

CGS698C - Assignment 3

Sahil Tomar (210898)

2024-06-21

Part 1 - Estimating the posterior distribution using different computational methods

We are given:

- the data : $y = \{10, 15, 15, 14, 14, 14, 13, 11, 12, 16\}$
- the likelihood assumption : $y_i \sim \text{Binomial}(n = 20, \theta)$
- The prior assumption : $\theta \sim \text{Beta}(1, 1)$
- Analytically Derived Posterior Distribution : $\theta|y \sim \text{Beta}(135, 67)$

1.1

Graph of the analytically derived posterior function is shown below -

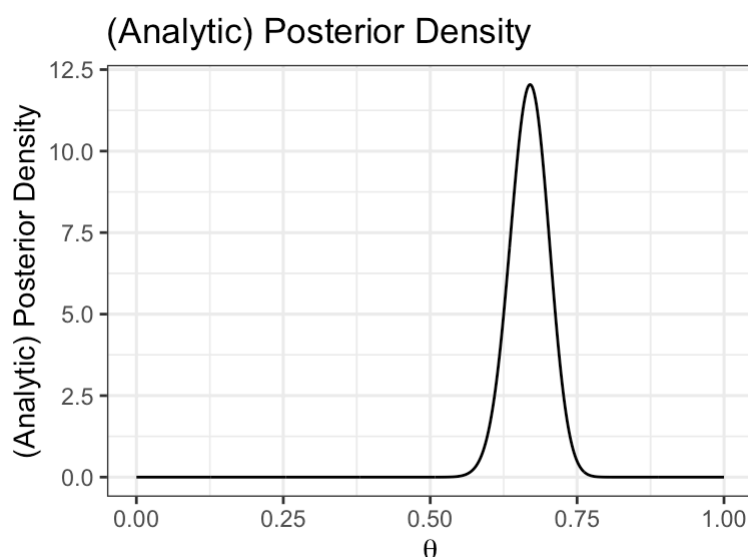
```
y1 <- c(10, 15, 15, 14, 14, 14, 13, 11, 12, 16)

theta_grid <- seq(from=0, to=1, by=0.001)

df1.posterior <- data.frame(theta = theta_grid)

df1.posterior$analytic <- dbeta(df1.posterior$theta, 135, 67)

ggplot(df1.posterior, aes(x=theta, y=analytic)) + geom_line() +
  theme_bw() + xlab(expression(theta)) + ylab("(Analytic) Posterior Density")+
  ggtitle("(Analytic) Posterior Density")
```



1.2

Grid Point Approximation for the given case is -

```

df1.posterior$likelihood <- rep(NA, length(theta_grid))
df1.posterior$prior <- rep(NA, length(theta_grid))
df1.posterior$posterior <- rep(NA, length(theta_grid))

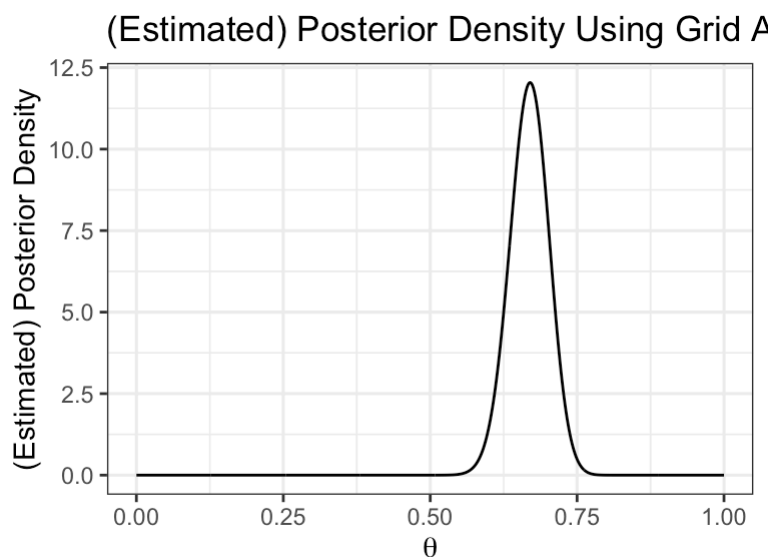
for (i in 1:length(theta_grid))
{
  df1.posterior$likelihood[i] <- prod(dbinom(y1, size = 20, prob = theta_grid[i]))
  df1.posterior$prior[i] <- dbeta(theta_grid[i], shape1=1, shape2=1)
  df1.posterior$posterior[i] <- df1.posterior$likelihood[i] * df1.posterior$prior[i]
}

ML_GA <- sum(df1.posterior$posterior* (theta_grid[2]-theta_grid[1]))

for (i in 1:length(theta_grid))
{
  df1.posterior$posterior[i] <- df1.posterior$posterior[i]/ML_GA
}

ggplot(df1.posterior, aes(x=theta, y=posterior)) + geom_line()+
  theme_bw() + xlab(expression(theta)) + ylab("(Estimated) Posterior Density")+
  ggtitle("(Estimated) Posterior Density Using Grid Approximation")

```



1.3

```

nsamp <- 10000

df1.estimate <- data.frame(theta_sample = rbeta(4*nsamp, 1, 1))
df1.estimate$likelihood <- NA
for (i in 1:length(df1.estimate$theta_sample))
{
  df1.estimate$likelihood[i] <- prod(dbinom(y1, 20, df1.estimate$theta_sample[i]))
}

ML_MC <- mean(df1.estimate$likelihood)

```

Using Monte-Carlo Integration to estimate, the marginal likelihood comes out to be 1.413e-10

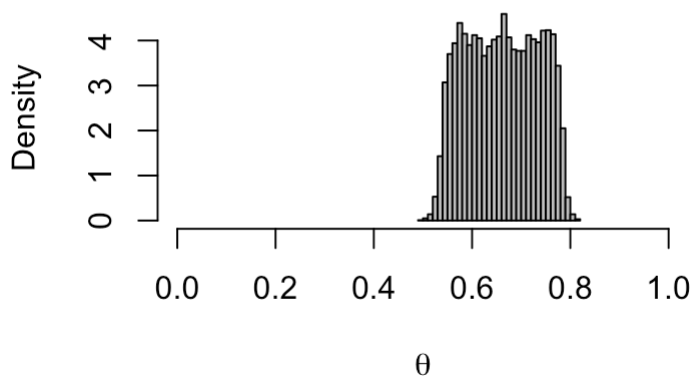
The marginal likelihood calculated during Grid Approximation was 1.406e-10).

1.4

Using Importance Sampling to sample from the posterior density (the proposal density is the prior itself) -

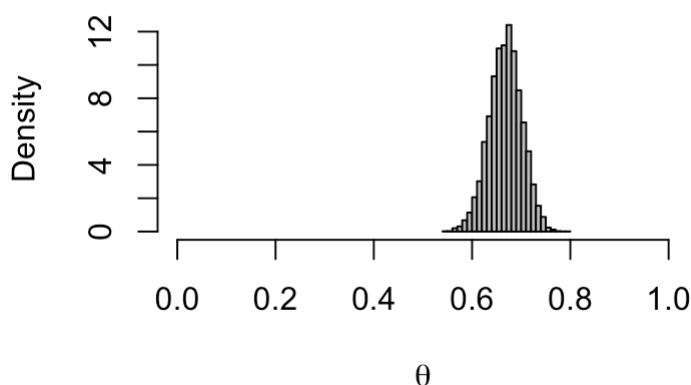
```
estimated_posterior <- sample(df1.estimate$theta_sample,  
                             size = nrow(df1.estimate)/4,  
                             prob = df1.estimate$likelihood)  
hist(estimated_posterior, freq=FALSE, main="Importance Sampling", col = "gray", border="black", xlab = expression(theta), breaks=25, xlim = c(0, 1))
```

Importance Sampling



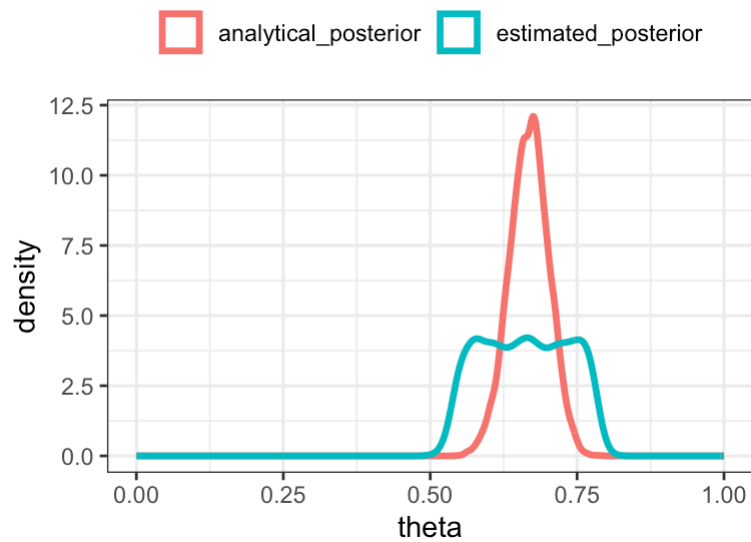
```
analytical_posterior <- rbeta(nrow(df1.estimate)/4,135,67)  
hist(analytical_posterior, freq=FALSE, main="(Analytical) Posterior Density", col =  
"gray", border="black", xlab = expression(theta), breaks=25, xlim = c(0, 1))
```

(Analytical) Posterior Density



```
posteriors <- data.frame(analytical_posterior,estimated_posterior)  
ggplot(melt(posterior),aes(x=value,colour=variable))+  
  geom_density(linewidth=1.2)+ xlim(c(0,1)) +theme_bw()+  
  xlab("theta")+theme(legend.title = element_blank(),  
                      legend.position = "top")
```

```
## No id variables; using all as measure variables
```



1.5

Using the Markov Chain Monte-Carlo Method to sample from the posterior distribution of θ

Using the following function to get the θ -chain

```

nsamp <- 10000

mcmc <- function(y,step_size){

  theta_chain <- rep(NA,nsamp)
  reject <- 0

  theta_chain[1] <- rbeta(1,1,1)
  i <- 1

  while(i < nsamp){
    proposal_theta <-
      rtruncnorm(1,a=0,b=1,mean=theta_chain[i],sd=step_size)
    lkl_proposal <- sum(dbinom(y,20, proposal_theta,log=TRUE))
    prior_proposal <- dbeta(proposal_theta,1,1, log=TRUE)
    forward_density <-
      dtruncnorm(proposal_theta,a=0,b=Inf,mean=theta_chain[i],sd=step_size)
    lkl_current <- sum(dbinom(y,20, theta_chain[i], log=TRUE))
    prior_current <- dbeta(theta_chain[i],1,1, log=TRUE)
    backward_density <-
      dtruncnorm(theta_chain[i],a=0,b=Inf,mean=proposal_theta,sd=step_size)
    H <- (exp(lkl_proposal+prior_proposal - (lkl_current+prior_current)))*(backward_d
ensity)/(forward_density)
    astr <- runif(1,0,1)
    if(astr<min(1,H)){
      theta_chain[i+1] <- proposal_theta
      i <- i+1
    }else{
      reject <- reject+1
    }
  }
  return(list(theta_chain=theta_chain,reject=reject))
}

```

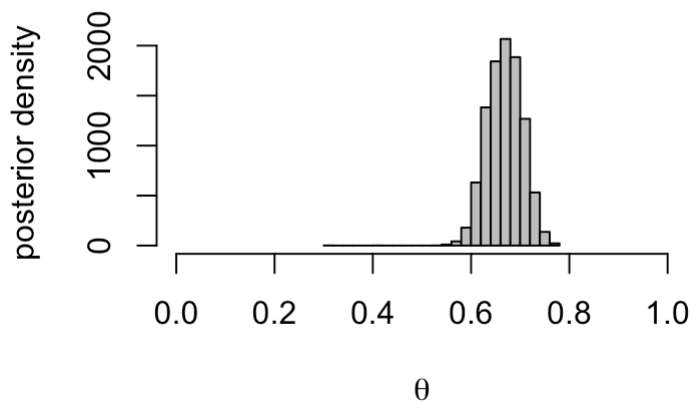
Run the function for the given data and plotting the estimated posterior density

```

markov_chain <- mcmc(y = y1, step_size = 0.08)
hist(markov_chain$theta_chain, col = "gray", border="black", breaks=20, xlim=(c(0,
1)), xlab = expression(theta),ylab="posterior density", main = "Markov Chain Monte-Ca
rlo")

```

Markov Chain Monte-Carlo

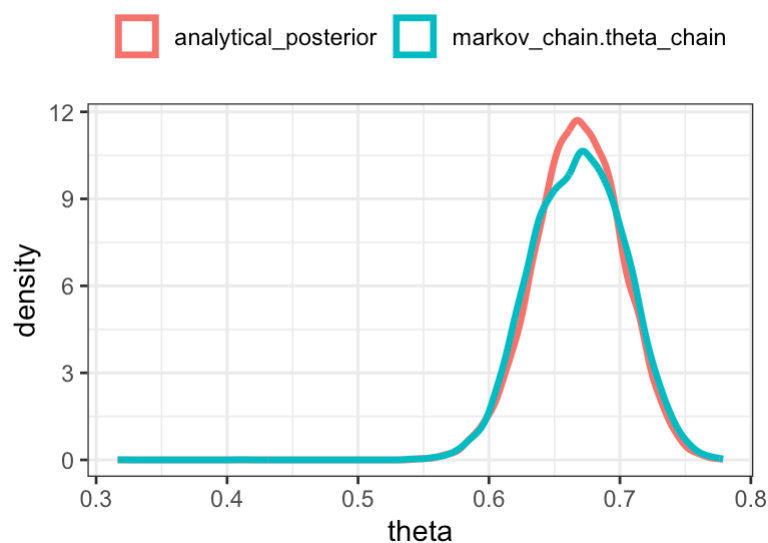


Plotting the comparison between the analytic and estimated posterior densities

```
analytical_posterior <- rbeta(nsamp,135,67)
posteriors <- data.frame(analytical_posterior,markov_chain$theta_chain)
ggplot(melt(posterior),aes(x=value,colour=variable))+
  geom_density(size=1.2)+theme_bw()+
  xlab("theta")+theme(legend.title = element_blank(),
                      legend.position = "top")
```

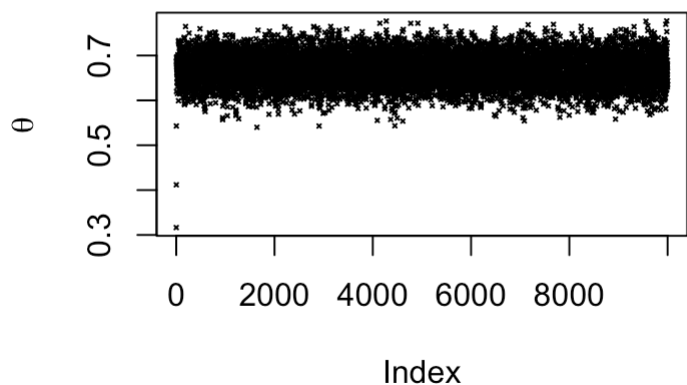
```
## No id variables; using all as measure variables
```

```
## 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.
```



Check if the Markov Chain is proper or not

```
plot(markov_chain$theta_chain, pch=4, cex=0.25, ylab = expression(theta), xlab="Index")
```



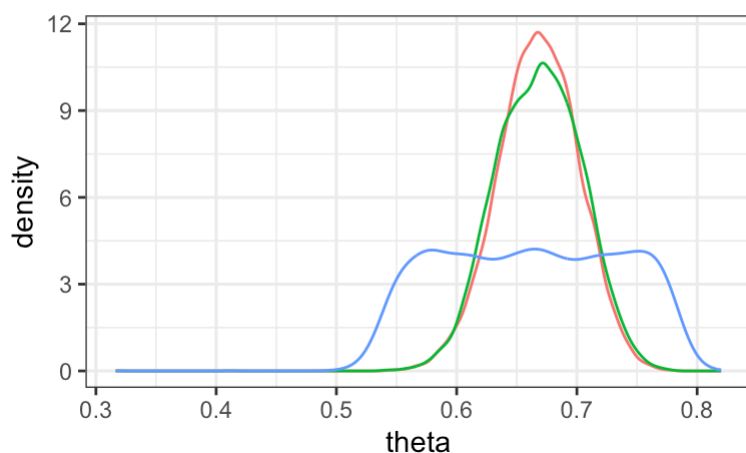
1.6

Plotting the Analytic posterior density, the estimated posterior density (using markov chain) and the estimated posterior density (using importance sampling)

```
posterior <- data.frame(analytical_posterior, markov_chain$theta_chain, estimated_pos-
terior)
colnames(posterior) <- c("Analytical Posterior", "Estimated Posterior \nUsing Markov
Chain Monte-Carlo", "Estimated Posterior \nUsing Importance Sampling")
ggplot(melt(posterior), aes(x=value, colour=variable)) +
  geom_density() + theme_bw() +
  xlab("theta") + theme(legend.title = element_blank(),
    legend.position = "top")
```

```
## No id variables; using all as measure variables
```

lytical Posterior □ Estimated Posterior □ Estimated
Using Markov Chain Monte-Carlo Using Impc



Part 2: Writing your own sampler for Bayesian

inference

2.1

The research problem - Is the mean recognition time for the non-words larger than the mean recognition time for the words?

2.2

Lexical-access hypothesis - The mean recognition time for the words is longer than the mean recognition time for the non-words.

2.3 -

Null Hypothesis Model

- Likelihood Assumptions: $RT_i \sim Normal(\mu_i, \sigma)$ where $\mu_i = \alpha + \beta \cdot type_i$
- Prior Assumptions : $\alpha \sim Normal(400, 50), \sigma = 50, \beta \sim Normal_+(0, 50)$

2.4

Data

```
dat <- read.table(
  "https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main/notes/Data/word-re
  cognition-times.csv", sep=";", header = T)[-1]

dat$typei <- NA

for (i in 1:nrow(dat))
{
  dat$typei[i] <- ifelse(dat$type[i] == "word", 0, 1)
}

flextable::flextable(head(dat))
```

type	RT	typei
word	423.1019	0
word	429.9432	0
non-word	486.9959	1
non-word	451.4400	1
non-word	482.2657	1
non-word	470.8003	1

2.5

2.5.1

Using Markov Chain - Monte Carlo to estimate α and β posterior distributions

```
RT <- as.numeric(dat$RT)
typei <- as.integer(dat$typei)
```

Using the following function to generate α -chain and β -chain

```

mcmc <- function(RT, typei, step_size, nsamp){

  alpha_chain <- rep(NA, nsamp)
  beta_chain <- rep(NA, nsamp)
  sigma <- 30

  reject <- 0

  alpha_chain[1] <- rnorm(1, 400, 50)
  beta_chain[1] <- rtruncnorm(1, a=0, b=Inf, mean=0, sd=50)
  i <- 1

  while(i < nsamp){

    proposal_alpha <- rnorm(1, mean=alpha_chain[i], sd = step_size)
    proposal_beta <- rtruncnorm(n=1, mean=beta_chain[i], sd=step_size, a=0, b=Inf)

    lkl_proposal <- sum(dnorm(RT, mean = proposal_alpha + (proposal_beta*typei), sd=sigma, log=TRUE))
    prior_proposal <- dnorm(proposal_alpha, 400, 50)*dtruncnorm(proposal_beta, a=0, b=Inf, mean=0, sd=50)
    forward_density <-
      dnorm(proposal_alpha, mean=alpha_chain[i], sd=step_size)*dtruncnorm(proposal_beta, a=0, b=Inf, mean=beta_chain[i], sd=step_size)

    lkl_current <- sum(dnorm(RT, mean = alpha_chain[i] + (beta_chain[i]*typei), sd=sigma, log = TRUE))
    prior_current <- dnorm(alpha_chain[i], 400, 50) * dtruncnorm(beta_chain[i], a=0, b=Inf, mean=0, sd=50)
    backward_density <-
      dnorm(alpha_chain[i], mean=proposal_alpha, sd=step_size)*dtruncnorm(beta_chain[i], a=0, b=Inf, mean=proposal_beta, sd=step_size)

    H <- exp(lkl_proposal-lkl_current)*(prior_proposal*backward_density)/(prior_current*forward_density)

    astr <- runif(1,0,1)

    if(astr<min(1,H)){
      alpha_chain[i+1] <- proposal_alpha
      beta_chain[i+1] <- proposal_beta
      i <- i+1
    }

    else{
      reject <- reject+1
    }
  }

  return(list(alpha_chain=alpha_chain, beta_chain=beta_chain, reject=reject))
}

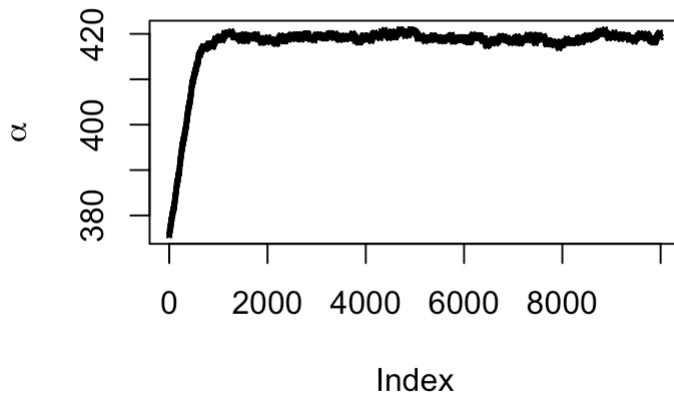
```

Run the function to get the markov chains

```
markov_chain <- mcmc(RT, typei, 0.1, nsamp)
```

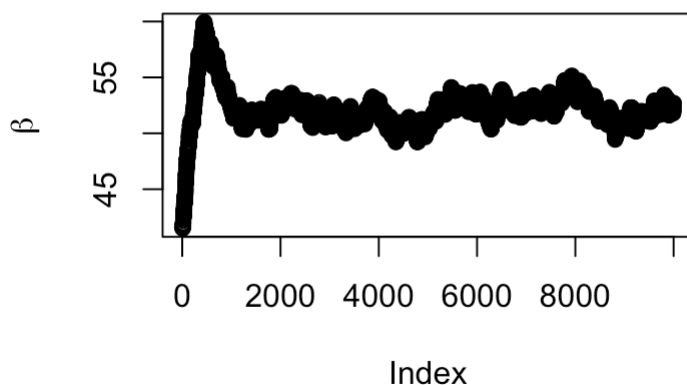
Plotting the α -chain to check its convergence

```
plot(markov_chain$alpha_chain, pch=4, cex=0.25, ylab = expression(alpha), xlab="Index")
```



Plotting the β -chain to check its convergence

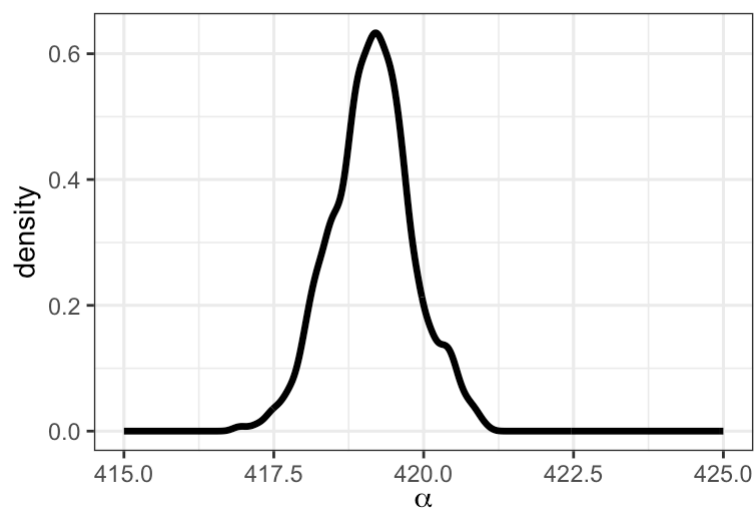
```
plot(markov_chain$beta_chain, pch=1, ylab = expression(beta), xlab="Index")
```



Plotting the posterior distributions of α and β as estimated by the MCMC

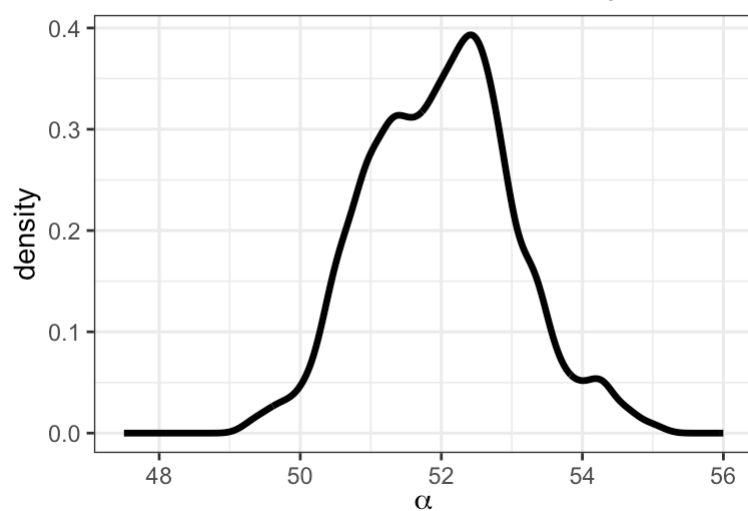
```
posteriors <- data.frame(alpha_chain=markov_chain$alpha_chain,beta_chain=markov_chain$beta_chain)
ggplot(posteriors[-(1:2000),],aes(x=alpha_chain))+
  geom_density(size=1.2)+ xlim(c(415,425)) +
  theme_bw()+xlab(expression(alpha))+ ggtitle("Posterior distribution estimated by MC MC")+
  theme(legend.title = element_blank(),
        legend.position = "top")
```

Posterior distribution estimated by MCMC



```
ggplot(posterioriors[-(1:2000),],aes(x=beta_chain))+
  geom_density(size=1.2)+ xlim(c(47.5,56)) +
  theme_bw()+xlab(expression(alpha))+ ggtitle("Posterior distribution estimated by MC
MC")+
  theme(legend.title = element_blank(),
        legend.position = "top")
```

Posterior distribution estimated by MCMC



2.5.2

```
range_alpha <- quantile(markov_chain$alpha_chain,probs=c(.025,.975))
range_alpha
```

```
##      2.5%      97.5%
## 394.0711 420.4752
```

The 95% quantile range for α is [394.0711011, 420.4751731]

```
range_beta <- quantile(markov_chain$beta_chain,probs=c(.025,.975))
range_beta
```

```
##      2.5%    97.5%  
## 49.78672 57.02159
```

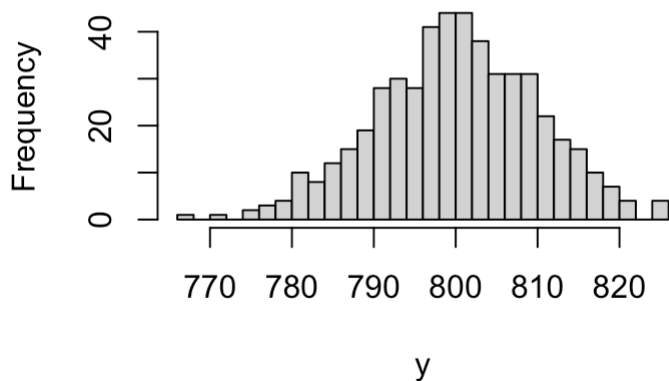
The 95% quantile range for β is [49.7867212, 57.0215869]

Part 2 - Hamiltonian Monte-Carlo

Data Generation -

```
true_mu <- 800  
true_var <- 100 #sigma^2  
y <- rnorm(500, mean=true_mu, sd=sqrt(true_var))  
hist(y, breaks=25)
```

Histogram of y



Gradient Function

```
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  
}
```

Hamiltonian MC Sampler

```

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

  mu_chain[1] <- initial_q[1]
  sigma_chain[1] <- initial_q[2]

  i <- 1
  while(i < nsamp){

    q <- c(mu_chain[i],sigma_chain[i])
    p <- rnorm(length(q),0,1)

    current_q <- q
    current_p <- p

    current_V = V(current_q[1],current_q[2],y,n,m,s,a,b)
    current_T = sum(current_p^2)/2

    for(l in 1:L){

      p <- p-((step/2)*gradient(q[1],q[2],y,n,m,s,a,b))

      q <- q + step*p

      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_T = sum(proposed_p^2)/2

    accept.prob <- min(1,exp(current_V+current_T-proposed_V-proposed_T))

    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
}

```

Exercise 2.1

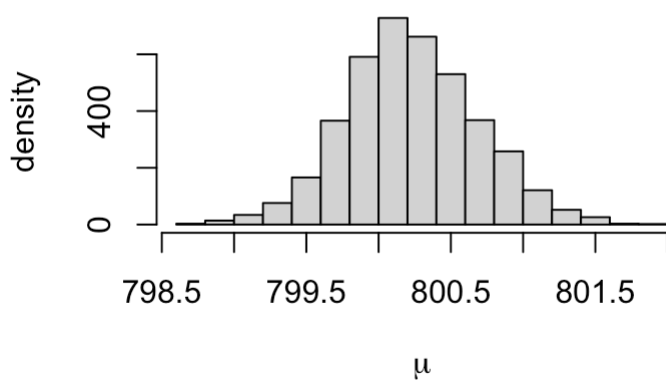
Run the sampler with the given parameters

```
df.posterior <- HMC(y=y,n=length(y),
                    m=1000,s=20,a=10,b=2,
                    step=0.02,
                    L=12,
                    initial_q=c(1000,11),
                    nsamp=6000,
                    nburn=2000)
```

Plotting the posterior for μ

```
hist(df.posterior$mu_chain, xlab=expression(mu), ylab="density", main="Posterior Density using HMC")
```

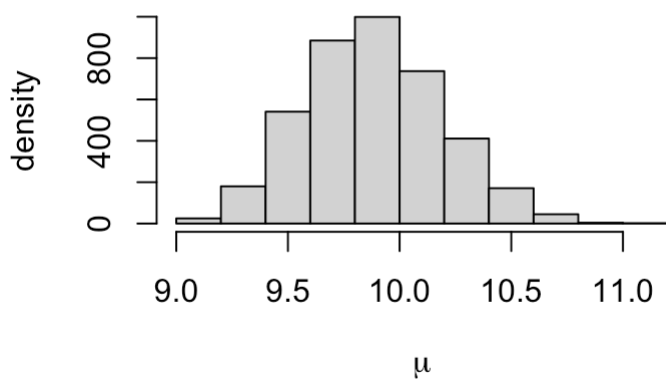
Posterior Density using HMC



Plotting the posterior for σ

```
hist(df.posterior$sigma_chain, xlab=expression(mu), ylab="density", main="Posterior D  
ensity using HMC")
```

Posterior Density using HMC



Exercise 2.2

Comparing how the HMC is sensitive to number of samples -

```

# Perform HMC sampling for different nsamp values
df_posterior_100 <- HMC(y=y, n=length(y),
                        m=1000, s=20, a=10, b=2,
                        step=0.02,
                        L=12,
                        initial_q=c(1000,11),
                        nsamp=100,
                        nburn=33)

df_posterior_1000 <- HMC(y=y, n=length(y),
                        m=1000, s=20, a=10, b=2,
                        step=0.02,
                        L=12,
                        initial_q=c(1000,11),
                        nsamp=1000,
                        nburn=333)

df_posterior_6000 <- HMC(y=y, n=length(y),
                        m=1000, s=20, a=10, b=2,
                        step=0.02,
                        L=12,
                        initial_q=c(1000,11),
                        nsamp=6000,
                        nburn=2000)

# Extract mu_chain and sigma_chain from each posterior
mu_chain_100 <- df_posterior_100$mu_chain
sigma_chain_100 <- df_posterior_100$sigma_chain

mu_chain_1000 <- df_posterior_1000$mu_chain
sigma_chain_1000 <- df_posterior_1000$sigma_chain

mu_chain_6000 <- df_posterior_6000$mu_chain
sigma_chain_6000 <- df_posterior_6000$sigma_chain

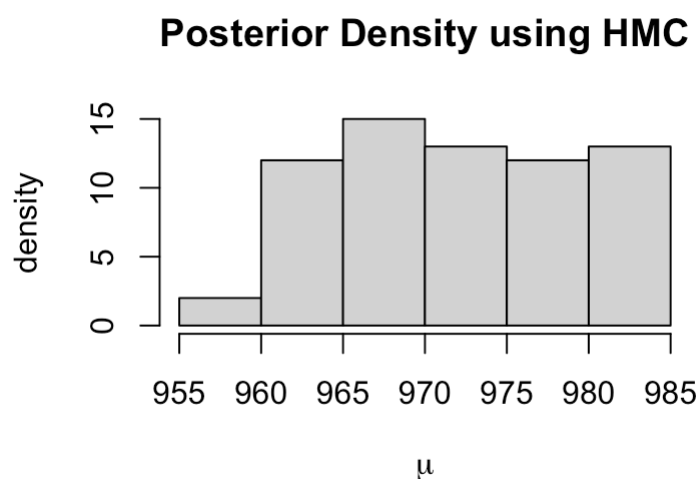
```

Plot the posteriors of μ for all the values of sample size

```

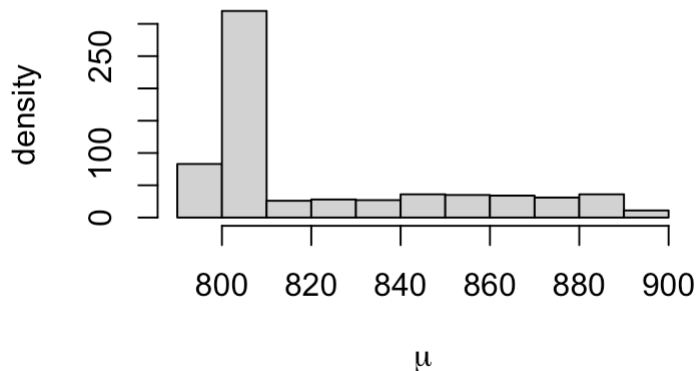
hist(mu_chain_100, xlab=expression(mu), ylab="density", main="Posterior Density using
HMC")

```



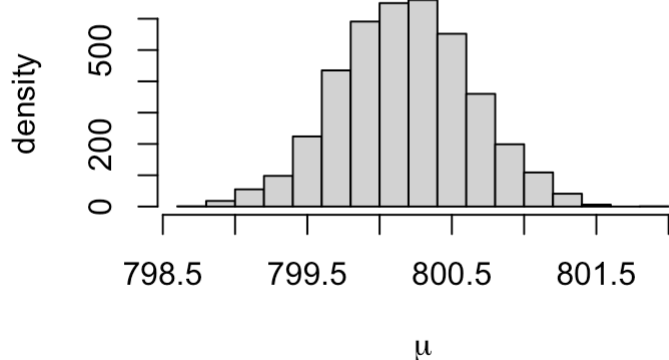

```
hist(mu_chain_1000, xlab=expression(mu), ylab="density", main="Posterior Density using HMC")
```

Posterior Density using HMC



```
hist(mu_chain_6000, xlab=expression(mu), ylab="density", main="Posterior Density using HMC")
```

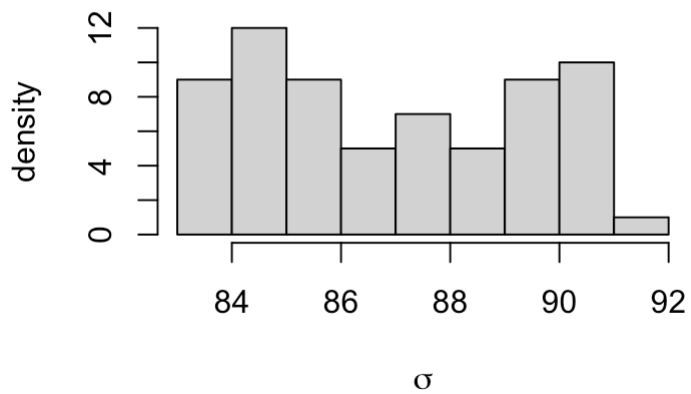
Posterior Density using HMC



Plot the posteriors of σ for all the values of sample size

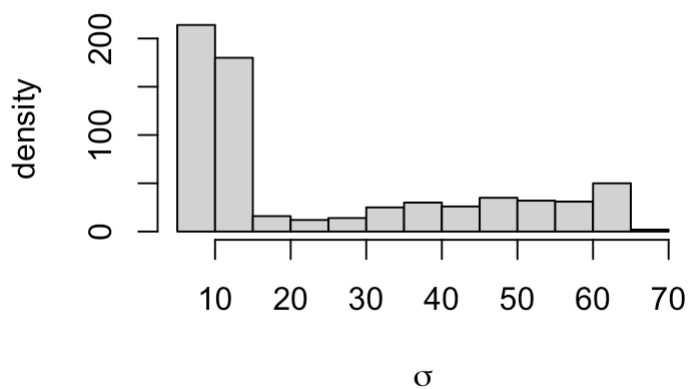
```
hist(sigma_chain_100, xlab=expression(sigma), ylab="density", main="Posterior Density using HMC")
```

Posterior Density using HMC



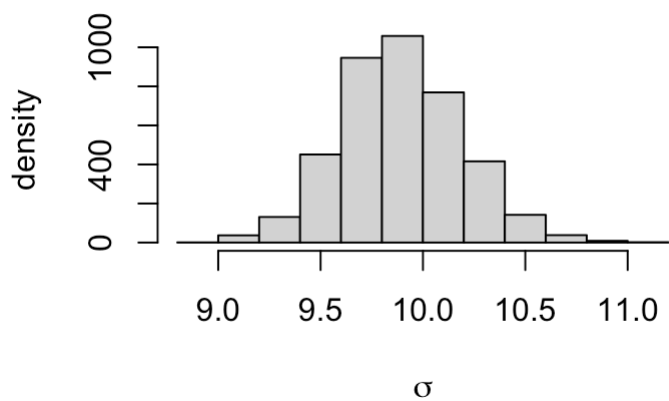
```
hist(sigma_chain_1000, xlab=expression(sigma), ylab="density", main="Posterior Density using HMC")
```

Posterior Density using HMC



```
hist(sigma_chain_6000, xlab=expression(sigma), ylab="density", main="Posterior Density using HMC")
```

Posterior Density using HMC



One can note that HMC sampler is very sensitive to sample size and sample size should be large enough to get a good estimation of parameters μ & σ . The estimation is very poor at a very small sample size.

Exercise 2.3

```
# Perform HMC sampling for different step values
df_posterior_step_001 <- HMC(y=y, n=length(y),
                             m=1000, s=20, a=10, b=2,
                             step=0.001,
                             L=12,
                             initial_q=c(1000,11),
                             nsamp=6000,
                             nburn=2000)

df_posterior_step_005 <- HMC(y=y, n=length(y),
                             m=1000, s=20, a=10, b=2,
                             step=0.005,
                             L=12,
                             initial_q=c(1000,11),
                             nsamp=6000,
                             nburn=2000)

df_posterior_step_02 <- HMC(y=y, n=length(y),
                             m=1000, s=20, a=10, b=2,
                             step=0.02,
                             L=12,
                             initial_q=c(1000,11),
                             nsamp=6000,
                             nburn=2000)

# Extract mu_chain and sigma_chain from each posterior
mu_chain_step_001 <- df_posterior_step_001$mu_chain
sigma_chain_step_001 <- df_posterior_step_001$sigma_chain

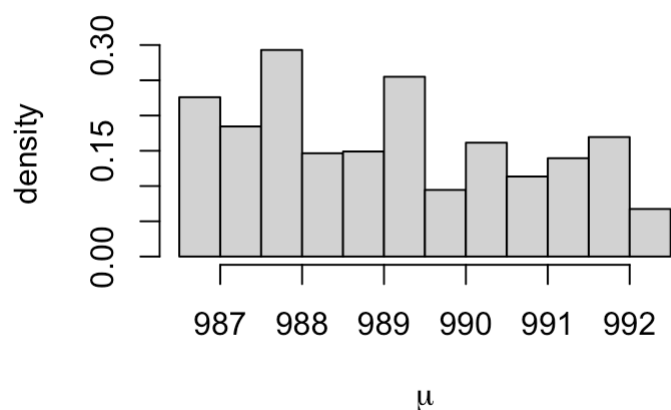
mu_chain_step_005 <- df_posterior_step_005$mu_chain
sigma_chain_step_005 <- df_posterior_step_005$sigma_chain

mu_chain_step_02 <- df_posterior_step_02$mu_chain
sigma_chain_step_02 <- df_posterior_step_02$sigma_chain
```

Plot the posteriors of μ for all the values of step size

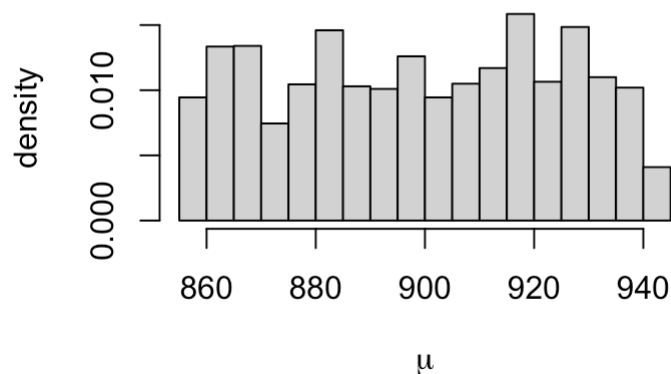
```
hist(mu_chain_step_001, xlab=expression(mu), ylab="density", main="Posterior Density
of mu using HMC (step=0.001)", freq=FALSE)
```

Posterior Density of mu using HMC (step=0



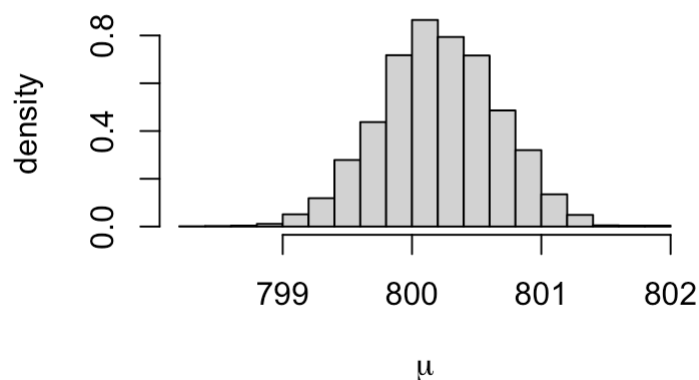
```
hist(mu_chain_step_005, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (step=0.005)", freq=FALSE)
```

Posterior Density of mu using HMC (step=0



```
hist(mu_chain_step_02, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (step=0.02)", freq=FALSE)
```

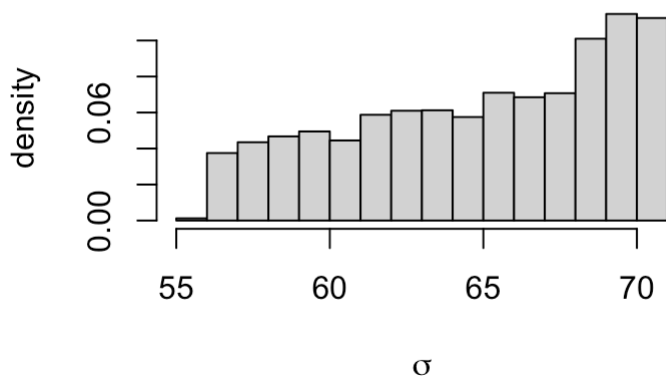
Posterior Density of mu using HMC (step=0



Plot the posteriors of σ for all the values of step size

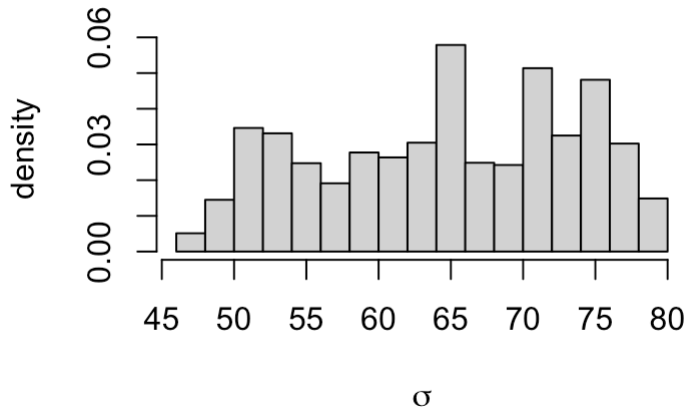
```
hist(sigma_chain_step_001, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (step=0.001)", freq=FALSE)
```

Posterior Density of sigma using HMC (step=



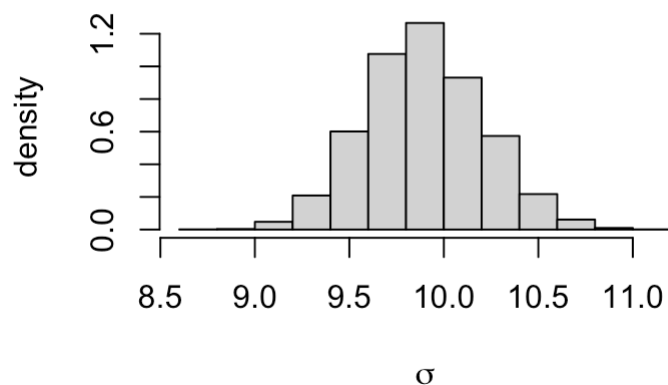
```
hist(sigma_chain_step_005, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (step=0.005)", freq=FALSE)
```

Posterior Density of sigma using HMC (step=



```
hist(sigma_chain_step_02, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (step=0.02)", freq=FALSE)
```

Posterior Density of sigma using HMC (step



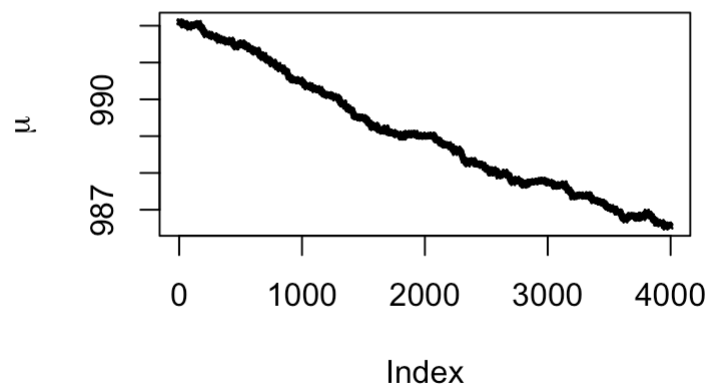
Again, one can note that HMC sampler is very

sensitive to step size and step size should be large enough to get a good estimation of parameters μ & σ . The estimation is very poor at a very small step size.

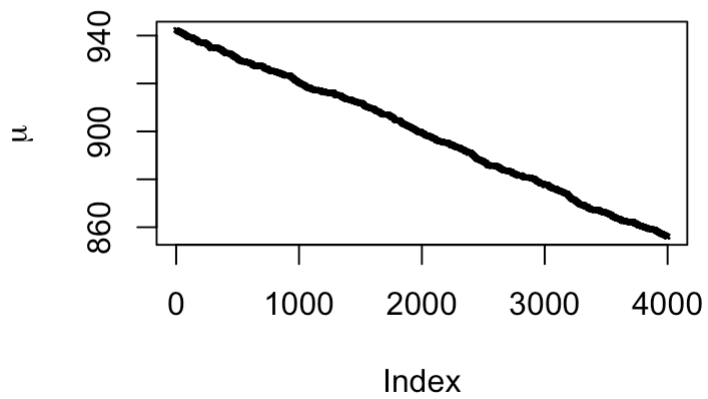
Exercise 2.4

Visually Inspecting the markov chains for μ

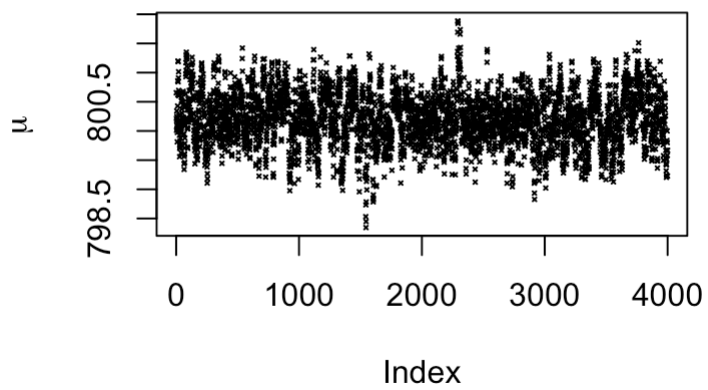
```
plot(mu_chain_step_001, pch=4, cex=0.25, ylab = expression(mu), xlab="Index")
```



```
plot(mu_chain_step_005, pch=4, cex=0.25, ylab = expression(mu), xlab="Index")
```

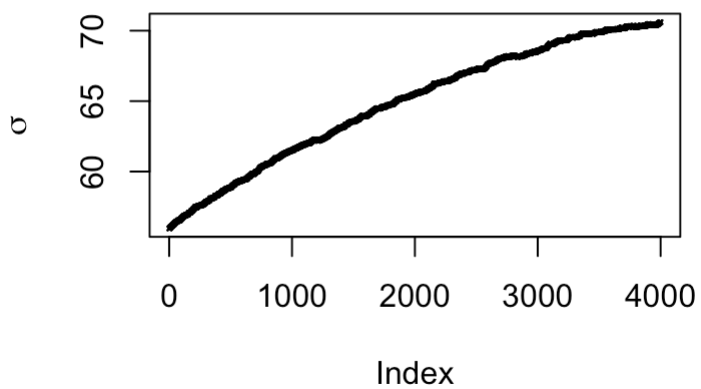


```
plot(mu_chain_step_02, pch=4, cex=0.25, ylab = expression(mu), xlab="Index")
```

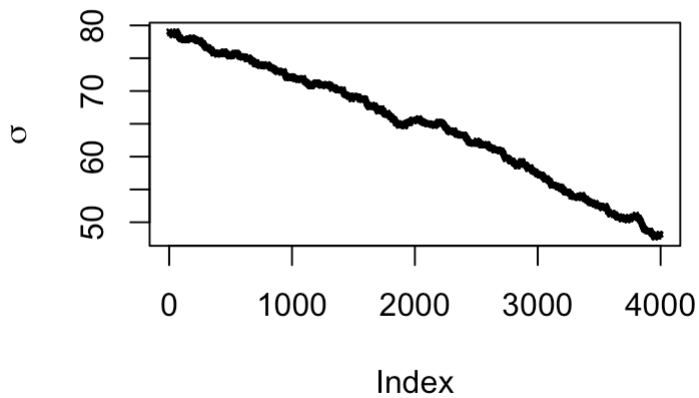


Visually Inspecting the markov chains for σ

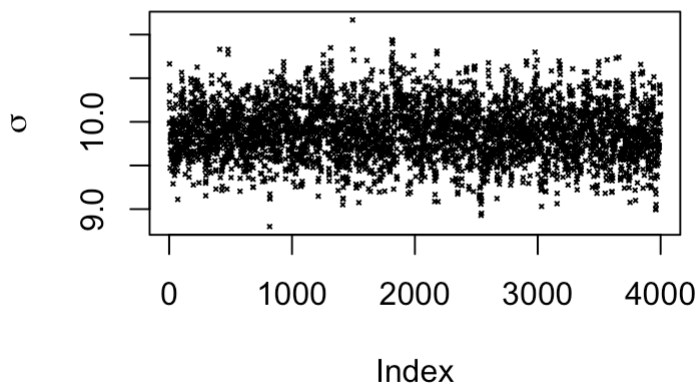
```
plot(sigma_chain_step_001, pch=4, cex=0.25, ylab = expression(sigma), xlab="Index")
```



```
plot(sigma_chain_step_005, pch=4, cex=0.25, ylab = expression(sigma), xlab="Index")
```



```
plot(sigma_chain_step_02, pch=4, cex=0.25, ylab = expression(sigma), xlab="Index")
```



Upon visual inspection of the markov chains, one can clearly note that within 6000 samples, the markov chain where the step size was insufficiently small, they fail to converge and hence give these very very far estimates of the parameters μ and σ .

Exercise 2.5

Plotting the estimated posteriors for different priors


```

# Perform HMC sampling for different priors on mu
df_posterior_prior_1 <- HMC(y=y, n=length(y),
                           m=400, s=5, a=10, b=2,
                           step=0.02,
                           L=12,
                           initial_q=c(1000,11),
                           nsamp=6000,
                           nburn=2000)

df_posterior_prior_2 <- HMC(y=y, n=length(y),
                           m=400, s=20, a=10, b=2,
                           step=0.02,
                           L=12,
                           initial_q=c(1000,11),
                           nsamp=6000,
                           nburn=2000)

df_posterior_prior_3 <- HMC(y=y, n=length(y),
                           m=1000, s=5, a=10, b=2,
                           step=0.02,
                           L=12,
                           initial_q=c(1000,11),
                           nsamp=6000,
                           nburn=2000)

df_posterior_prior_4 <- HMC(y=y, n=length(y),
                           m=1000, s=20, a=10, b=2,
                           step=0.02,
                           L=12,
                           initial_q=c(1000,11),
                           nsamp=6000,
                           nburn=2000)

df_posterior_prior_5 <- HMC(y=y, n=length(y),
                           m=1000, s=100, a=10, b=2,
                           step=0.02,
                           L=12,
                           initial_q=c(1000,11),
                           nsamp=6000,
                           nburn=2000)

# Extract mu_chain and sigma_chain from each posterior
mu_chain_prior_1 <- df_posterior_prior_1$mu_chain
sigma_chain_prior_1 <- df_posterior_prior_1$sigma_chain

mu_chain_prior_2 <- df_posterior_prior_2$mu_chain
sigma_chain_prior_2 <- df_posterior_prior_2$sigma_chain

mu_chain_prior_3 <- df_posterior_prior_3$mu_chain
sigma_chain_prior_3 <- df_posterior_prior_3$sigma_chain

mu_chain_prior_4 <- df_posterior_prior_4$mu_chain
sigma_chain_prior_4 <- df_posterior_prior_4$sigma_chain

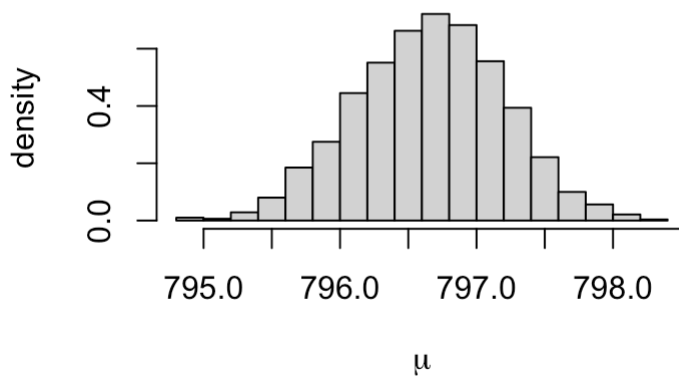
```

```
mu_chain_prior_5 <- df_posterior_prior_5$mu_chain
sigma_chain_prior_5 <- df_posterior_prior_5$sigma_chain
```

Plot the posteriors of μ for all the different priors

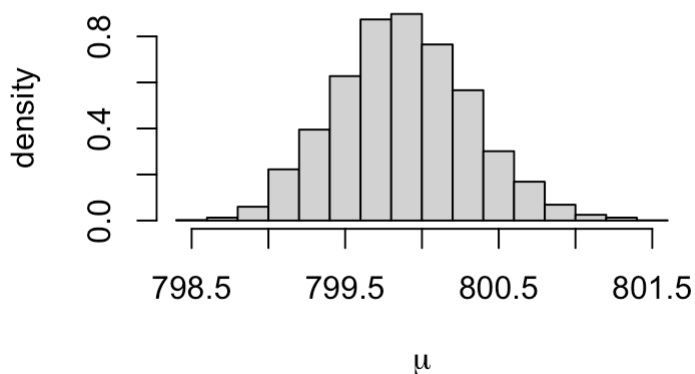
```
hist(mu_chain_prior_1, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (m=400, s=5)", freq=FALSE)
```

Posterior Density of mu using HMC (m=400



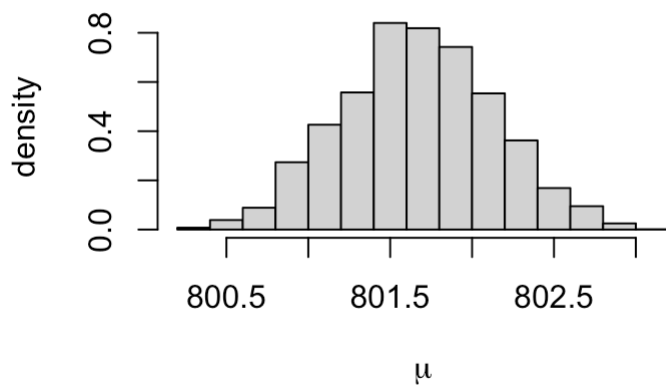
```
hist(mu_chain_prior_2, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (m=400, s=20)", freq=FALSE)
```

Posterior Density of mu using HMC (m=400,



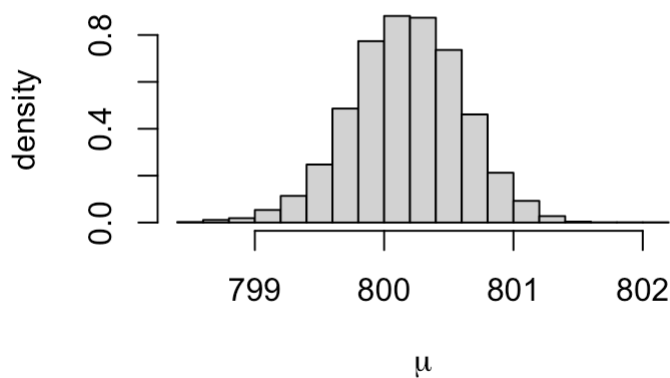
```
hist(mu_chain_prior_3, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (m=1000, s=5)", freq=FALSE)
```

Posterior Density of mu using HMC (m=1000, s=20)



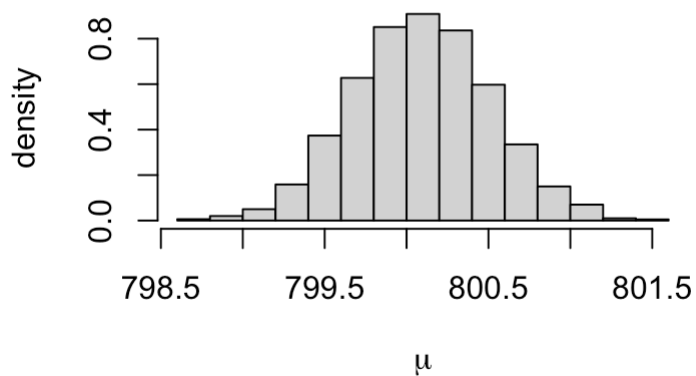
```
hist(mu_chain_prior_4, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (m=1000, s=20)", freq=FALSE)
```

Posterior Density of mu using HMC (m=1000, s=100)



```
hist(mu_chain_prior_5, xlab=expression(mu), ylab="density", main="Posterior Density of mu using HMC (m=1000, s=100)", freq=FALSE)
```

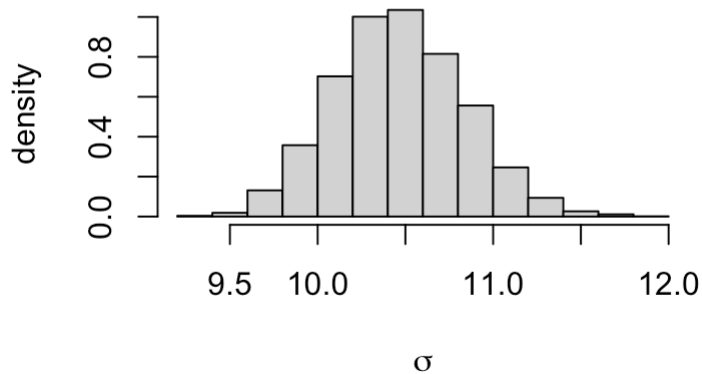
Posterior Density of mu using HMC (m=1000, s=1000)



Plot the posteriors of σ for all the different priors

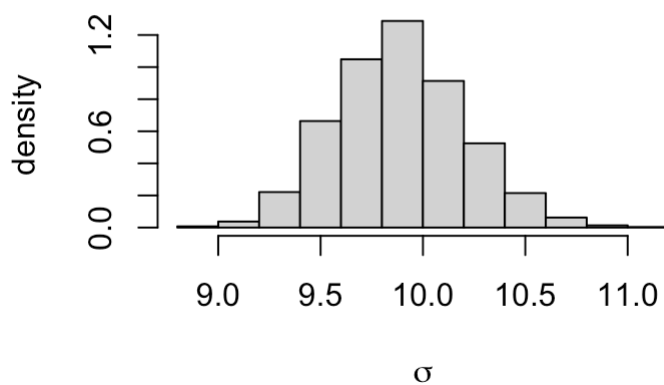
```
hist(sigma_chain_prior_1, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (m=400, s=5)", freq=FALSE)
```

Posterior Density of sigma using HMC (m=400, s=5)



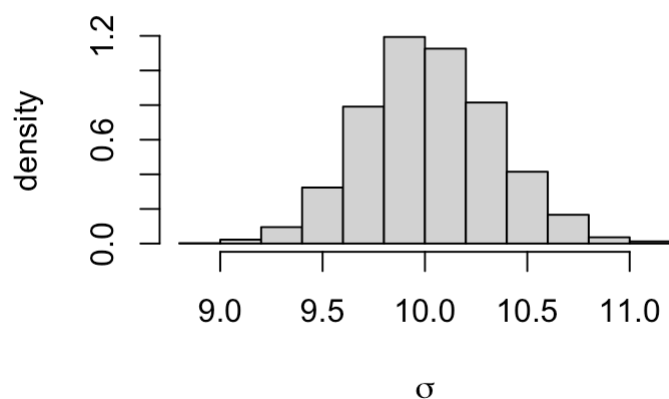
```
hist(sigma_chain_prior_2, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (m=400, s=20)", freq=FALSE)
```

Posterior Density of sigma using HMC (m=400, s=20)



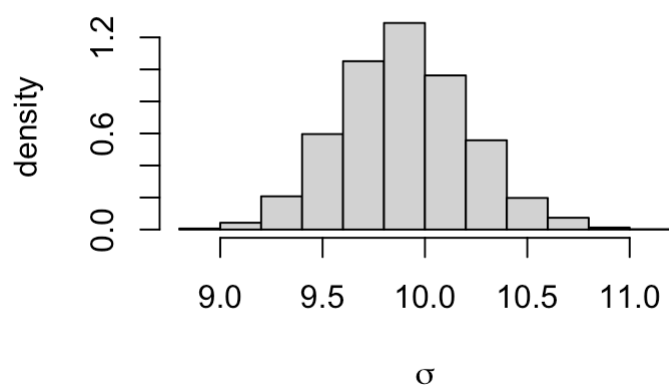
```
hist(sigma_chain_prior_3, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (m=1000, s=5)", freq=FALSE)
```

Posterior Density of sigma using HMC (m=1000, s=20)



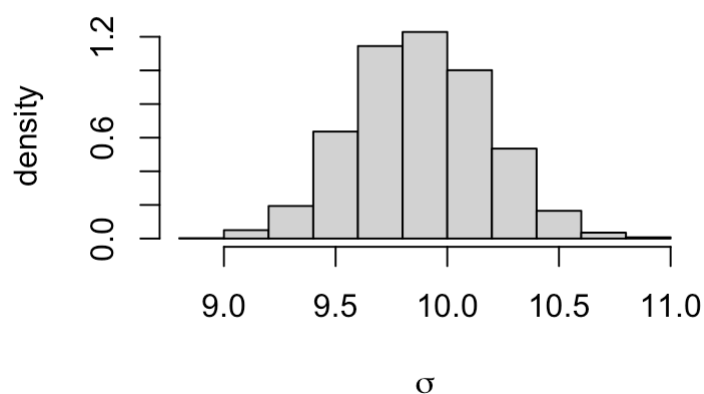
```
hist(sigma_chain_prior_4, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (m=1000, s=20)", freq=FALSE)
```

Posterior Density of sigma using HMC (m=1000, s=100)



```
hist(sigma_chain_prior_5, xlab=expression(sigma), ylab="density", main="Posterior Density of sigma using HMC (m=1000, s=100)", freq=FALSE)
```

Posterior Density of sigma using HMC (m=1000, s=1000)



When the prior for μ was tight - i.e., with small standard deviation s), the posterior distribution of μ was influenced by the prior mean m . But when the prior for μ was loose - i.e., with larger standard deviation s), the posterior distribution of μ was very weakly influenced by the prior mean m . The posterior distribution of σ was not affected by the change in the prior of μ . The shape and center of the posterior distribution for σ remained relatively consistent