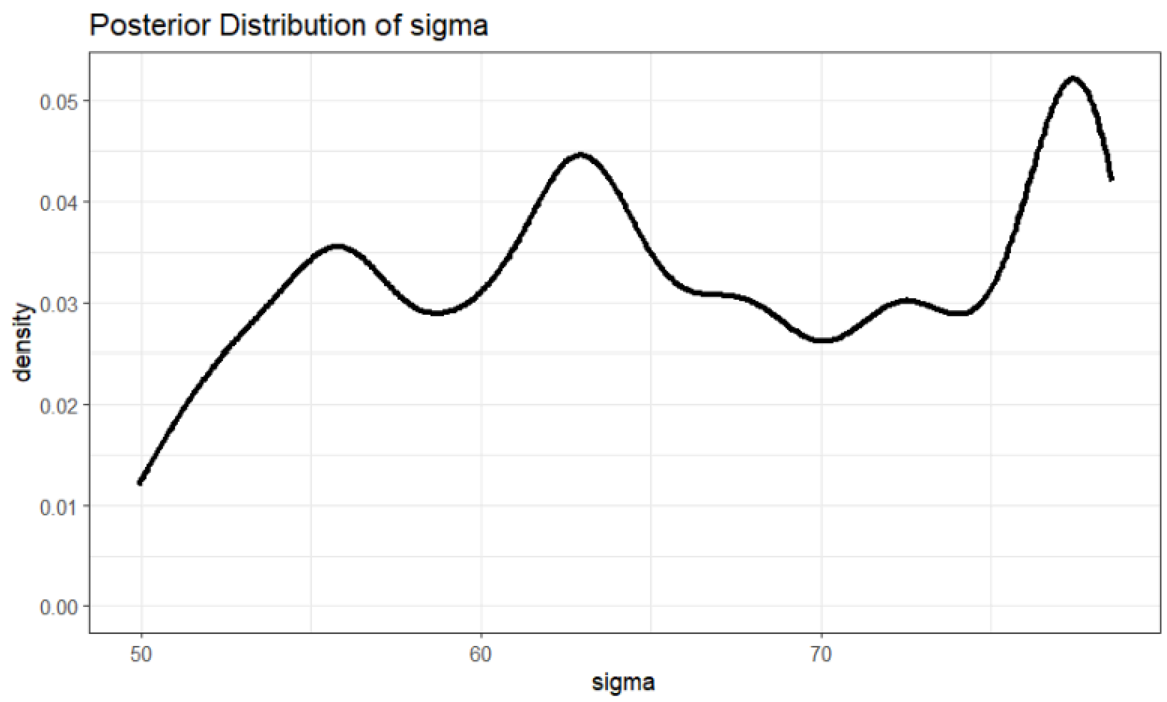
step-size=0.01



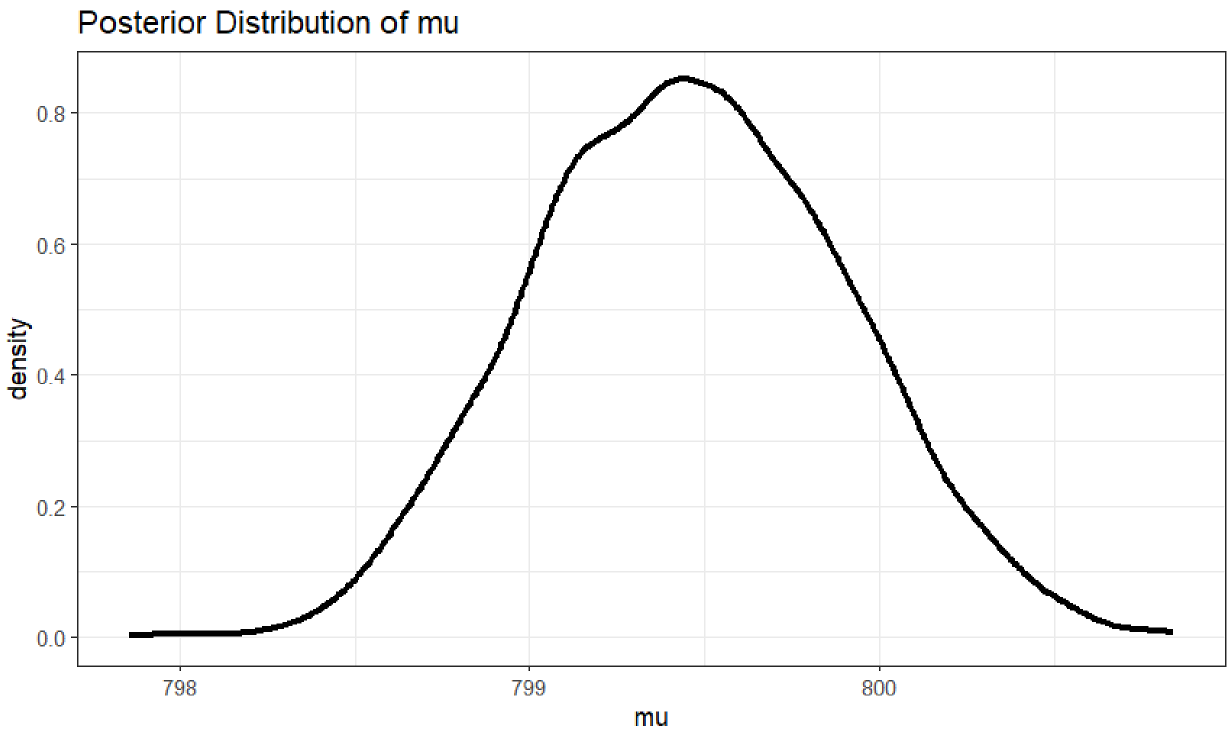


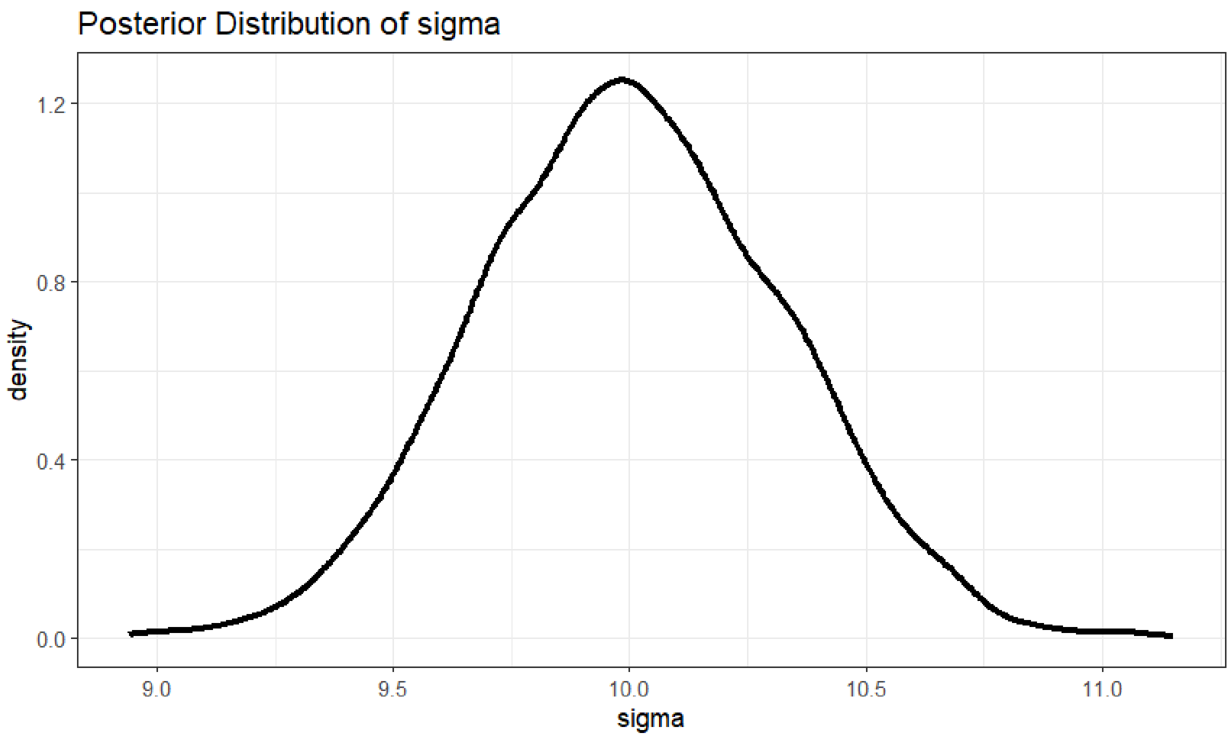
step-size=0.005





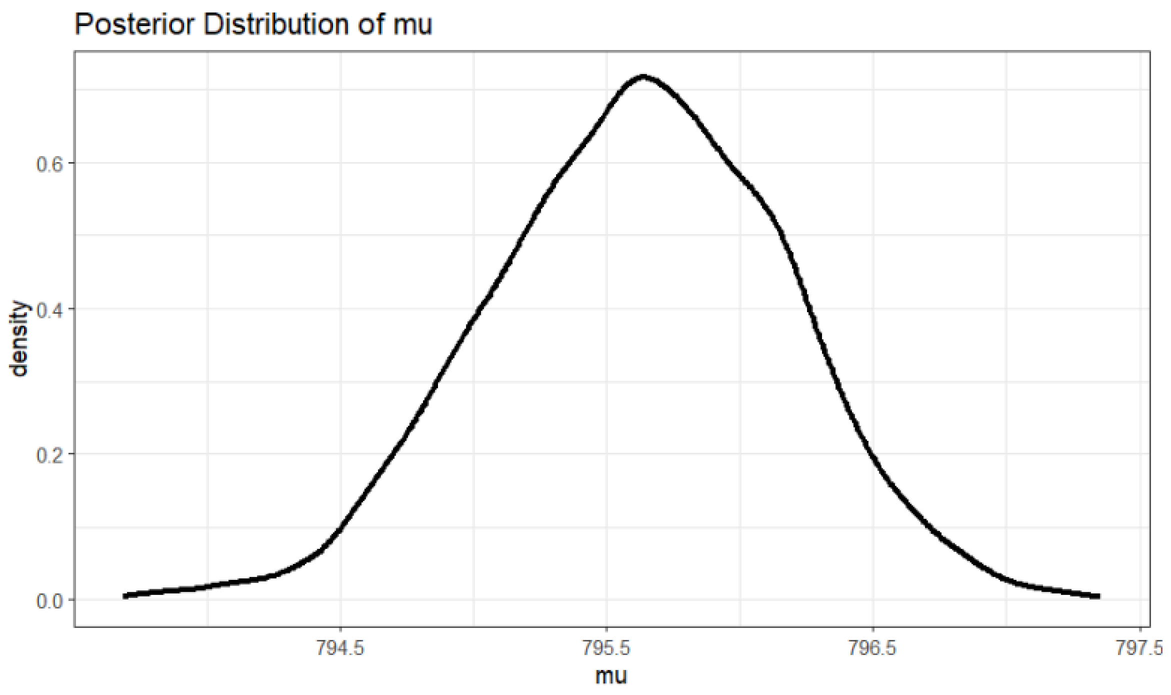
step-size=0.02

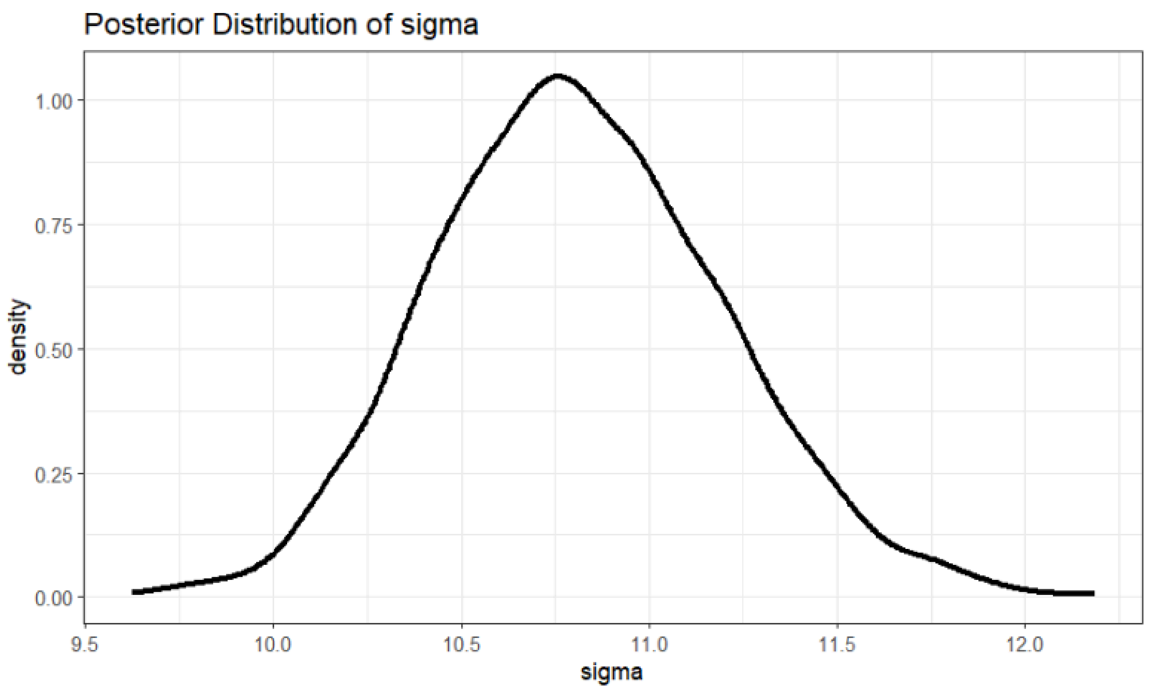




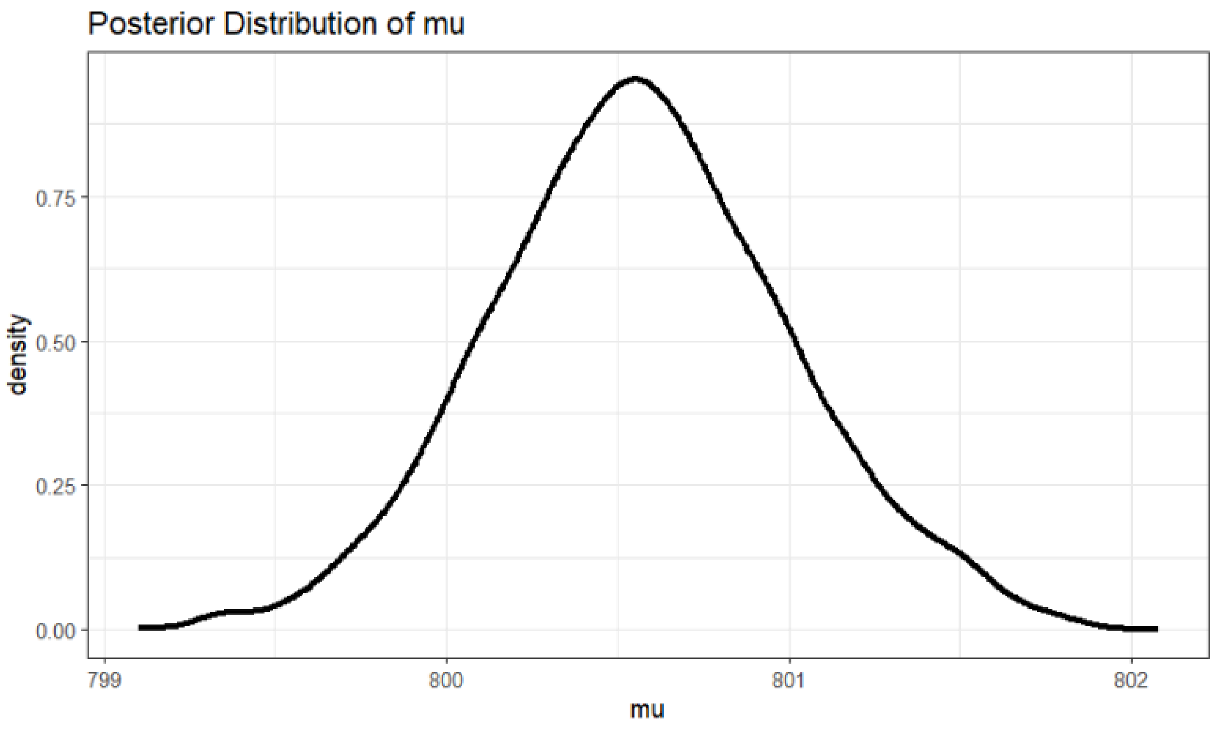
2.4:
We can see that the above graphs the chain seems to move to slow and gets stuck around the mean.

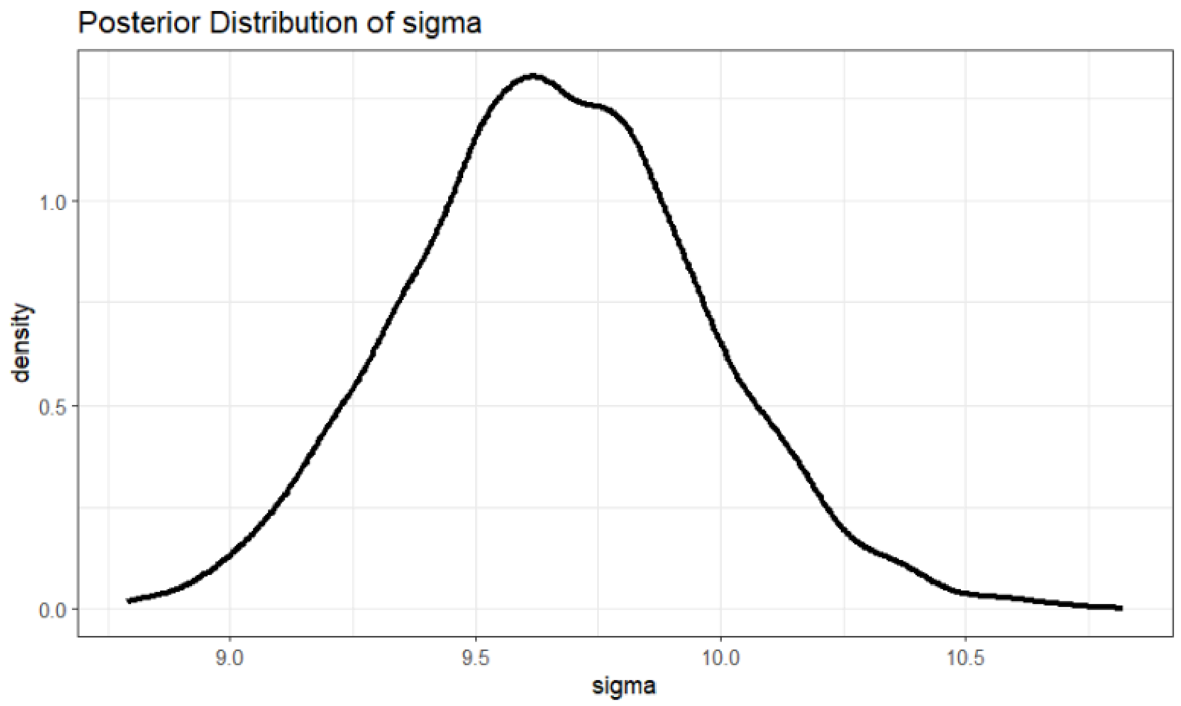
2.5:
μ ~ Normal(m = 400,s = 5)



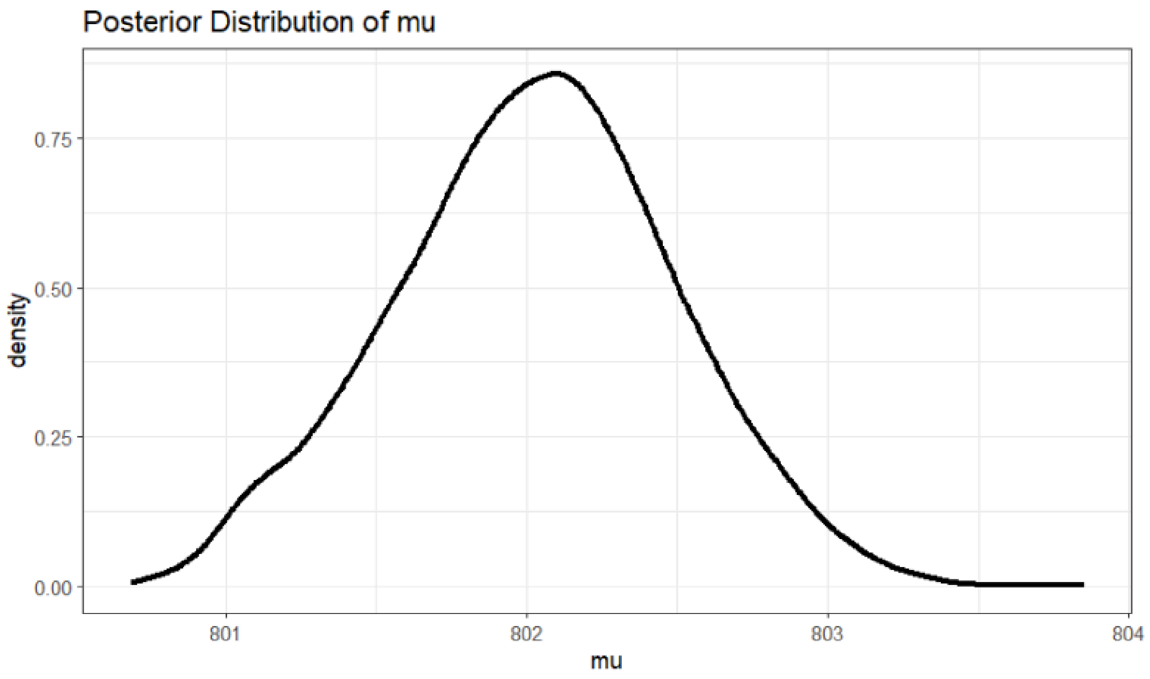


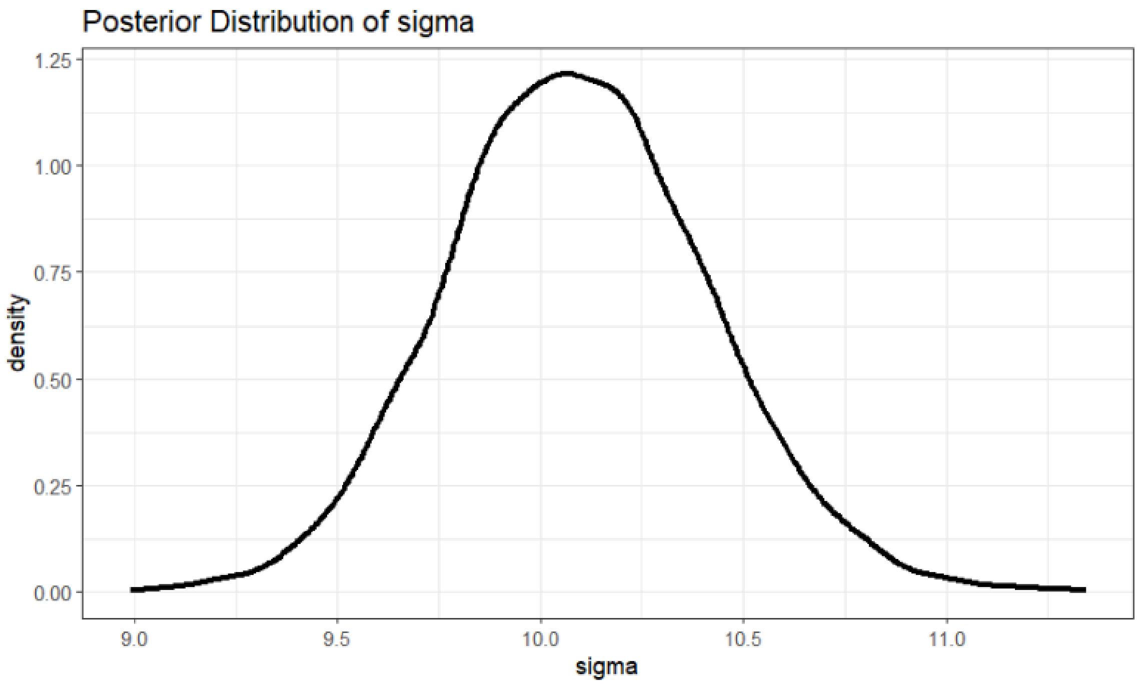
μ ~ Normal(m = 400,s = 20)





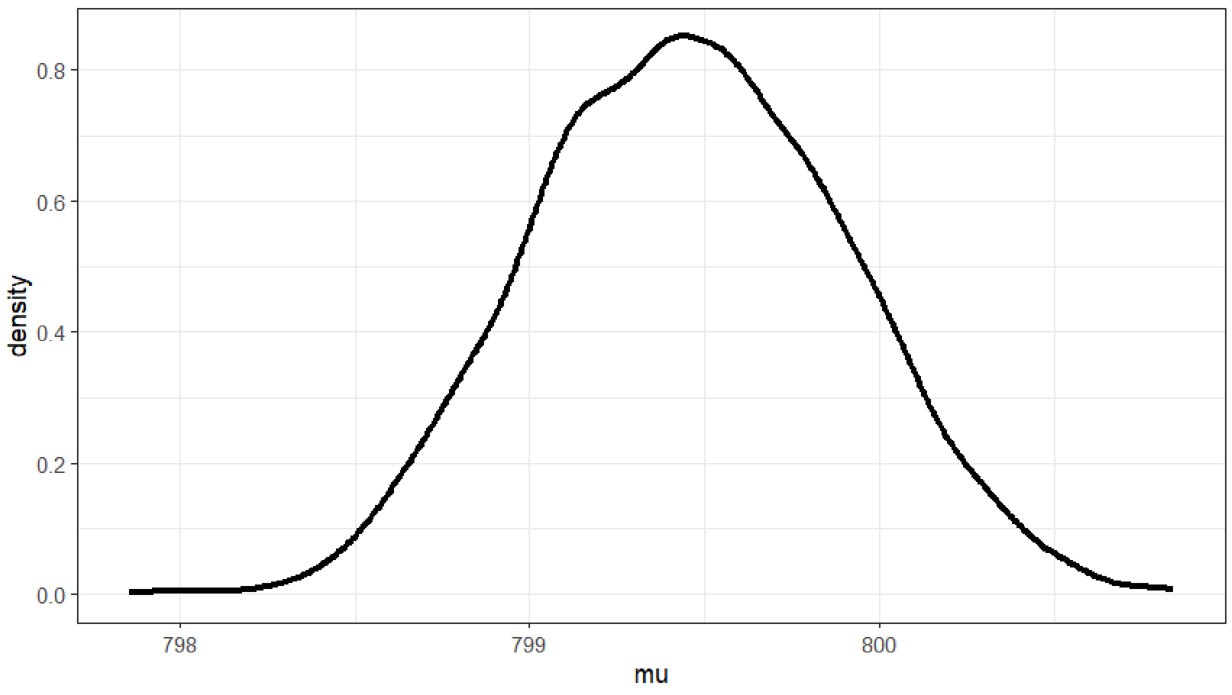
μ ~ Normal(m = 1000,s = 5)



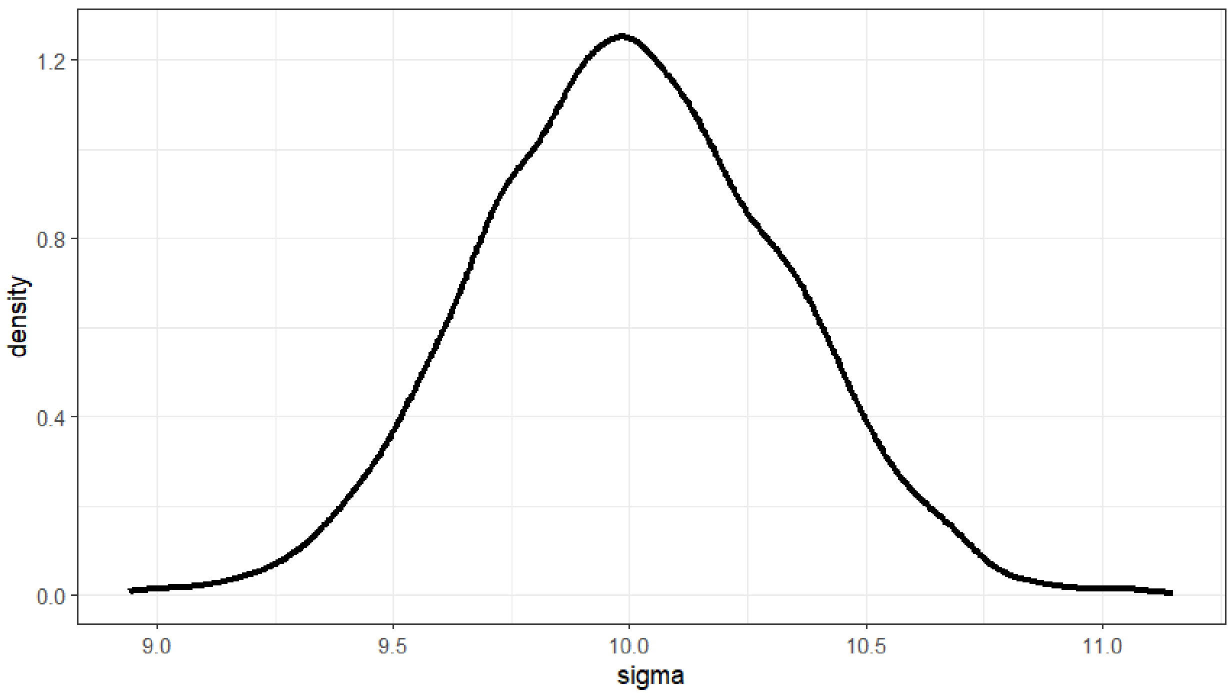


μ ~ Normal(m = 1000,s = 20)

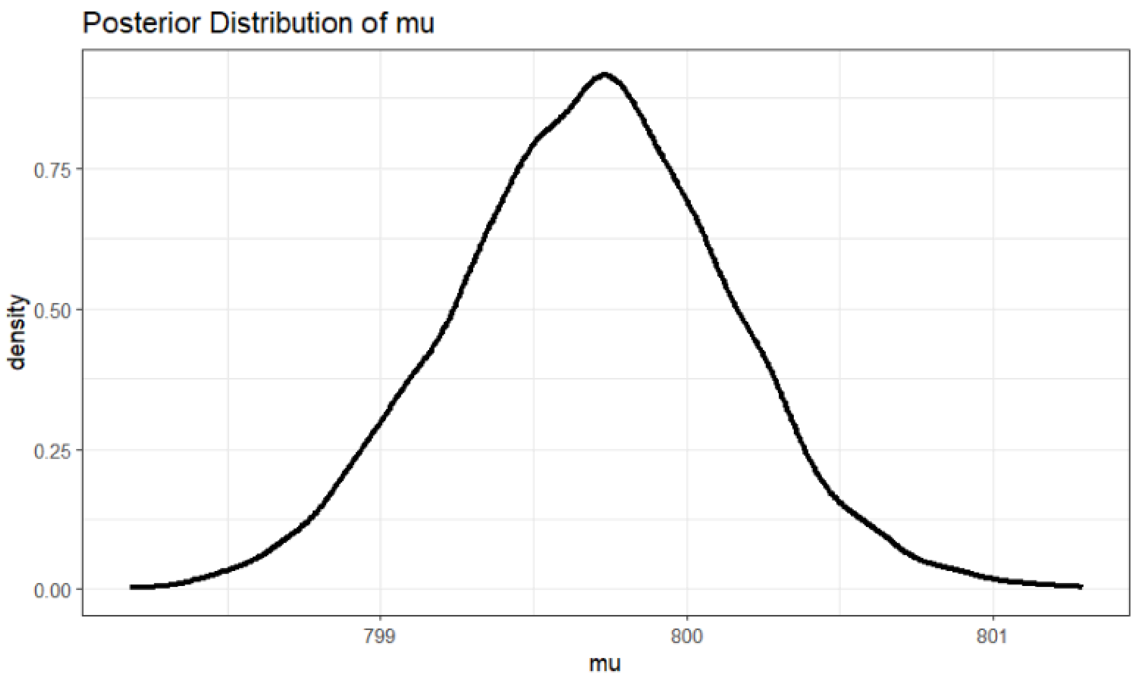
Posterior Distribution of mu

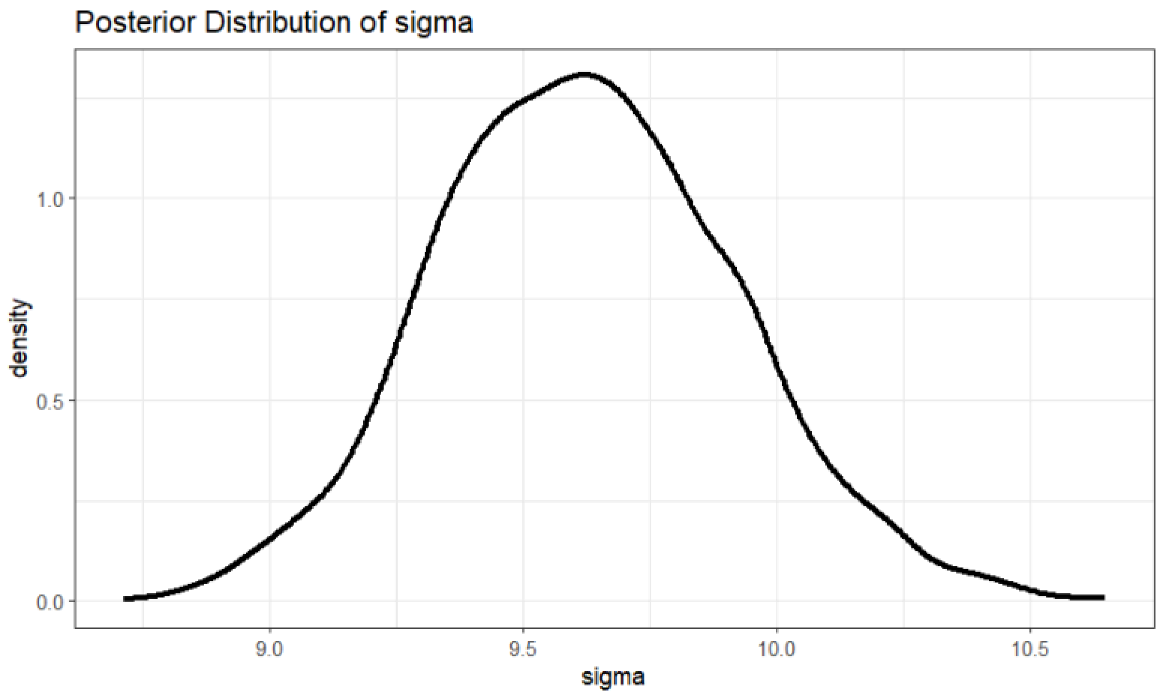


Posterior Distribution of sigma



μ ~ Normal(m = 1000,s = 100)


Posterior Distribution of mu
density
mu


Posterior Distribution of sigma
density
sigma