

Assignment 3

Aryan Thada : 210205

2024-06-21

{Q1}

```
library(ggplot2)
library(truncnorm)
library(dplyr)

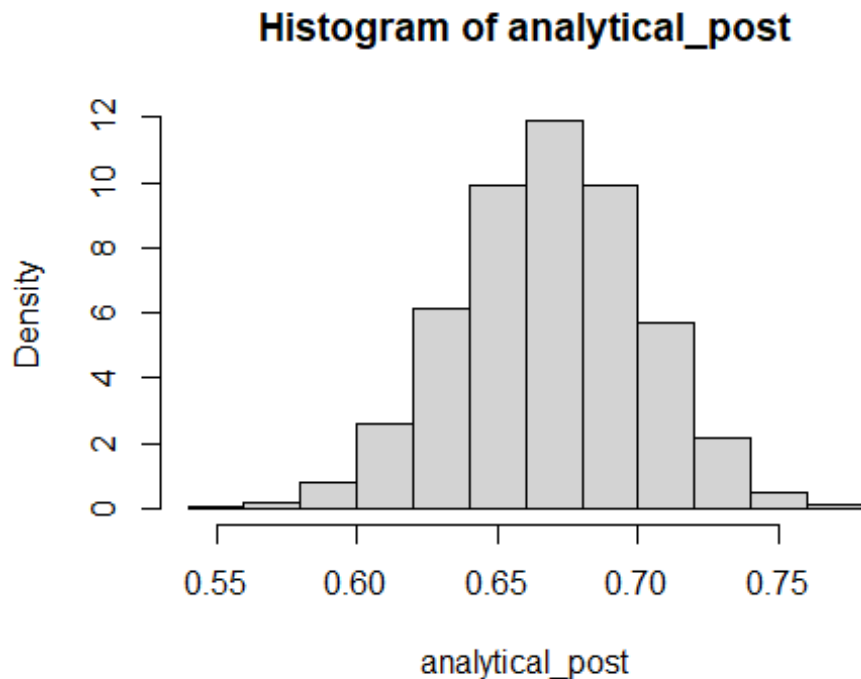
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(reshape2)

analytical_post <- rbeta(10000, 135, 67)
hist(analytical_post, freq = FALSE)
```



```

y <- c(10, 15, 15, 14, 14, 14, 13, 11, 12, 16)

theta_grid <- seq(0, 1, length = 10000)

df.posterior <- data.frame(matrix(ncol = 3, nrow = length(theta_grid)))
colnames(df.posterior) <- c("theta", "likelihood", "prior")

for (i in 1:length(theta_grid)) {
  likelihood <- prod(dbinom(y, size = 20, theta_grid[i]))
  prior <- dbeta(theta_grid[i], 1, 1)
  df.posterior[i, ] <- c(theta_grid[i], likelihood, prior)
}

df.posterior$ML <- rep(sum(df.posterior$likelihood * df.posterior$prior),
10000)

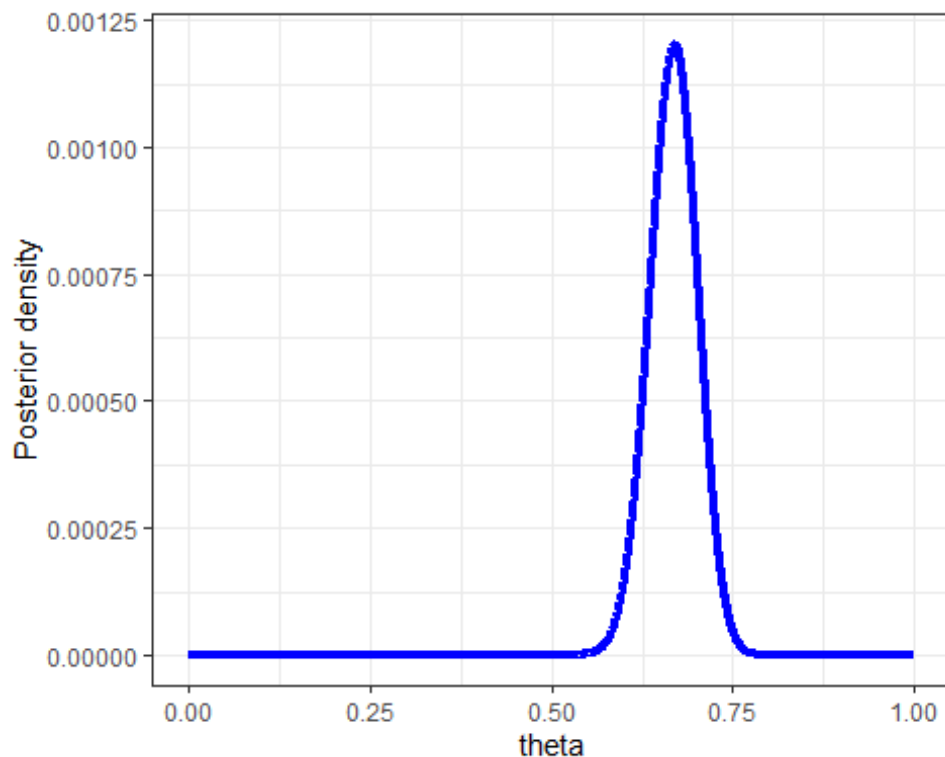
df.posterior <- df.posterior %>%
  mutate(posterior = likelihood * prior / ML)

ggplot(df.posterior, aes(x = theta, y = posterior)) +
  geom_line(aes(group = NA), size = 1.5, colour = "blue") +
  theme_bw() +

```

```
scale_x_continuous() +
ylab("Posterior density")
```

```
## Warning: 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.
```



```
df.estimate <- data.frame(matrix(ncol = 2, nrow = 40000))
colnames(df.estimate) <- c("theta_sample", "likelihood")

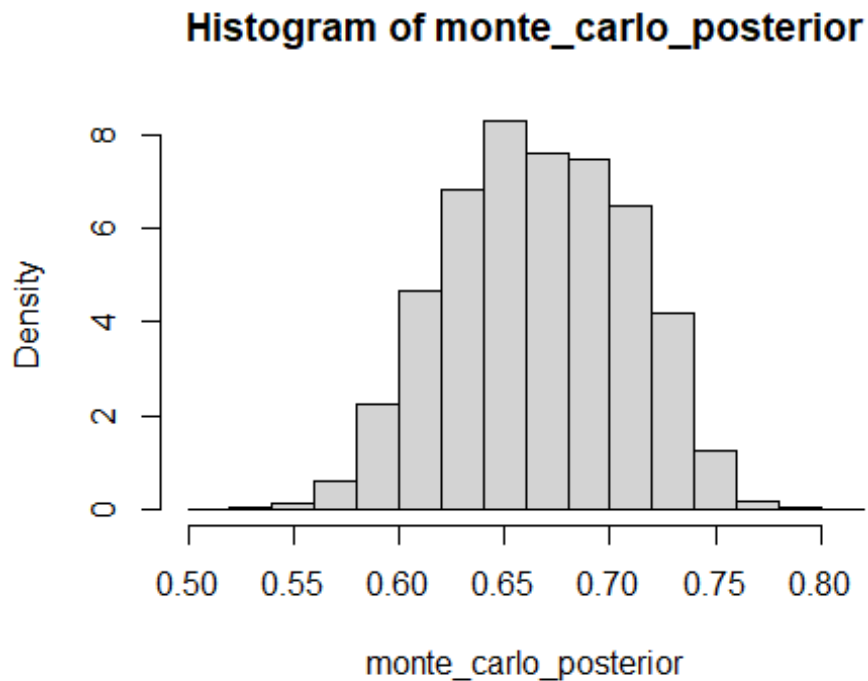
for (i in 1:40000) {
  theta_i <- rbeta(1, 1, 1) # independent sample from the prior
  likelihood <- prod(dbinom(y, 20, theta_i))
  df.estimate[i, ] <- c(theta_i, likelihood)
}

ML <- mean(df.estimate$likelihood)
ML

## [1] 1.386346e-10

monte_carlo_posterior <- sample(df.estimate$theta_sample,
  size = 5000,
  prob = df.estimate$likelihood / sum(df.estimate$likelihood))
```

```
)  
hist(monte_carlo_posterior, freq = FALSE)
```



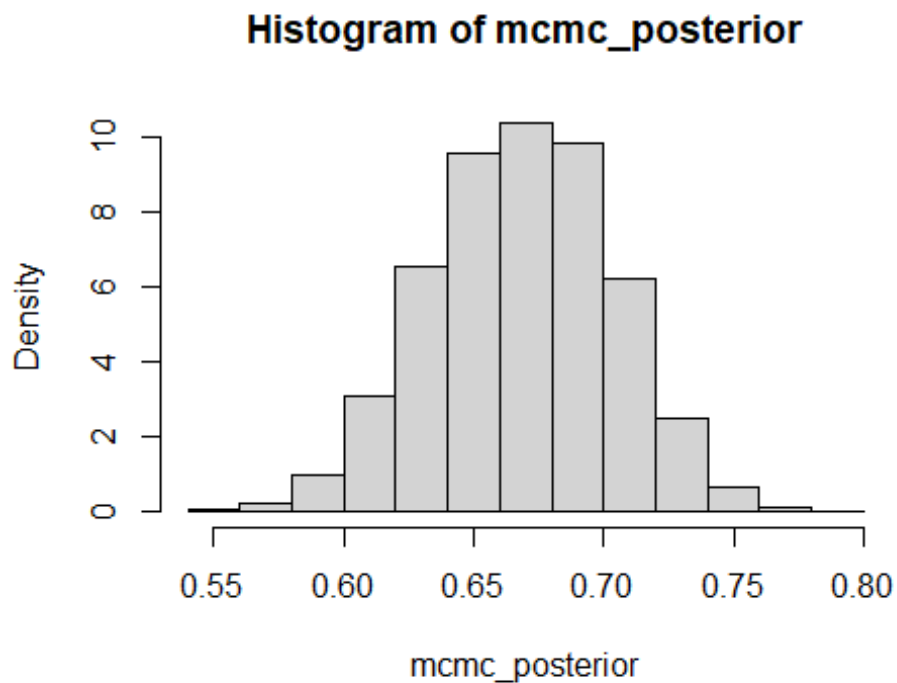
```
nsamp <- 20000  
theta_chain <- rep(NA, nsamp)  
  
theta_chain[1] <- rbeta(1, 1, 1)  
  
i <- 1  
step <- 0.08  
size = 20  
while (i < nsamp) {  
  proposal_theta <- rnorm(1, theta_chain[i], step)  
  if (proposal_theta > 0 & proposal_theta < 1) {  
    post_new <- prod(dbinom(y, size, proposal_theta)) * dbeta(proposal_theta,  
1, 1)  
    post_prev <- prod(dbinom(y, size, theta_chain[i])) *  
dbeta(theta_chain[i], 1, 1)  
  
    Hastings_ratio <- (post_new * dnorm(theta_chain[i], proposal_theta,  
step)) /  
    (post_prev * dnorm(proposal_theta, theta_chain[i], step))  
    p_str <- min(Hastings_ratio, 1)  
  }  
}
```

```

    if (p_str > runif(1, 0, 1)) {
      theta_chain[i + 1] <- proposal_theta
      i <- i + 1
    }
  }
}

burn_in <- 2000
theta_chain <- theta_chain[(burn_in + 1):nsamp]
mcmc_posterior <- theta_chain[1:5000] hist(mcmc_posterior, freq = FALSE)

```



```

# Analytical Posterior
analytical_posterior <- rbeta(5000, 135, 67)

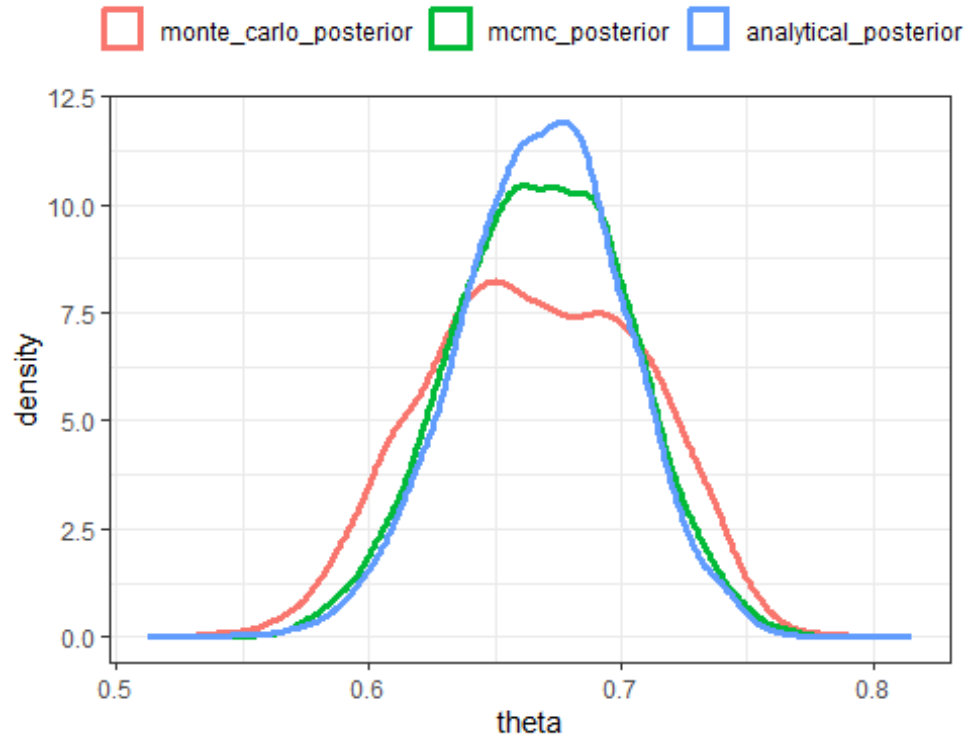
# Combine posteriors
posteriors <- data.frame(
  monte_carlo_posterior = monte_carlo_posterior,
  mcmc_posterior = mcmc_posterior,
  analytical_posterior = analytical_posterior
)

posteriors_melted <- melt(posteriors)

## No id variables; using all as measure variables

```

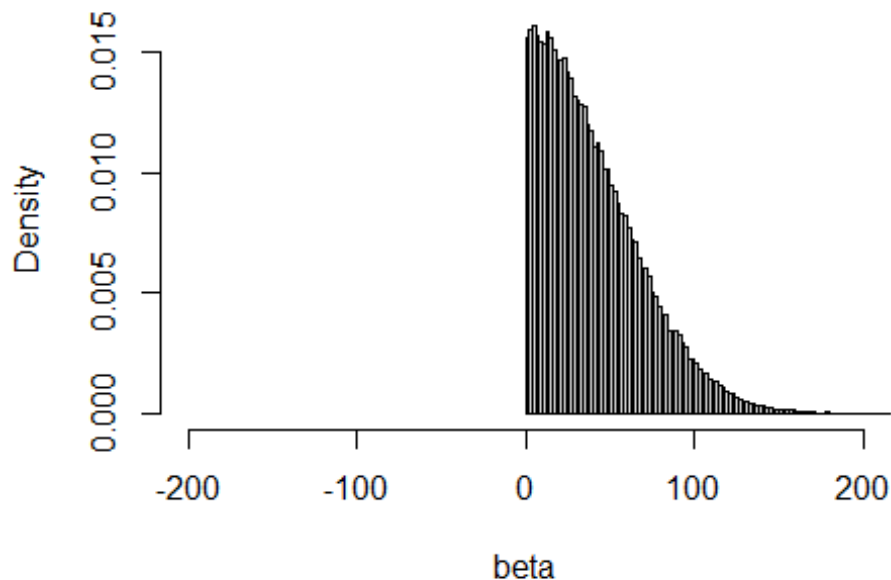
```
ggplot(posterior_melted, aes(x = value, colour = variable)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("theta") +
  theme(legend.title = element_blank(), legend.position = "top")
```



{Q2}

```
# Load required libraries
library(ggplot2)
library(truncnorm)
library(truncnorm)
# You can generate from a truncated normal distribution using rtruncnorm
beta <- rtruncnorm(100000, a=0, b=Inf, mean=0, sd=50)
hist(beta, xlim = c(-200, 200), probability = T, breaks = 100)
```

Histogram of beta



```
x <- 20
density_x <- dtruncnorm(x,a=0,b=Inf,mean=0,sd=50)

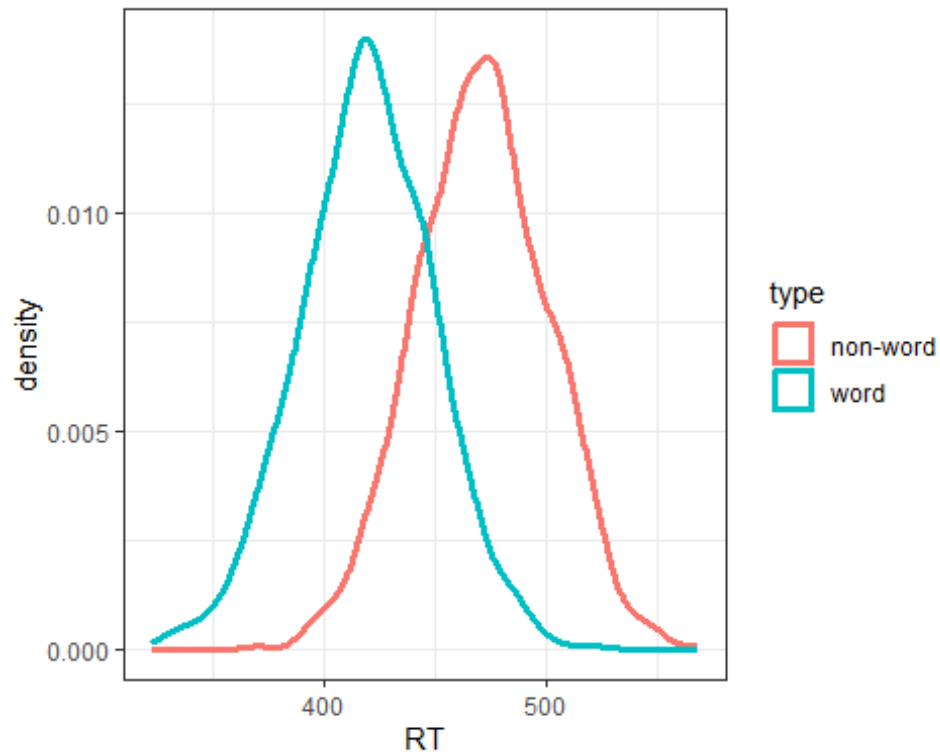
# Load the data
dat <- read.table(

  "https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main/notes/Data/w
ord-recognition-times.csv",
  sep = ",", header = TRUE
)[, -1]

# Inspect the first few rows of the data
head(dat)

##      type      RT
## 1    word 423.1019
## 2    word 429.9432
## 3 non-word 486.9959
## 4 non-word 451.4400
## 5 non-word 482.2657
## 6 non-word 470.8003

# Plot the density of recognition times for words and non-words
ggplot(dat, aes(x = RT, color = type)) +
  geom_density(linewidth = 1.2) +
  theme_bw()
```



```
library(MCMCpack)

## Loading required package: coda
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##   select
##
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2024 Andrew D. Martin, Kevin M. Quinn, and Jong Hee
## ## Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##
library(truncnorm)
```



```

likelihood <- function(alpha, beta, sigma, data) {
  mu <- alpha + beta * (data$type == "non-word")
  sum(dnorm(data$RT, mean = mu, sd = sigma, log = TRUE))
}

dtruncnorm_log <- function(x, a, b, mean, sd) {
  dnorm(x, mean = mean, sd = sd, log = TRUE) - log(pnorm(b, mean = mean, sd =
sd) - pnorm(a, mean = mean, sd = sd))
}

prior <- function(alpha, beta) {
  dnorm(alpha, 400, 50, log = TRUE) + dtruncnorm_log(beta, a = 0, b = Inf,
mean = 0, sd = 50)
}

posterior <- function(alpha, beta, sigma, data) {
  likelihood(alpha, beta, sigma, data) + prior(alpha, beta)
}

set.seed(123)
nsamp <- 10000
alpha_chain <- numeric(nsamp)
beta_chain <- numeric(nsamp)
sigma <- 30

alpha_chain[1] <- 400
beta_chain[1] <- 0

for (i in 2:nsamp)
{
  alpha_proposal <- rnorm(1, alpha_chain[i - 1], 1)
  beta_proposal <- rtruncnorm(1, a = 0, b = Inf, mean = beta_chain[i - 1], sd
= 1)

  post_current <- posterior(alpha_chain[i - 1], beta_chain[i - 1], sigma,
dat)
  post_proposal <- posterior(alpha_proposal, beta_proposal, sigma, dat)

  accept_prob <- exp(post_proposal - post_current)

```

```

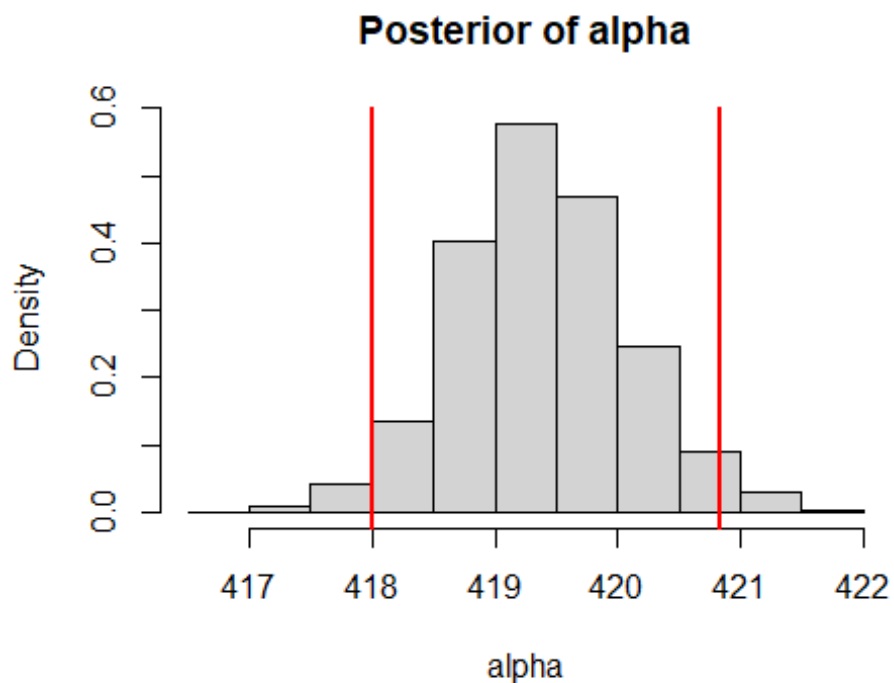
if (runif(1) < accept_prob)
{
  alpha_chain[i] <- alpha_proposal
  beta_chain[i] <- beta_proposal
}
else
{
  alpha_chain[i] <- alpha_chain[i - 1]
  beta_chain[i] <- beta_chain[i - 1]
}
}

burn_in <- 2000
alpha_chain <- alpha_chain[(burn_in + 1):nsamp]
beta_chain <- beta_chain[(burn_in + 1):nsamp]

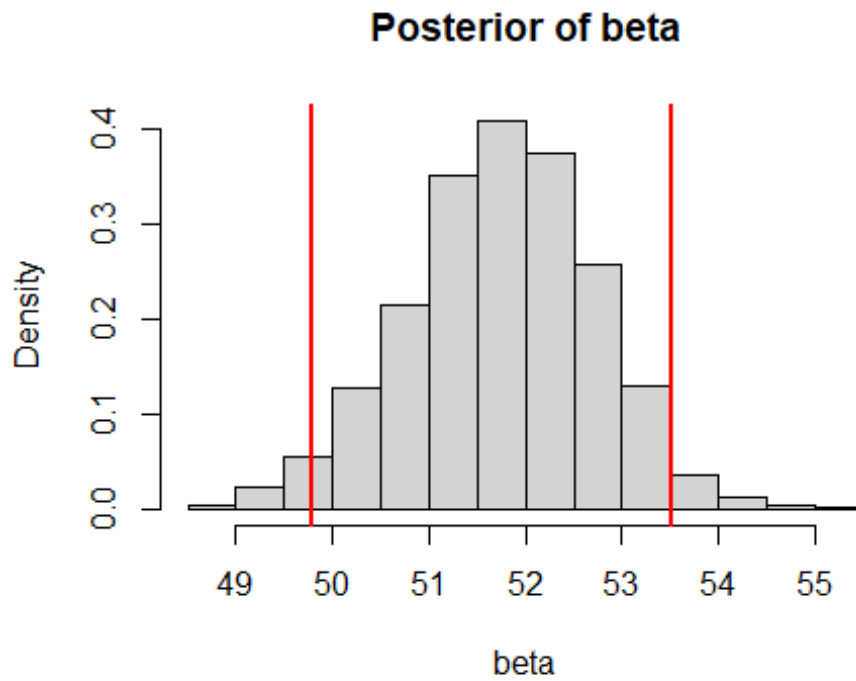
alpha_cred_int <- quantile(alpha_chain, probs = c(0.025, 0.975))
beta_cred_int <- quantile(beta_chain, probs = c(0.025, 0.975))

hist(alpha_chain, probability = TRUE, main = "Posterior of alpha", xlab =
"alpha")
abline(v = alpha_cred_int, col = "red", lwd = 2)

```



```
hist(beta_chain, probability = TRUE, main = "Posterior of beta", xlab =
"beta")
abline(v = beta_cred_int, col = "red", lwd = 2)
```



```
95% credible interval for alpha: 418.0035 420.834
95% credible interval for beta: 49.78226 53.50979

cat("95% credible interval for alpha:", alpha_cred_int, "\n")
## 95% credible interval for alpha: 418.0035 420.834
cat("95% credible interval for beta:", beta_cred_int, "\n")
## 95% credible interval for beta: 49.78226 53.50979

{Q3}

cat("\014");
```

```

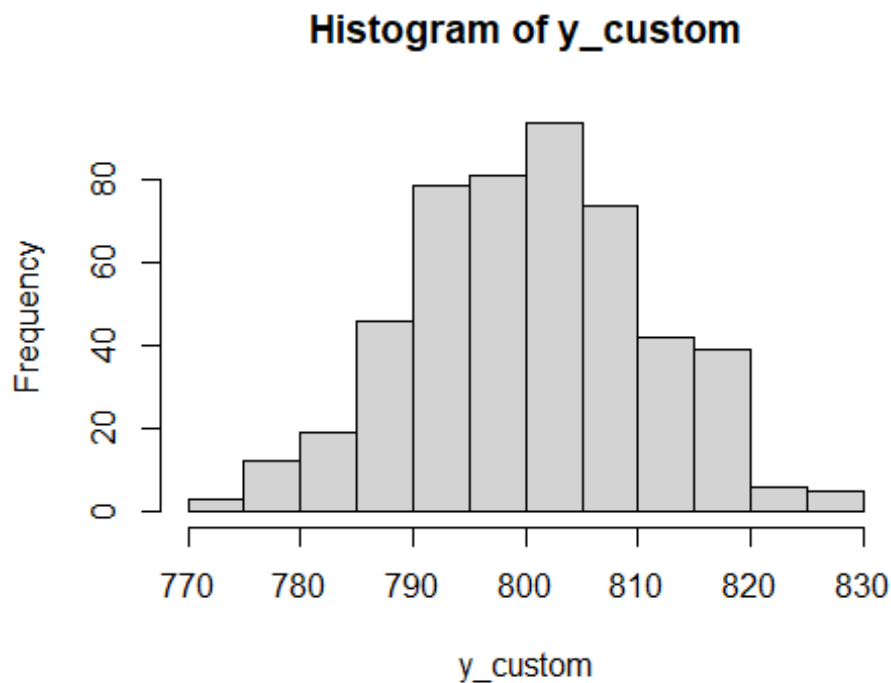
library(ggplot2)
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine

custom_mu <- 800
custom_var <- 100 #sigma^2
y_custom <- rnorm(500,mean=custom_mu,sd=sqrt(custom_var))
hist(y_custom)

```



```

#Gradient functions
gradient_custom <- 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_custom <- 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
}

```

```

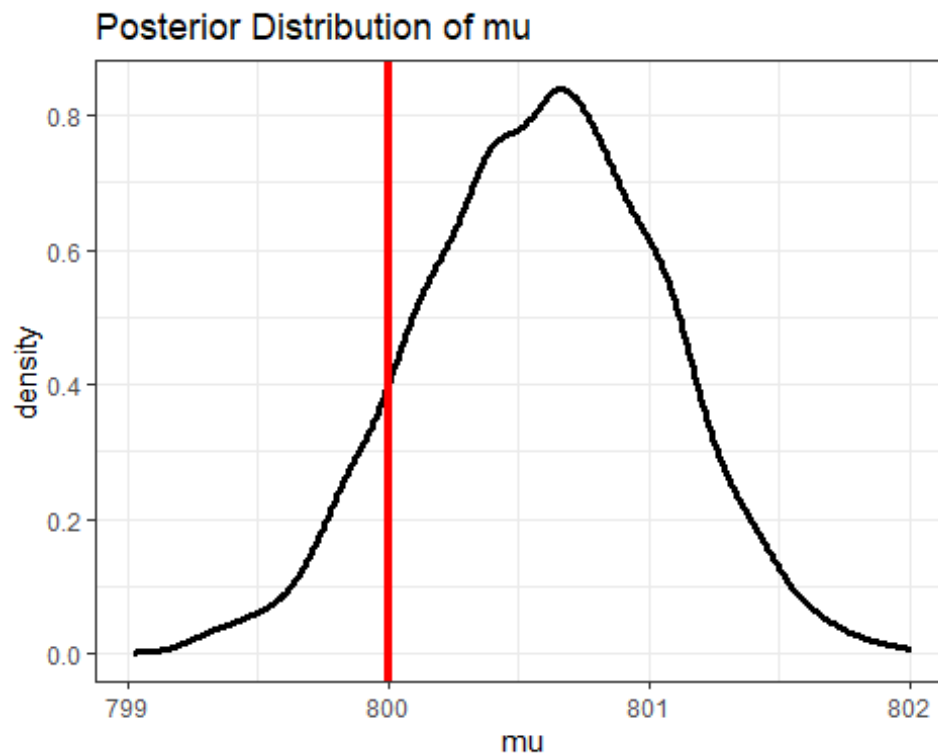
#HMC sampler
HMC_custom <- function(y,n,m,s,a,b,step,L,initial_q,nsamp,nburn){
  mu_chain <- rep(NA,nsamp)
  sigma_chain <- rep(NA,nsamp)
  reject <- 0
  #Initialization of Markov chain
  mu_chain[1] <- initial_q[1]
  sigma_chain[1] <- initial_q[2]
  #Evolution of Markov chain
  i <- 1
  while(i < nsamp){
    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_custom(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_custom(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_custom(q[1],q[2],y,n,m,s,a,b))
    }
    proposed_q <- q
    proposed_p <- p
    proposed_V = V_custom(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_custom <- HMC_custom(y=y_custom,n=length(y_custom), # 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_custom,aes(x=mu_chain))+
  geom_density(size=1.2)+
  theme_bw()+xlab("mu")+
  labs(title = "Posterior Distribution of mu") +
  theme(legend.title = element_blank(),
  legend.position = "top")+
  geom_vline(xintercept=800,size=1.5,color="red")

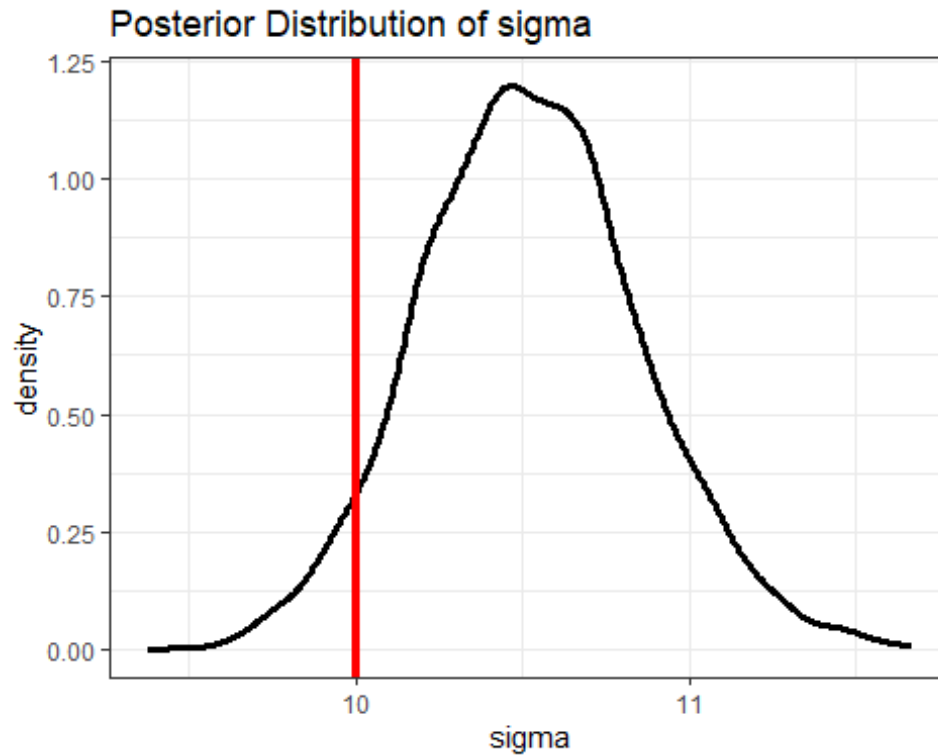
```



```

ggplot(df_posterior_custom,aes(x=sigma_chain))+
  geom_density(size=1.2)+
  theme_bw()+xlab("sigma")+
  labs(title = "Posterior Distribution of sigma") +
  theme(legend.title = element_blank(),
  legend.position = "top")+
  geom_vline(xintercept = 10,size=1.5,color="red")

```



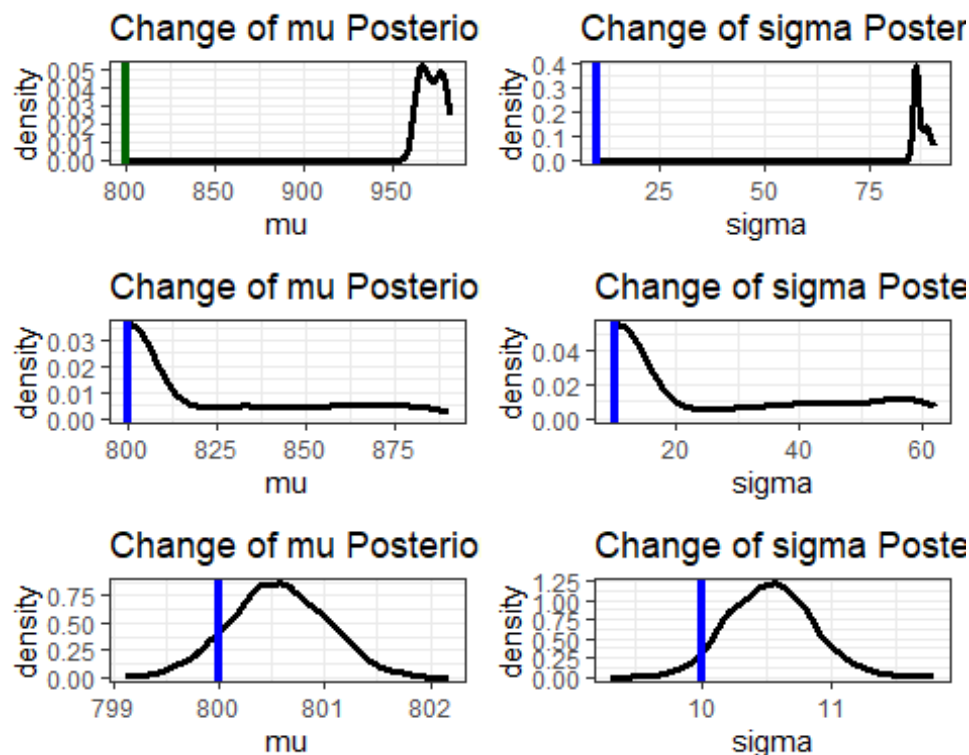
```
df1_custom <- HMC_custom(y=y_custom,n=length(y_custom), # data
  m=1000,s=20,a=10,b=2, # priors
  step=0.02, # step-size
  L=12, # no. of leapfrog steps
  initial_q=c(1000,11), # Chain initialization
  nsamp=100, # total number of samples
  nburn=34) # number of burn-in samples
df2_custom <- HMC_custom(y=y_custom,n=length(y_custom), # 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=1000, # total number of samples
  nburn=334) # number of burn-in samples
df3_custom <- HMC_custom(y=y_custom,n=length(y_custom), # 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

p1_custom <- ggplot(df1_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
```

```

labs(title = "Change of mu Posterior with nsamp = 100") +
theme(legend.title = element_blank(), legend.position = "top") +
geom_vline(xintercept = 800, size = 1.5, color = "darkgreen")
p2_custom <- ggplot(df1_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with nsamp = 100") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
p3_custom <- ggplot(df2_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
  labs(title = "Change of mu Posterior with nsamp = 1000") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 800, size = 1.5, color = "blue")
p4_custom <- ggplot(df2_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with nsamp = 1000") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
p5_custom <- ggplot(df3_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
  labs(title = "Change of mu Posterior with nsamp = 6000") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 800, size = 1.5, color = "blue")
p6_custom <- ggplot(df3_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with nsamp = 6000") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
# Arrange plots in a grid
grid.arrange(p1_custom, p2_custom, p3_custom, p4_custom, p5_custom,
p6_custom, nrow = 3)

```

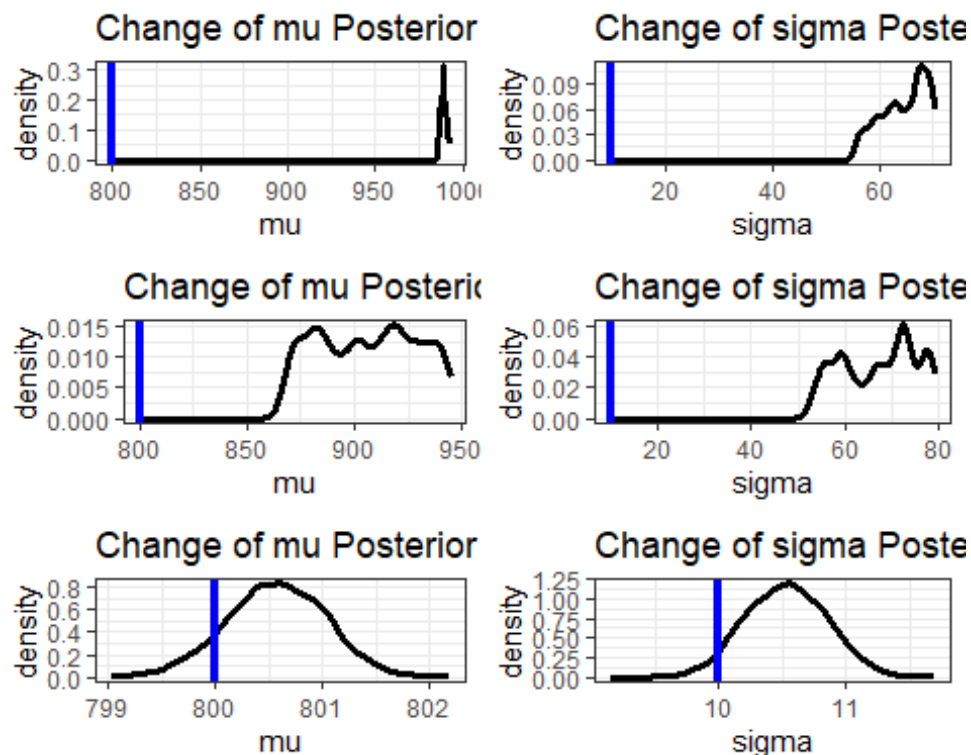



```
df_1_custom <- HMC_custom(y=y_custom,n=length(y_custom), # data
  m=1000,s=20,a=10,b=2, # priors
  step=0.001, # step-size
  L=12, # no. of Leapfrog steps
  initial_q=c(1000,11), # Chain initialization
  nsamp=6000, # total number of samples
  nburn=2000) # number of burn-in samples
df_2_custom <- HMC_custom(y=y_custom,n=length(y_custom), # data
  m=1000,s=20,a=10,b=2, # priors
  step=0.005, # 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
df_3_custom <- HMC_custom(y=y_custom,n=length(y_custom), # 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
p1_custom <- ggplot(df_1_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
  labs(title = "Change of mu Posterior with step = 0.001") +
  theme(legend.title = element_blank(), legend.position = "top") +
```

```

  geom_vline(xintercept = 800, size = 1.5, color = "blue")
p2_custom <- ggplot(df_1_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with step = 0.001") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
p3_custom <- ggplot(df_2_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
  labs(title = "Change of mu Posterior with step = 0.005") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 800, size = 1.5, color = "blue")
p4_custom <- ggplot(df_2_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with step = 0.005") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
p5_custom <- ggplot(df_3_custom, aes(x = mu_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("mu") +
  labs(title = "Change of mu Posterior with step = 0.02") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 800, size = 1.5, color = "blue")
p6_custom <- ggplot(df_3_custom, aes(x = sigma_chain)) +
  geom_density(size = 1.2) +
  theme_bw() +
  xlab("sigma") +
  labs(title = "Change of sigma Posterior with step = 0.02") +
  theme(legend.title = element_blank(), legend.position = "top") +
  geom_vline(xintercept = 10, size = 1.5, color = "blue")
grid.arrange(p1_custom, p2_custom, p3_custom, p4_custom, p5_custom,
p6_custom, nrow = 3)

```



```
df_1_custom$id <- 2001:6000
p1_custom <- ggplot(df_1_custom,aes(x=id,y=mu_chain))+
  geom_line(size=1.2,color="red")+
  theme_bw()+xlab("mu chain")
labs(title = "Change of mu Posterior with step = 0.001")

## $title
## [1] "Change of mu Posterior with step = 0.001"
##
## attr(,"class")
## [1] "labels"

df_2_custom$id <- 2001:6000
p2_custom <- ggplot(df_1_custom,aes(x=id,y=sigma_chain))+
  geom_line(size=1.2,color="red")+
  theme_bw()+xlab("sigma chain")
labs(title = "Change of sigma Posterior with step = 0.001")

## $title
## [1] "Change of sigma Posterior with step = 0.001"
##
## attr(,"class")
## [1] "labels"

df_3_custom$id <- 2001:6000
p3_custom <- ggplot(df_2_custom,aes(x=id,y=mu_chain))+
  geom_line(size=1.2,color="red")+
```

```

  theme_bw()+xlab("mu chain")
labs(title = "Change of mu Posterior with step = 0.005")

## $title
## [1] "Change of mu Posterior with step = 0.005"
##
## attr(,"class")
## [1] "labels"

p4_custom <- ggplot(df_2_custom,aes(x=id,y=sigma_chain))+
  geom_line(size=1.2,color="red")+
  theme_bw()+xlab("sigma chain")
labs(title = "Change of sigma Posterior with step = 0.005")

## $title
## [1] "Change of sigma Posterior with step = 0.005"
##
## attr(,"class")
## [1] "labels"

p5_custom <- ggplot(df_3_custom,aes(x=id,y=mu_chain))+
  geom_line(size=1.2,color="red")+
  theme_bw()+xlab("mu chain")
labs(title = "Change of mu Posterior with step = 0.02")

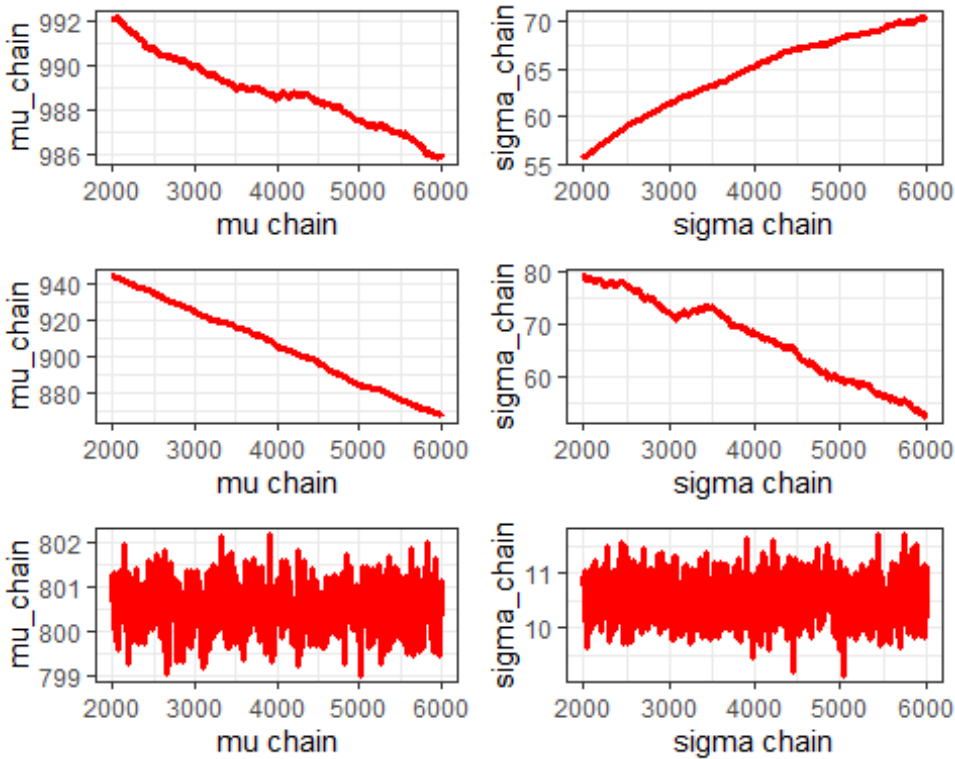
## $title
## [1] "Change of mu Posterior with step = 0.02"
##
## attr(,"class")
## [1] "labels"

p6_custom <- ggplot(df_3_custom,aes(x=id,y=sigma_chain))+
  geom_line(size=1.2,color="red")+
  theme_bw()+xlab("sigma chain")
labs(title = "Change of sigma Posterior with step = 0.02")

## $title
## [1] "Change of sigma Posterior with step = 0.02"
##
## attr(,"class")
## [1] "labels"

grid.arrange(p1_custom, p2_custom, p3_custom, p4_custom, p5_custom,
p6_custom,nrow = 3)

```



```
# Install and load required packages if not already installed
run_and_plot_hmc_custom <- function(m_prior, s_prior, title) {
  df_pos_custom <- HMC_custom(y = y_custom, n = length(y_custom),
    m = m_prior, s = s_prior, a = 10, b = 2,
    step = 0.02, L = 12,
    initial_q = c(1000, 11), nsamp = 6000, nburn = 2000)

  ggplot(df_pos_custom, aes(x = mu_chain)) +
    geom_density(size = 1.2) +
    theme_bw() + xlab("mu") +
    ggtitle(title) +
    geom_vline(xintercept = 800, size = 1.5, color = "red") +
    theme(legend.title = element_blank(),
    legend.position = "top")
}

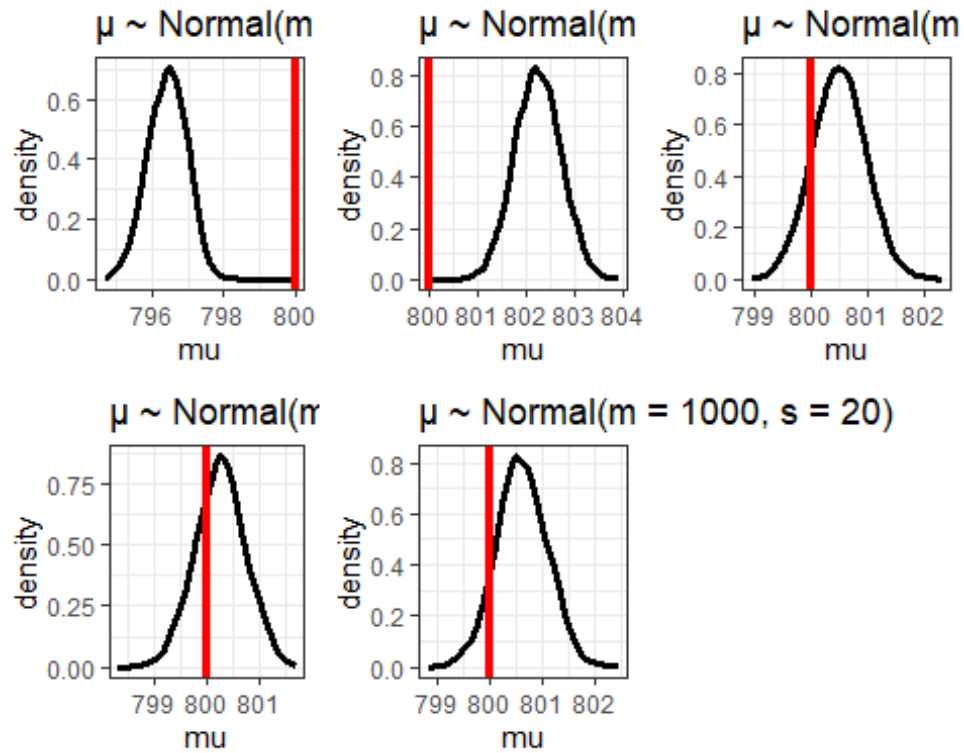
# Run HMC for five different priors and plot posterior distributions
prior_params_custom <- list(
  c(400, 5),
  c(400, 20),
  c(1000, 5),
  c(1000, 20),
  c(1000, 100)
)

titles_custom <- c("μ ~ Normal(m = 400, s = 5)",
  "μ ~ Normal(m = 400, s = 20)",
  "μ ~ Normal(m = 1000, s = 5)",
```

```

"μ ~ Normal(m = 1000, s = 20)",
"μ ~ Normal(m = 1000, s = 100)"
plot_list_custom <- lapply(1:5, function(i) {
  run_and_plot_hmc_custom(prior_params_custom[[i]][1],
prior_params_custom[[i]][2],
titles_custom[i])
})
# Plot all posterior distributions
multiplot_custom <- function(..., plotlist = NULL, file, cols = 1, layout =
NULL) {
  library(grid)
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  if (is.null(layout)) {
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots == 1) {
print(plots[[1]])
  } else {
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    for (i in 1:numPlots) {
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
layout.pos.col = matchidx$col))
    }
  }
}
multiplot_custom(plotlist = plot_list_custom, cols = 3)

```



Increasing the step-size improves the stability of the posterior density graphs for μ and σ . Conversely, decreasing the step-size leads to more and more upward or downward trends in these graphs.