MAST30027 Modern Applied Statistics Assignment 4

Patrick Tjahjadi (890003)

Tutor: Qiuyi Li — Tutorial: Monday 12:00 PM

1 Question One

$f(x_i) = \frac{\sqrt{\tau}}{\sqrt{2\pi}} e^{-\frac{\tau(x-\mu)^2}{2}}$
$\propto e^{-\frac{\tau(x-\mu)^2}{2}}$
$\implies f(x_1, ..., x_{100} \mid \mu, \tau) \propto \prod_{i=1}^{n} e^{-\frac{\tau(x-\mu)^2}{2}}$ where $n = 100$

1.1 Part A

$f(\tau, \mu, x_1, ..., x_{100}) = f(x_1, ..., x_{100} \mid \mu, \tau)\ p(\tau, \mu)$
$\propto \prod_{i=1}^{n}(e^{-\frac{\tau(x-\mu)^2}{2}})\ (\tau)^{-1}$

$p(\mu \mid \tau, x_1, ..., x_{100}) = \frac{f(\mu,\tau,x_1,...,x_{100})}{f(\tau,x_1,...,x_{100})}$
$\propto f(\mu, \tau, x_1, ..., x_{100})$
$\propto \prod_{i=1}^{n}(e^{-\frac{\tau(x-\mu)^2}{2}})\ (\tau)^{-1}$
$\propto e^{-\sum_{i=1}^{n}(\frac{\tau}{2}(x_i-\mu)^2)}$
$= e^{-\sum_{i=1}^{n}(\frac{\tau}{2}(x_i^2 - 2\mu x_i + \mu^2)}$
Given that $\sum x_i = n\bar{x}$,
$= e^{-\sum_{i=1}^{n}(\frac{\tau}{2}x_i^2)}\ e^{-(-2\mu n\bar{x} + n\mu^2)\frac{\tau}{2}}$
$= e^{-\sum_{i=1}^{n}(\frac{\tau}{2}x_i^2)}\ e^{n(2\mu\bar{x} - \mu^2)\frac{\tau}{2}}$
$= e^{-\sum_{i=1}^{n}(\frac{\tau}{2}x_i^2)}\ e^{n(2\mu\bar{x} - \mu^2 + \bar{x}^2 - \bar{x}^2)\frac{\tau}{2}}$
$= e^{-\sum_{i=1}^{n}(\frac{\tau}{2}x_i^2)}\ e^{n(2\mu\bar{x} - \mu^2 - \bar{x}^2)\frac{\tau}{2}}\ e^{\bar{x}^2\frac{\tau}{2}}$
$\propto e^{n(2\mu\bar{x} - \mu^2 - \bar{x}^2)\frac{\tau}{2}}$
$= e^{-n(\bar{x}+\mu)^2\frac{\tau}{2}}$
$= e^{-\frac{(\mu-\bar{x}^2)}{2\frac{1}{\tau n}}}$

This belongs to a normal distribution with mean $\bar{x}$ and variance $\frac{1}{\tau n}$.

$p(\tau \mid \mu, x_1, ..., x_{100}) = \frac{f(\mu,\tau,x_1,...,x_{100})}{f(\mu,x_1,...,x_{100})}$
$\propto f(\mu, \tau, x_1, ..., x_{100})$
$\propto \prod_{i=1}^{n}(\sqrt{\tau}e^{-\frac{\tau(x-\mu)^2}{2}})\ (\tau)^{-1}$
$= \tau^{\frac{n}{2}} e^{-\tau\sum_{i=1}^{n}(\frac{1}{2}(x_i-\mu)^2)}\ (\tau)^{-1}$
$= \tau^{(\frac{n}{2}-1)} e^{-\tau\sum_{i=1}^{n}(\frac{1}{2}(x_i-\mu)^2)}$

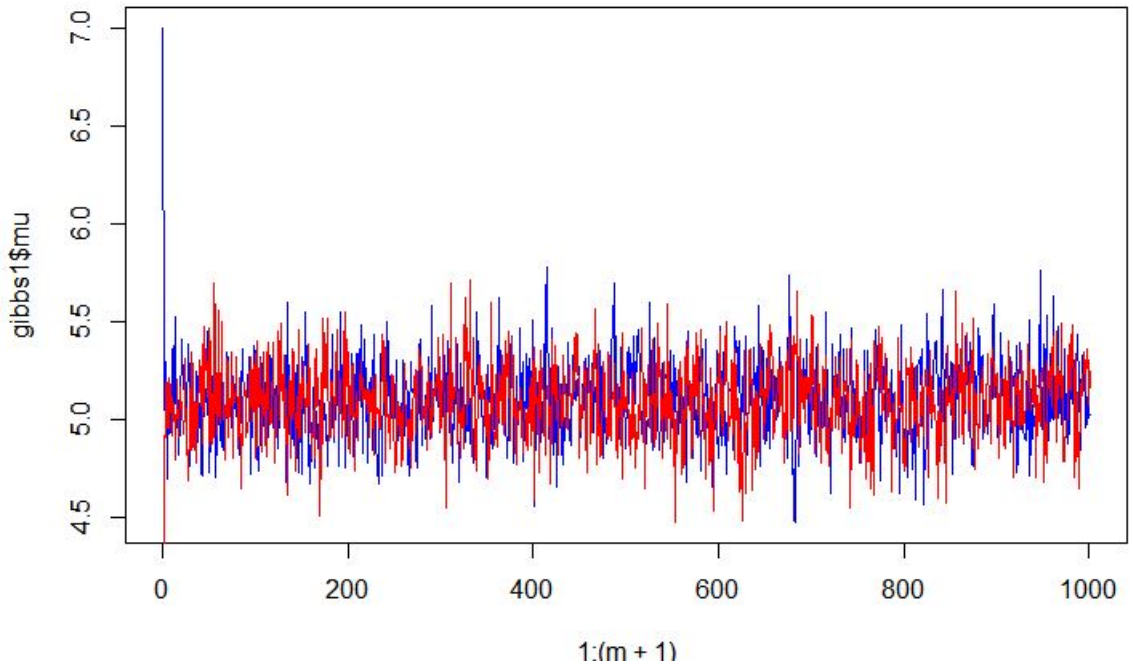
This belongs to a gamma distribution with shape $\frac{n}{2}$ and rate $\sum_{i=1}^{n}(\frac{1}{2}(x_i-\mu)^2)$.

1.2 Part B

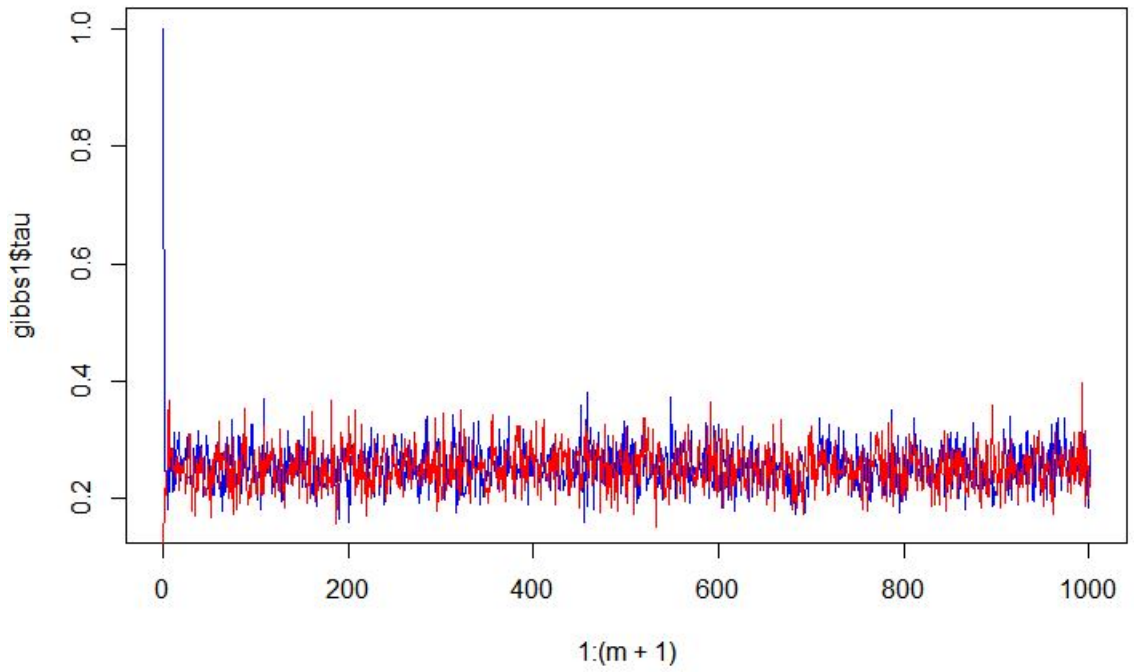
A code that uses Gibbs sampling is derived from Heejung Shim's simulation_Gibbs.pdf. The code is as follows:

```r
n = 100
xbar = 0
gamma_rate = 0
mu = 5
for(i in 1:n) {
  xbar = xbar + Assign4Data[i]
  gamma_rate = gamma_rate + 0.5 * (Assign4Data[i] - mu)^2
}
xbar = as.numeric(xbar)/n
gamma_rate = as.numeric(gamma_rate)
tau = 0.25

#-------------------------------------------------------------
gibbs.f2 = function(mu0, tau0, m){
  mu.seq = tau.seq = rep(-1, m+1)
  mu.seq[1] = mu0
  tau.seq[1] = tau0
  for(j in 2:(m+1)) {
    mu.seq[j] = rnorm(1, xbar, sqrt(1/(n*tau.seq[j-1])))
    tau.seq[j] = rgamma(1, n/2, gamma_rate)
  }
  result = list(mu = mu.seq, tau = tau.seq)
  result
}

#m = number of samples of (mu, tau) values.
m = 1000
#Perform Gibbs sampling
gibbs1 = gibbs.f2(7, 1, m)
gibbs2 = gibbs.f2(0.5, 0.1, m)


plot(1:(m+1), gibbs1$mu, type='l', col='blue')
points(1:(m+1), gibbs2$mu, type='l', col='red')


plot(1:(m+1), gibbs1$tau, type='l', col='blue')
points(1:(m+1), gibbs2$tau, type='l', col='red')
```

1000 samples is performed for this simulation. Notice that no burn-in is used in this code.
Here, gibbs1 has inital values that are significantly higher than the actual values while gibbs2 has initial values that are significantly lower. The graph for mu becomes:



The graph for tau becomes:



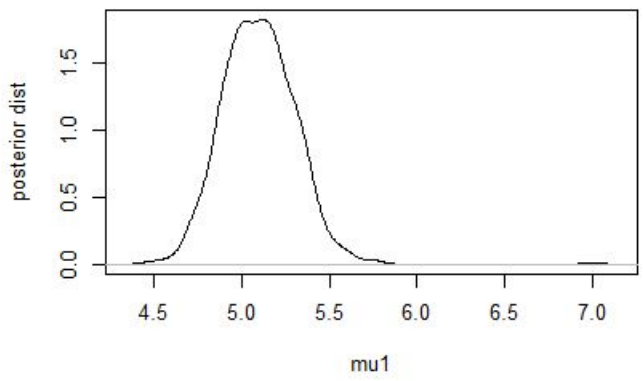
We can see that the Gibbs samples converge to approximately the actual values after a few samples for both chains. Hence, these samples from two different chains are mixed well and behave similarly.

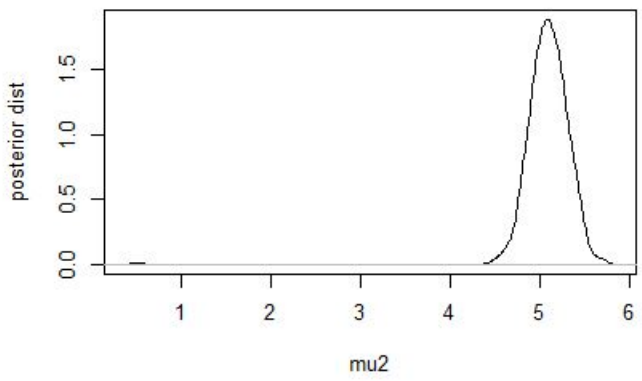
1.3 Part C

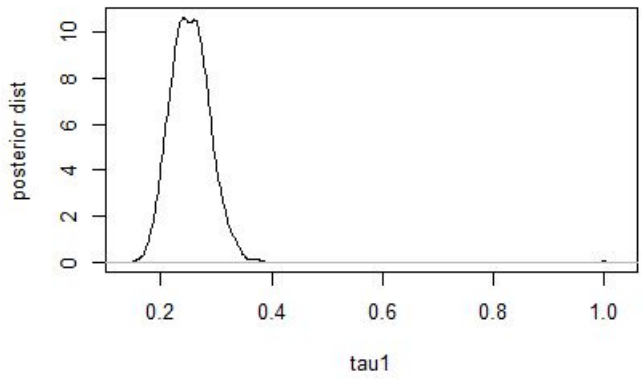
The code is as follows:

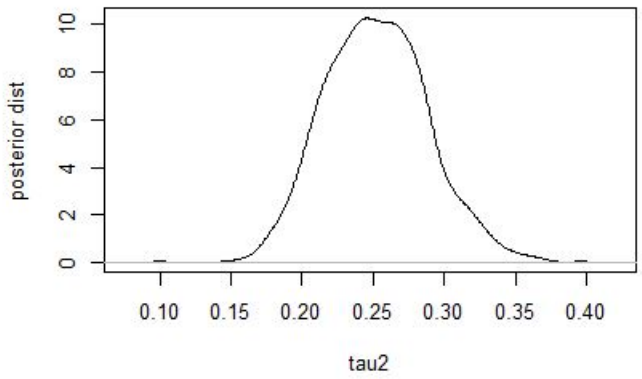
```
# Plots for marginal posterior distributions
par(mfrow=c(2,2))
plot(density(gibbs1$mu), ylab = "posterior dist", xlab = "mu1", main="")
plot(density(gibbs2$mu), ylab = "posterior dist", xlab = "mu2", main="")
plot(density(gibbs1$tau), ylab = "posterior dist", xlab = "tau1", main="")
plot(density(gibbs2$tau), ylab = "posterior dist", xlab = "tau2", main="")

# Marginal posterior means
mean(gibbs1$mu)
mean(gibbs2$mu)
mean(gibbs1$tau)
mean(gibbs2$tau)

#Fidning credible interval percentage for mu and tau
cdfmu = ecdf(gibbs1$mu)
cdfmu(c(4.77, 5.42))

cdftau = ecdf(gibbs1$tau)
cdftau(c(0.199,0.312))
```

Plotting the marginal posterior distributions yield the following graphs:









The marginal posterior means and credible intervals are:

```
> # Marginal posterior means
> mean(gibbs1$mu)
[1] 5.099271
> mean(gibbs2$mu)
[1] 5.088324
> mean(gibbs1$tau)
[1] 0.2536919
> mean(gibbs2$tau)
[1] 0.2520363
>
> #Fidning credible interval percentage for mu and tau
> cdfmu = ecdf(gibbs1$mu)
> cdfmu(c(4.77, 5.42))
[1] 0.04895105 0.94805195
>
> cdftau = ecdf(gibbs1$tau)
> cdftau(c(0.199,0.312))
[1] 0.04895105 0.95004995
>
```

This shows that the 95% credible interval for mu is approximately (4.77, 5.42) and the 95% credible interval for tau is approximately (0.199, 0.312).

2 Question Two

To determine the acceptance probability for this algorithm:
We are simulating Normal(μ_c, τ_n) for μ and Gamma($5\tau_c$, 5) for τ.
The proposed function Q will be symmetric, so we will be implementing a Random Walk MH algorithm:

$$\text{A (Acceptance Probability)} = \min(1, \frac{\pi(\theta')}{\pi(\theta)}) = \min(1, \frac{\pi(\mu',\tau')}{\pi(\mu,\tau)})$$

$\pi(\mu', \tau') = f(\mu, \tau, x_1, ..., x_{100})$
$= p(\tau', \mu') * p(x_1, ..., x_{100} \mid \mu', \tau')$

$\implies \log\pi(\mu', \tau') = \log p(\tau', \mu') + \log p(x_1, ..., x_{100} \mid \mu', \tau')$
$\propto \log(\frac{1}{\tau'}) + \sum_{i=1}^{n} log(p(x_i|\mu', \tau'))$

$\implies \pi(\mu', \tau') = e^{(\log\frac{1}{\tau'}) + \sum_{i=1}^{n} log(p(x_i|\mu', \tau'))}$ and $\pi(\mu, \tau) = e^{(\log\frac{1}{\tau}) + \sum_{i=1}^{n} log(p(x_i|\mu, \tau))}$

$\therefore A = \min(1, \frac{e^{(\log\frac{1}{\tau'}) + \sum_{i=1}^{n} log(p(x_i|\mu', \tau'))}}{e^{(\log\frac{1}{\tau}) + \sum_{i=1}^{n} log(p(x_i|\mu, \tau))}})$

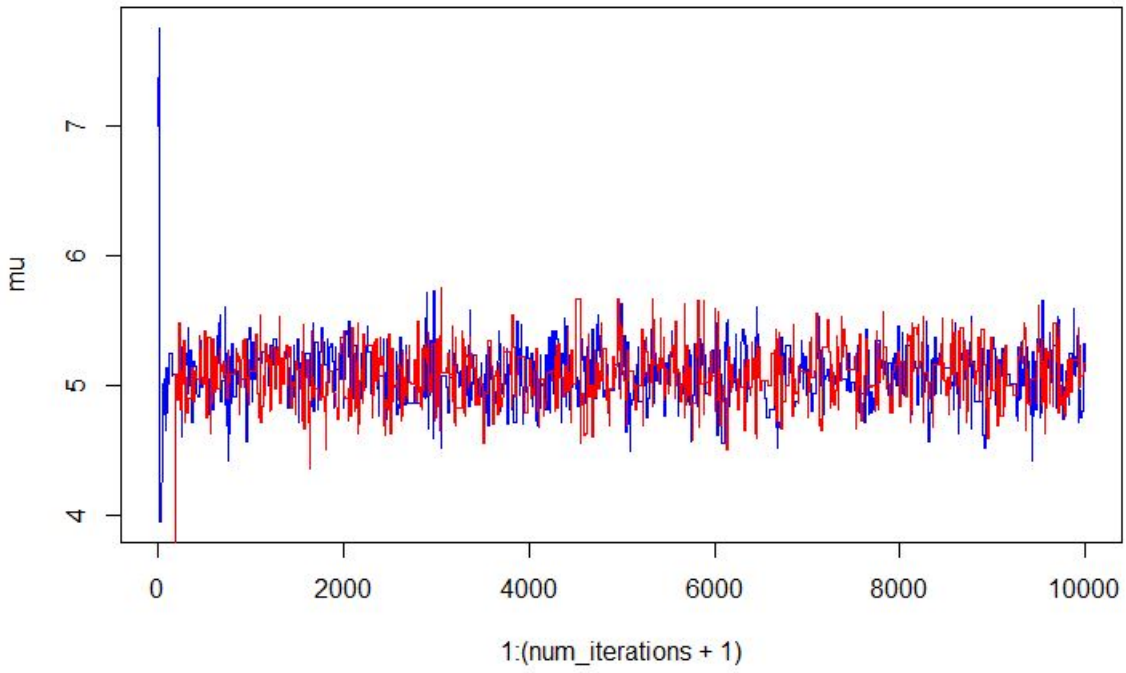
2.1 Part A

The code to run the MH algorithm is as follows:

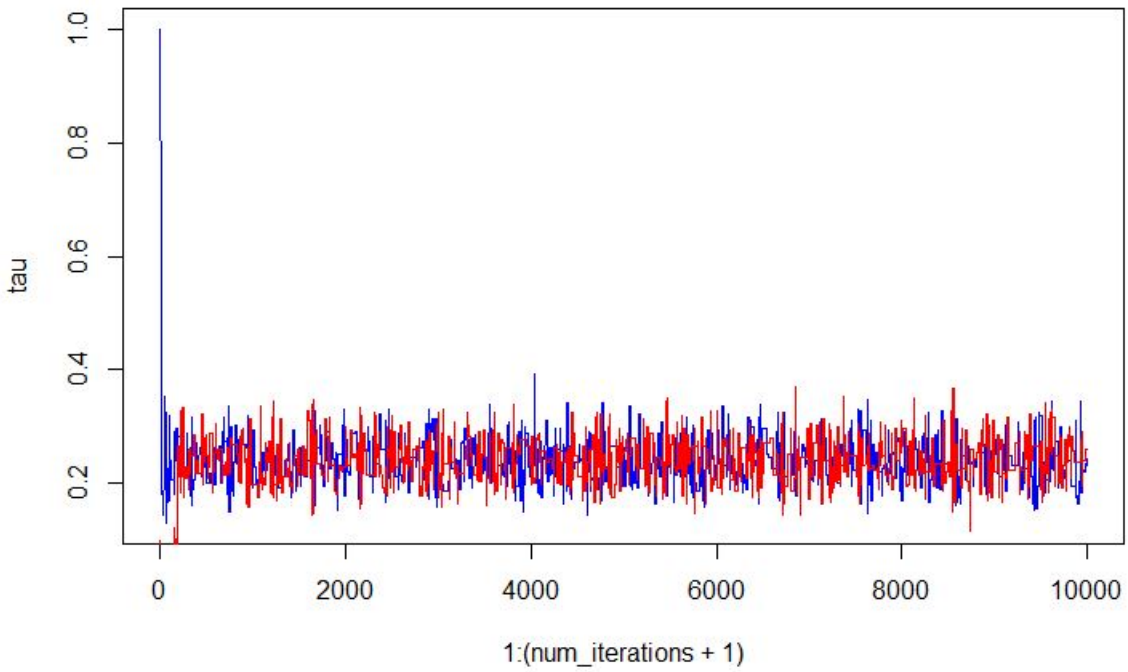
```r
# Convert the data into a vector
data_list = c()
for(i in 1:100) {
  data_list = c(data_list, as.numeric(Assign4Data[i]))
}

# Determining the Likelihood Function
l.likelihood = function(parameters) {
  mu = parameters[1]
  tau = parameters[2]

  singlelikelihood = dnorm(data_list, mean=mu, sd = sqrt(1/tau), log=T)
  return(sum(singlelikelihood))
}

# Determining the Prior
prior = function(parameters) {
  return(log(1/parameters[2]))
}

# Determining the Posterior
posterior = function(parameters) {
  return(l.likelihood(parameters) + prior(parameters))
}

# Determining the Proposal Function
prop.f = function(parameters){
  tau.n = rgamma(1, 5*parameters[2], 5)
  mu.n = rnorm(1, parameters[1], sqrt(tau.n))
  return(c(mu.n, tau.n))
}

# The Metropolis-Hastings Algorithm
run_metropolis_MCMC <- function(startvalue, iterations){
  chain = array(dim = c(iterations+1, 2))
  chain[1,] = startvalue
  for (i in 1:iterations){
    proposal = prop.f(chain[i,])
    probability = exp(posterior(proposal) - posterior(chain[i,]))
    if (runif(1) < probability){
      chain[i+1,] = proposal
    }else{
      chain[i+1,] = chain[i,]
    }
  }
  return(chain)
}
```

```
#m = Number of iterations
m = 10000
startvalue = c(7, 1)
startvalue2 = c(0.5, 0.1)
chain = run_metropolis_MCMC(startvalue, m)
chain2 = run_metropolis_MCMC(startvalue2, m)


plot(1:(m+1), chain[,1], ylab = "mu", type='l', col='blue')
points(1:(m+1), chain2[,1], ylab = "mu", type='l', col='red')
plot(1:(m+1), chain[,2], ylab = "tau", type='l', col='blue')
points(1:(m+1), chain2[,2], ylab = "tau", type='l', col='red')
```

10000 iterations are performed with 2 chains. Note that chain[,1] refers to mu while chain[,2] refers to tau. The initial values are set the same as for the Gibbs sampling to see how they compare. Here, "chain" has initial values that are significantly higher than the actual values while "chain2" has initial values that are significantly lower. The graph for mu becomes:



The graph for tau becomes:



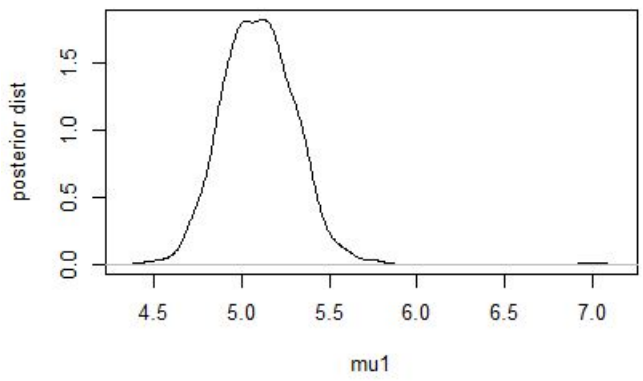
We can see that the MH algorithm samples also converge to approximately the actual values after a few samples for both chains. Therefore, it can also be inferred that these samples from two different chains are mixed well and behave similarly.

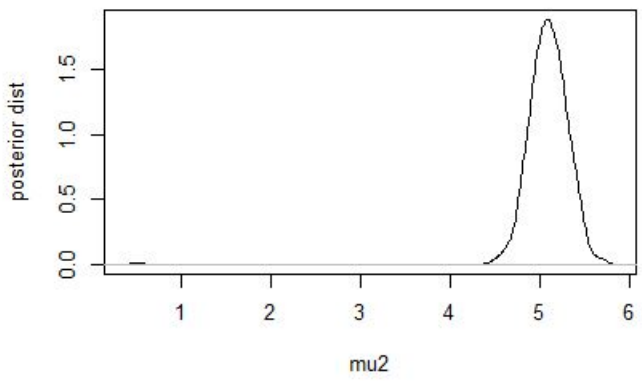
2.2 Part B

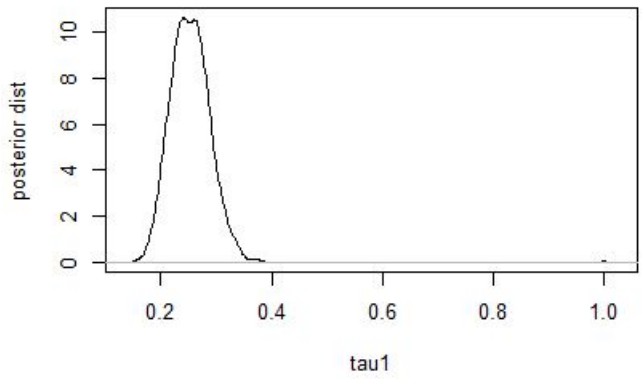
Similar to question 1C, the code is as follows:

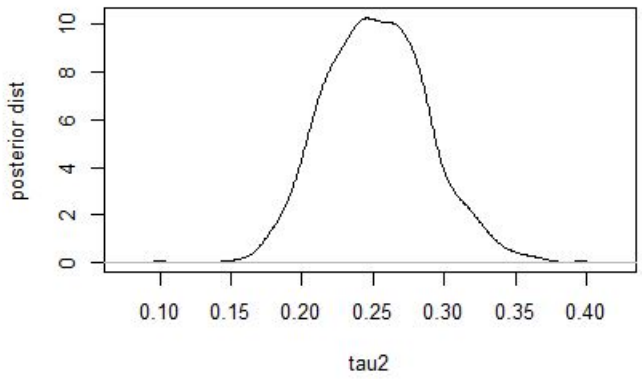
```
# Plots for marginal posterior distributions
par(mfrow=c(2,2))
plot(density(chain[,1]), ylab = "posterior dist", xlab = "mu1", main="")
plot(density(chain2[,1]), ylab = "posterior dist", xlab = "mu2", main="")
plot(density(chain[,2]), ylab = "posterior dist", xlab = "tau1", main="")
plot(density(chain2[,2]), ylab = "posterior dist", xlab = "tau2", main="")

# Marginal posterior means
mean(chain[,1])
mean(chain2[,1])
mean(chain[,2])
mean(chain2[,2])

#Find credible interval percentage for mu and tau
cdfmu = ecdf(chain[,1])
cdfmu(c(4.77, 5.41))

cdftau = ecdf(chain[,2])
cdftau(c(0.188,0.302))
```

Plotting the marginal posterior distributions yield the following graphs:









The marginal posterior means and credible intervals are:

```
> mean(chain[,1])
[1] 5.083556
> mean(chain2[,1])
[1] 5.051152
> mean(chain[,2])
[1] 0.2436084
> mean(chain2[,2])
[1] 0.2367363
> cdfmu(c(4.77, 5.41))
[1] 0.05219478 0.94860514
> cdftau(c(0.188,0.302))
[1] 0.05129487 0.95160484
```

This shows that the 95% credible interval for mu is approximately (4.77, 5.41) and the 95% credible interval for tau is approximately (0.188, 0.302).