MAST30025 Assingment 4

Kim Seang CHY

998008

Question 1: Posterior inference using Gibbs Sampling

#Prequestion

```
X = scan(file="assignment4_x_2021.txt", what=double())
Y = scan(file="assignment4_y_2021.txt", what=double())

length(X)
```

```
## [1] 100
```

```
mean(X)
```

```
## [1] 3.196441
```

```
length(Y)
```

```
## [1] 150
```

```
mean(Y)
```

```
## [1] -1.979781
```

a. Deriving conditional distrabution. Let $X = (x_1, \ldots, x_{100})$, $Y = (y_1, \ldots, y_{150})$.

For matrix $\Sigma = \begin{bmatrix} \frac{3}{5} & -\frac{2}{5} \\ -\frac{3}{5} & \frac{3}{5} \end{bmatrix}$, this implies $\Sigma^{-1} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$ and $\det(\Sigma) = |\Sigma| = \frac{1}{5}$.

Using the above the joint density of the μ_1 and μ_2 is given by:

$$
\begin{aligned}
f(\mu_1, \mu_2) &= \frac{1}{2\pi\sqrt{5}} \exp\left(-\frac{(\mu^T \Sigma^{-1} \mu)}{2}\right) \\
&= \frac{1}{2\pi\sqrt{5}} \exp\left(-\frac{(3\mu_1^2 + 4\mu_1\mu_2 + 3\mu_2^3)}{2}\right)
\end{aligned}
$$

Thus the joint portability of (μ_1, μ_2, X, Y) is given by:

$$
\begin{aligned}
P(\mu_1, \mu_2, X, Y) &= P(\mu_1, \mu_2) \prod_{i=1}^{100} P(x_i|\mu_1) \prod_{j=1}^{150} P(y_j|\mu_2) \\
&\propto \exp\left(-\frac{(3\mu_1^2 + 4\mu_1\mu_2 + 3\mu_2^3)}{2}\right) \prod_{i=1}^{100} \exp\left(-\frac{(x_i - \mu_1)^2}{2}\right) \prod_{j=1}^{150} \exp\left(-\frac{(y_j - \mu_2)^2}{4}\right) \\
&\propto \exp\left[\frac{-1}{2}\left(3\mu_1^2 + 4\mu_1\mu_2 + 3\mu_2^2 + \sum_{i=1}^{100}(x_i - \mu_1)^2 + \frac{1}{2}\sum_{j=4}^{150}(y_i - \mu_2)^2\right)\right]
\end{aligned}
$$

Using the above, we get the conditional density for $(\mu_1|\mu_2, X, Y)$ to be:

$$
\begin{aligned}
P(\mu_1|\mu_2, X, Y) &\propto P(\mu_1, \mu_2, X, Y) \\
&\propto \exp\left[-\frac{1}{2}\left(3\mu_1^2 + 4\mu_1\mu_2 - 2\sum_{i=1}^{100} x_i\mu_1 + 100\mu_1^2\right)\right] \\
&= \exp\left[-\frac{1}{2}\left(103\mu_1^2 - 2\left(\sum_{i=1}^{100} x_i - 2\mu_2\right)\mu_1\right)\right] \\
&\propto \exp\left[-\frac{103}{2}\left(\mu_1 - \frac{1}{103}(\sum_{i=1}^{100} x_i - 2\mu_2)\right)^2\right]
\end{aligned}
$$

By inspection of the proportional form we can see that the above is a normal distribution with a mean of $\frac{\sum_{i=1}^{100} x_i - 2\mu_2}{103}$ and variance of $\frac{1}{103}$ or $\mu_1|\mu_2, X, Y \sim N\left(\frac{1}{103}\left[\sum_{i=1}^{100} x_i - 2\mu_2\right], \frac{1}{103}\right)$.

Similarly, the conditional $(\mu_2|\mu_1, X, Y)$ is given by:

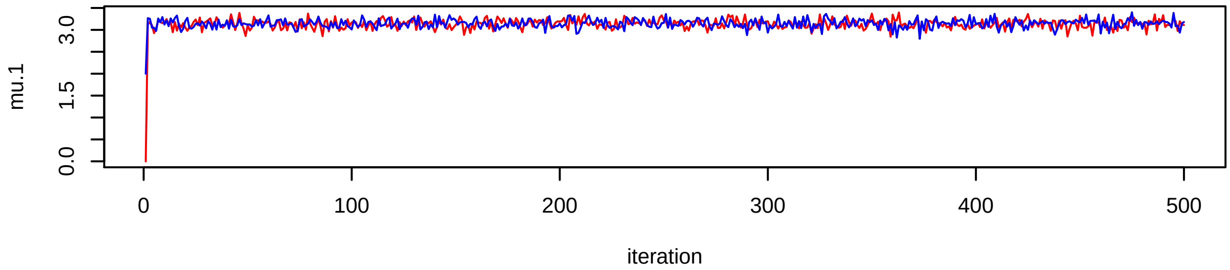
$$
\begin{aligned}
P(\mu_2|\mu_1, X, Y) &\propto P(\mu_1, \mu_2, X, Y) \\
&\propto \exp\left[-\frac{1}{2}\left(4\mu_1\mu_2 + 3\mu_2^3 - \sum_{j=1}^{150} y_j\mu_2 - 75\mu_2^2\right)\right] \\
&= \exp\left[-\frac{1}{2}\left(78\mu_2^2 - 2\left(\frac{1}{2}\sum_{j=1}^{150} y_j - 2\mu_1\right)\mu_2\right)\right] \\
&\propto \exp\left[-\frac{78}{2}\left(\mu_2 - \frac{1}{78}(\frac{1}{2}\sum_{j=1}^{150} y_j - 2\mu_1)\right)^2\right]
\end{aligned}
$$

By inspection the above is $\mu_1|\mu_2, X, Y$ is a normal distribution with mean $\frac{1}{156}\left(\sum_{j=1}^{150} y_j - 4\mu_1\right)$ and variance $\frac{1}{78}$ or $\mu_1|\mu_2, X, Y \sim N\left(\frac{1}{156}\left(\sum_{j=1}^{150} y_j - 4\mu_1\right), \frac{1}{78}\right)$

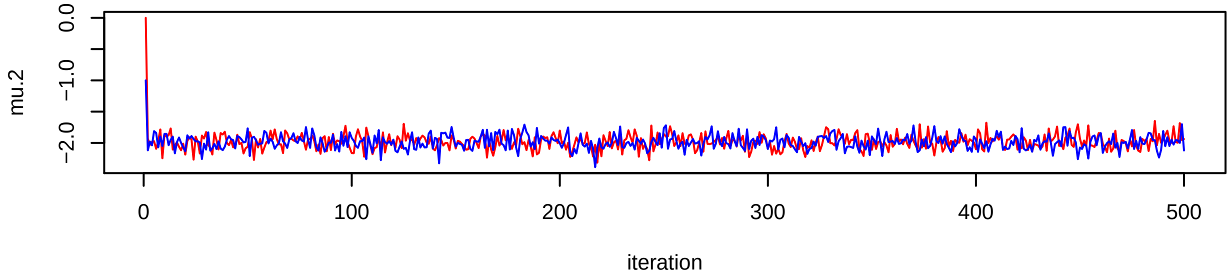
b.Run two Gibbs sampling chains with the following two initial values.

```
#Gibbs Sample function
GibbsS <- function(X,Y,mu.1,mu.2,num_it){

  #Gibbs Sample matrix
  GibbsSample <- matrix(nrow = num_it, ncol=2)
  GibbsSample[1,] <- c(mu.1,mu.2)

  #Simulating
  for (i in 2:num_it) {
    mu.1 <- rnorm(1,(1/103)*(sum(X)-2*mu.2),sqrt(1/103))
    mu.2 <- rnorm(1, (1/156)*(sum(Y)-4*mu.1),sqrt(1/78))
    GibbsSample[i,] <- c(mu.1,mu.2)
  }
  return(GibbsSample=GibbsSample)
}

#Computing Gibbs Sample use both set of intial value with seed of 30025
GibbsS1 = GibbsS(X,Y,0,0,500)
```

```
GibbsS2 = GibbsS(X,Y,2,-1,500)

#Plotting Trace Plot for MU1
par(mfrow=c(3,1), mar=c(4,4,1,1))
plot(1:500, GibbsS1[,1], type="l", col="red", ylim = c(0,max(GibbsS1[,1],GibbsS2[,1])), xlab = "iterati
points(1:500, GibbsS2[,1], type="l", col="blue")

#Plotting Trace Plot for MU2
par(mfrow=c(3,1), mar=c(4,4,1,1))
```




```
plot(1:500, GibbsS1[,2], type="l", col="red", ylim = c(min(GibbsS1[,2],GibbsS2[,2]),0), xlab = "iterati
points(1:500, GibbsS2[,2], type="l", col="blue")
```



c. Plotting the empirical posterior and empirical mean and CI

```
#Checking if burning 100 is enough for MU1
burnIn = 100
#Combing the two chain after burn in and compute standard deviation
Combined = c(GibbsS1[-(1:burnIn),1],GibbsS2[-(1:burnIn),1])
B = sd(Combined)
#Computing average standard deviation of chain 1 + chain 2 after burn in
W = mean(sd(GibbsS1[-(1:burnIn),1]),sd(GibbsS2[-(1:burnIn),1]))
#Computing BGR diagnostic
R=B/W
R
```

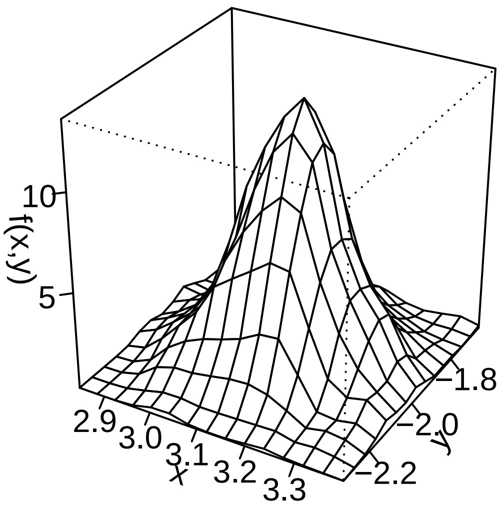
```
## [1] 1.010934
```

```
#Checking if burning 100 is enough for MU2
#Combing the two chain after burn in and compute standard deviation
Combined = c(GibbsS1[-(1:burnIn),2],GibbsS2[-(1:burnIn),2])
B = sd(Combined)
#Computing average standard deviation of chain 1 + chain 2 after burn in
W = mean(sd(GibbsS1[-(1:burnIn),2]),sd(GibbsS2[-(1:burnIn),2]))
#Computing BGR diagnostic
R=B/W
R
```

```
## [1] 0.9898576
```

Since, R is less than 1.05 and is approximately 1 we can infer that both mu_1 and μ_2 have converges.Hence, we will burn the first 100 value.

```
#Plotting the empirical density distribution
library(MASS)
chain_density <- kde2d(GibbsS1[-(1:burnIn),1], GibbsS1[-(1:burnIn),2], n = 15)
persp(chain_density, phi = 30, theta = 30, d = 5,
      xlab = "x", ylab = "y", zlab = "f(x,y)",
      ticktype = "detailed")
```




```
#Computing mean and Credible interval for posterior mu
mu1.mean = mean(GibbsS1[-(1:burnIn),1])
sd.bar = sd(GibbsS1[-(1:burnIn),1])
MU1.CI <-  mu1.mean+c(-1.64*sd.bar,1.64*sd.bar)
#Mean for MU1 Sample
mu1.mean
```

```
## [1] 3.139354
```

```
#MU1 90% Credible Interval
MU1.CI
```

```
## [1] 2.977479 3.301229
```

```
#Computing mean and Credible interval for posterior mu
mu2.mean = mean(GibbsS1[-(1:burnIn),2])
sd.bar = sd(GibbsS2[-(1:burnIn),1])
MU2.CI <-  mu2.mean+c(-1.64*sd.bar,1.64*sd.bar)
#Mean for MU1 Sample
mu2.mean
```

```
## [1] -1.98289
```

```
#MU1 90% Credible Interval
MU2.CI
```

```
## [1] -2.148257 -1.817523
```

Question 2: Posterior inference Using MH Algorithm

a. Write a code for MH algorithm.

$$
\begin{aligned}
P(\mu_1,\mu_2|X,Y) &\propto P(\mu_1,\mu_2,X,Y) \\
&\propto \exp\left(-\frac{1}{2}\left[3\mu_1^2+4\mu_1\mu_2+3\mu_2^2+\sum_{i=1}^{100}(x_i-\mu_1)^2+\frac{1}{2}\sum_{j=1}^{150}(y_i-\mu_2)^2\right]\right) \\
&\propto \exp\left(3\mu_1^2+4\mu_1\mu_2+3\mu_2^2-2\sum_{i=1}^{100}x_i\mu_1+100\mu_1-\sum_{j=1}^{150}y_j\mu_2+75\mu_2\right) \\
&= \exp\left[-\frac{1}{2}(103\mu_1^2+4\mu_1\mu_2+78\mu_2^2-2\sum_{i=1}^{100}x_i\mu_1-\sum_{j=1}^{150}y_j\mu_2)\right]
\end{aligned}
$$

We can ignore the proposal distribution in acceptance/rejection state as the proposal distribution or $q(.|.)$ is symmetric.

```
#Defining the Approximated Posterior Distribution
log.posterier <- function(X,Y,mu1,mu2){
  n.prob <- -0.5*(103*mu1^2+4*mu2*mu1-2*sum(X)*mu1+78*mu2^2-sum(Y)*mu2)
  return(n.prob)
}

#Defining MH Algorithm
MH_Algo <- function(X,Y,mu.1,mu.2,num.it){

  #Creating a matrix to store mu1, mu2 new value in the chain
  Chain.sample = matrix(nrow = num.it+2,ncol = 2)
  Chain.sample[1,] = c(mu.1,mu.2)

  #Setting the accepted value to 0
  accepted <- 0

  #Running MH-algorithm with num.it of interation
  for (i in 1:num.it){
    #Generating proposal for mu1 and mu2
    p.mu1 <- rnorm(1, mean = Chain.sample[i,1],sd=0.1)
    p.mu2 <- rnorm(1, mean = Chain.sample[i,2],sd=0.1)

    #Checking whether to accept or reject the proposed mu1 and mu2
    prob <- exp(log.posterier(X,Y,p.mu1,p.mu2)-log.posterier(X,Y,Chain.sample[i,1],Chain.sample[i,2]))
    if (runif(1) < prob) {
      Chain.sample[i+1,] = c(p.mu1,p.mu2)
      accepted <- accepted+1
    } else{
      Chain.sample[i+1,] <- Chain.sample[i,]
    }
  }
  Chain.sample[(num.it+2),]= accepted/num.it
  return(Chain.sample)
}
```

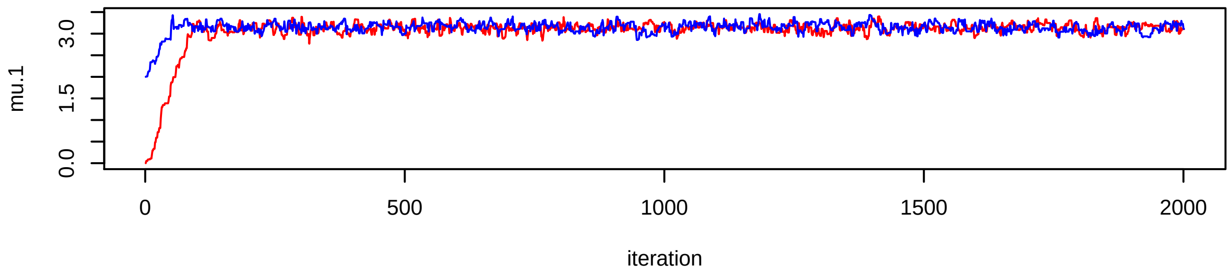
```r
num.it = 2000

#Running the MH Algorithm with two set of initial value
Sample.1 = MH_Algo(X,Y,0,0,num.it)
MH_A1 <- Sample.1[1:(num.it+1),]
Accepted_Rate1 <- Sample.1[(num.it+2),1]
Accepted_Rate1
```

```
## [1] 0.5595
```

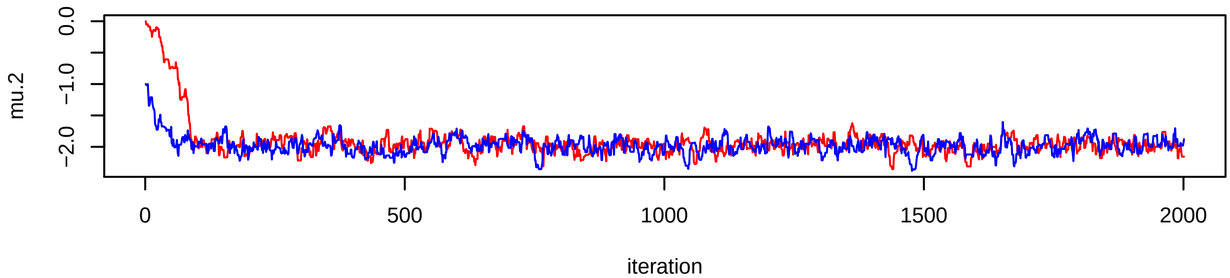
```r
Sample.2 = MH_Algo(X,Y,2,-1,num.it)
MH_A2 = Sample.2[1:(num.it+1),]
Accepted_Rate2 <- Sample.2[(num.it+2),1]
Accepted_Rate2
```

```
## [1] 0.585
```

```r
#Graphing the trace plot of MU1 and MU2
par(mfrow=c(3,1), mar=c(4,4,1,1))
plot(1:(num.it+1), MH_A1[,1], type="l", col="red", ylim = c(0,max(MH_A1[,1],MH_A2[,1])), xlab = "iterati
points(1:(num.it+1), MH_A2[,1], type="l", col="blue")

par(mfrow=c(3,1), mar=c(4,4,1,1))
```




```r
plot(1:(num.it+1), MH_A1[,2], type="l", col="red", ylim = c(min(MH_A1[,2],MH_A2[,2]),0), xlab = "iterati
points(1:(num.it+1), MH_A2[,2], type="l", col="blue")
```



b. Plotting the empirical posterior and empirical mean and CI

```r
#Checking if burning 500 is enough for MU1 using the same method as Q1
burnIn = 300
Combined = c(MH_A1[-(1:burnIn),1],MH_A2[-(1:burnIn),1])
B = sd(Combined)
W = mean(sd(MH_A1[-(1:burnIn),1]),sd(MH_A2[-(1:burnIn),1]))
R=B/W
R
```

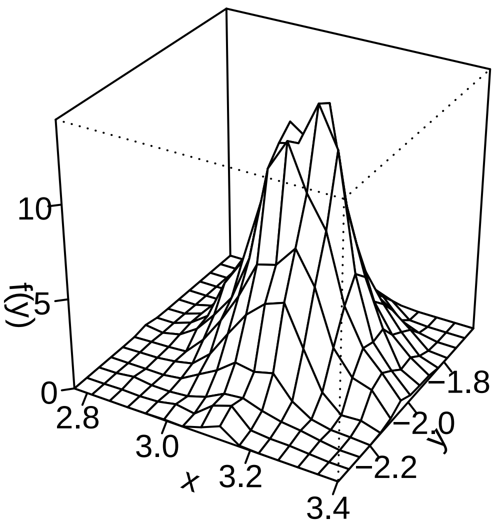
```
## [1] 1.031919
```

```r
#Checking if burning 500 is enough for MU2 using the same method as Q1
Combined = c(MH_A1[-(1:burnIn),2],MH_A2[-(1:burnIn),2])
B = sd(Combined)
W = mean(sd(MH_A1[-(1:burnIn),2]),sd(MH_A2[-(1:burnIn),2]))
R=B/W
R
```

```
## [1] 1.025065
```

Since, R is less than 1.05 and is approximately 1 we can infer that both mu_1 and μ_2 have converges.Hence, we will burn the first 300 value.

```r
#Plotting the empirical density distribution
library(MASS)
chain_density <- kde2d(MH_A1[-(1:burnIn),1], MH_A1[-(1:burnIn),2], n = 15)
persp(chain_density, phi = 30, theta = 30, d = 5,
      xlab = "x", ylab = "y", zlab = "f(y)",
      ticktype = "detailed")
```




```r
#Computing mean and Credible interval for posterior mu
mu1.mean = mean(MH_A1[-(1:burnIn),1])
sd.bar = sd(MH_A1[-(1:burnIn),1])
MU1.CI <-  mu1.mean+c(-1.64*sd.bar,1.64*sd.bar)
#Mean for MU1 Sample
mu1.mean
```

```
## [1] 3.134702
```

```r
#MU1 90% Credible Interval
MU1.CI
```

```
## [1] 2.977531 3.291873
```

```r
#Computing mean and Credible interval for posterior mu2
mu2.mean = mean(MH_A1[-(1:burnIn),2])
sd.bar = sd(MH_A1[-(1:burnIn),2])
MU2.CI <-  mu2.mean+c(-1.64*sd.bar,1.64*sd.bar)
#Mean for mu2 Sample
mu2.mean
```

```
## [1] -1.975512
```

```r
#mu2 90% Credible Interval
MU2.CI
```

```
## [1] -2.165576 -1.785447
```

Question 3: Posterior Inference using Variational Inference

a. Derive $q^*_{\mu_1}(\mu_1)$ and $q^*_{\mu_2}(\mu_2)$.

From previous question we know:

$$\begin{aligned} P(\mu_1,\mu_2,X,Y) &\propto P(\mu_1,\mu_2,X,Y) \\ &\propto \exp\left[-\frac{1}{2}\left(3\mu_1^2+4\mu_1\mu_2+\sum_{i=1}^{100}(x_i-\mu_1)^2+\frac{1}{2}\sum_{j=1}^{150}(y_j-\mu_2)^2\right)\right] \end{aligned}$$

Thus:

$$\begin{aligned} q^*_{\mu_1}(\mu_1) &\propto \exp\left[\mathbb{E}_{\mu_2}\left(\log(P(\mu_1,\mu_2,X,Y))\right)\right] \\ \implies \log(q^*_{\mu_1}(\mu_1)) &\propto \mathbb{E}_{\mu_2}\left[-\frac{1}{2}\left(3\mu_1^2+4\mu_1\mu_2+\sum_{i=1}^{100}(x_i-\mu_1)^2+\frac{1}{2}\sum_{j=1}^{150}(y_j-\mu_2)^2\right)\right] \\ &\propto -\frac{1}{2}\left(103\mu_1^2+4\mu_1\mathbb{E}_{\mu_2}\mu_2-2\sum_{i=1}^{100}x_i\mu_1\right) \\ \implies q_{\mu_1}(\mu_1) &\propto \exp\left[-\frac{103}{2}\left(\mu_1-\frac{1}{103}(\sum_{i=1}^{100}x_i-2\mathbb{E}_{\mu_2}\mu_2)\right)^2\right] \end{aligned}$$

From the above $q^*_{\mu_1}(\mu_1)$ has a pdf of $N\left(\mu_1^{(*)},\sigma_1^{2(*)}\right)$ where $\mu_1^{(*)}=\frac{1}{103}\left(\sum_{i=1}^{100}x_i-2\mathbb{E}_{\mu_2}(\mu_2)\right)$ and $\sigma_1^{2(*)}=\frac{1}{103}$.

Similarly,

$$\begin{aligned} \log(q^*_{\mu_2}(\mu_2)) &\propto \mathbb{E}_{\mu_1}\left(\log(P(\mu_1,\mu_2,X,Y))\right) \\ \implies q^*_{\mu_2}(\mu_2) &\propto \exp\left[-\frac{78}{2}\left(\mu_2-\frac{1}{156}(\sum_{j=1}^{150}y_j-4\mathbb{E}_{\mu_1}(\mu_1))\right)^2\right] \end{aligned}$$

From the above $q^*_{\mu_2}(\mu_2)$ has a pdf given by $N\left(\mu_2^{(*)},\sigma_2^{2(*)}\right)$ where $\mu_2^{(*)}=\frac{1}{156}(\sum_{j=1}^{150}y_j-4\mathbb{E}_{\mu_1}(\mu_1))$ and $\sigma_2^{2(*)}=\frac{1}{78}$

Using the known distribution we get $\mathbb{E}_{\mu_2}(\mu_2)=\mu_2^{(*)}$ and $\mathbb{E}_{\mu_1}(\mu_1)=\mu_1^{(*)}$

b.Derive ELBO up to a constant

The ELBO of $q^*_{\mu_1}(\mu_1)$ and $q^*_{\mu_2}(\mu_2)$ is given by:

$$\mathrm{ELBO}(q^*_{\mu_1}(\mu_1)q^*_{\mu_1}(\mu_1))=\mathbb{E}_{\mu_1\mu_2}\left(\log(P(\mu_1,\mu_2,X,Y))-\log[q^*_{\mu_1}(\mu_1)q^*_{\mu_1}(\mu_1)]\right)$$

which

$$
\begin{aligned}
\mathbb{E}_{\mu_1\mu_2}(\log[P(\mu_1,\mu_2,X,Y)]) &\propto \mathbb{E}_{\mu_1\mu_2}\left(-\frac{1}{2}\left[3\mu_1^2+4\mu_1\mu_2+3\mu_2^2+\sum_{i=1}^{100}(x_i-\mu_1)^2+\sum_{j=1}^{150}(y_j-\mu_2)^2\right]\right)\\
&= -\frac{1}{2}\left(3\mathbb{E}_{\mu_1}(\mu_1^2)+4\mathbb{E}_{\mu_1}(\mu_1)\mathbb{E}_{\mu_2}(\mu_2)+3\mathbb{E}_{\mu_2}(\mu_2^2)+\sum_{i=1}^{100}\mathbb{E}_{\mu_1}(x_i-\mu_1)^2+\frac{1}{2}\sum_{j=1}^{150}\mathbb{E}_{\mu_2}(y_i-\mu_2)^2\right)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbb{E}_{\mu_1}(\mu_1^2) &= \sigma_1^{2(*)}+\mu_1^{2(*)}\\
\mathbb{E}_{\mu_1}[(x_i-\mu_1)^2] &= \mathbb{E}_{\mu_1}(\mu_1^2)-2x_i\mathbb{E}_{\mu_1}(\mu_1)+x_i^2\\
&= \sigma_1^{2(*)}+\mu_1^{2(*)}-2x_i\mu_1^{2(*)}+x_i^2\\
\mathbb{E}_{\mu_2^2}(\mu_2) &= \sigma_2^{2(*)}+\mu_2^{2(*)}\\
\mathbb{E}_{\mu_2}[(y_j-\mu_2)^2] &= \mathbb{E}_{\mu_2}(\mu_2^2)-2y_j\mathbb{E}_{\mu_2}(\mu_2)+y_j^2\\
&= \sigma_2^{2(*)}+\mu_2^{2(*)}-2y_j\mu_2^{2(*)}+y_j^2
\end{aligned}
$$

And

$$
\begin{aligned}
\mathbb{E}_{\mu_1\mu_2}(\log[q^*_{mu_1}(\mu_1)]) &\propto \mathbb{E}_{\mu_1\mu_2}\left[-\frac{\log(\sigma_1^{2(*)})}{2}-\frac{(\mu_1-\mu_1^{(*)})^2}{2}\right]\\
&\propto -\frac{\log(\sigma_1^{2(*)})}{2}-\frac{\mathbb{E}_{\mu_1}((\mu_1-\mu_1^{(*)})^2)}{2}\\
&\propto -\frac{\log(\sigma_1^{2(*)})}{2}
\end{aligned}
$$

Similarly

$$
\mathbb{E}_{\mu_1\mu_2}(\log[q^*_{mu_2}(\mu_2)]) \propto -\frac{\log(\sigma_2^{2(*)})}{2}
$$

Thus,

$\text{ELBO}(q^*_{\mu_1}(\mu_1)q^*_{\mu_1}(\mu_1))$

$$
= -\frac{1}{2}\left(3\mathbb{E}_{\mu_1}(\mu_1^2)+4\mathbb{E}_{\mu_1}(\mu_1)\mathbb{E}_{\mu_2}(\mu_2)+3\mathbb{E}_{\mu_2}(\mu_2^2)+\sum_{i=1}^{100}\mathbb{E}_{\mu_1}(x_i-\mu_1)^2+\frac{1}{2}\sum_{j=1}^{150}\mathbb{E}_{\mu_2}(y_i-\mu_2)^2\right)+\frac{\log(\sigma_1^{2(*)})}{2}+\frac{\log(\sigma_2^{2(*)})}{2}
$$

where $\mathbb{E}_{\mu_1}(\mu_1), \mathbb{E}_{\mu_2}(\mu_2), \mathbb{E}_{\mu_1}(\mu_1^2)\mathbb{E}_{\mu_2}(\mu_2^2), \mathbb{E}_{\mu_1}[(x_i-\mu_1)^2], \mathbb{E}_{\mu_2}[(y_j-\mu_2)^2]$ are defined above.

c. Implementing the CAVI algorithm and obtain $q^*_{\mu_1}(\mu 1)$ and $q^*_{\mu_2}(\mu 2)$ which minimise KL divergence.

```
#Creating CAVI Alogrithm
CAVI_A <- function(X,Y,mu10,mu20,num.it=100, epsilon=1e-5){

  #Redefine all the initial value and Sample used
  mu1.vi = mu10
  mu2.vi = mu20
```

```
sigma1.star2 = 1/103
sigma2.star2 = 1/78

#Storing ELBO, mu1 and mu2 value for each iteration
elbo = c()
mu1.list = c()
mu2.list = c()

#Computing ELBO Value
Elogq.mu1 = -log(sigma1.star2)/2
Elogq.mu2 = -log(sigma2.star2)/2
Emu1.2 = sigma1.star2+mu1.vi^2
Emu2.2 = sigma2.star2+mu2.vi^2
#Let A=sum(E(xi-mu1)^2) and B=sum(E(yj-mu2)^2) and define below
A=sum(Emu1.2-2*X*mu1.vi+X*X)
B=sum(Emu2.2-2*Y*mu2.vi+Y*Y)

Elogp.mu1.mu2 = -0.5*(3*Emu1.2+4*mu1.vi*mu2.vi+3*Emu2.2+A+0.5*B)

#Storing ELBO, mu1, mu2 value
elbo= c(elbo ,Elogp.mu1.mu2-Elogq.mu1-Elogq.mu2)
mu1.list = c(mu1.list, mu1.vi)
mu2.list= c(mu2.list,mu2.vi)

#Set initial iteration value to 1
curr.int = 1

#Set change in the ELBO as 1
delta.elbo = 1

#Using a loop where if change ELBO is less than epsilon or the max number of
#iteration reach it will stop CAVI algorithm
while ((delta.elbo > epsilon) & (curr.int <= num.it)){

  #Update mu1 and mu2
  mu1.vi = (1/103)*(sum(X)-2*mu2.vi)
  mu2.vi = (1/156)*(sum(Y)-4*mu1.vi)

  #Computing ELBO using the new mu1 and mu2
  Emu1.2 = sigma1.star2+mu1.vi^2
  Emu2.2 = sigma2.star2+mu2.vi^2

  #Let A=sum(E(xi-mu1)^2) and B=sum(E(yj-mu2)^2) and define below
  A=sum(Emu1.2-2*X*mu1.vi+X*X)
  B=sum(Emu2.2-2*Y*mu2.vi+Y*Y)
  Elogp.mu1.mu2 = -0.5*(3*Emu1.2+4*mu1.vi*mu2.vi+3*Emu2.2+A+0.5*B)

  #Storing the new value
  elbo= c(elbo,Elogp.mu1.mu2-Elogq.mu1-Elogq.mu2)
  mu1.list = c(mu1.list, mu1.vi)
  mu2.list= c(mu2.list,mu2.vi)

  #Computing Change in ELBO
```

```r
    delta.elbo = elbo[length(elbo)]-elbo[length(elbo)-1]

    #increase number of iteration
    curr.int = curr.int+1
  }
  return(list(elbo=elbo,mu1.list=mu1.list,mu2.list=mu2.list))
}
```

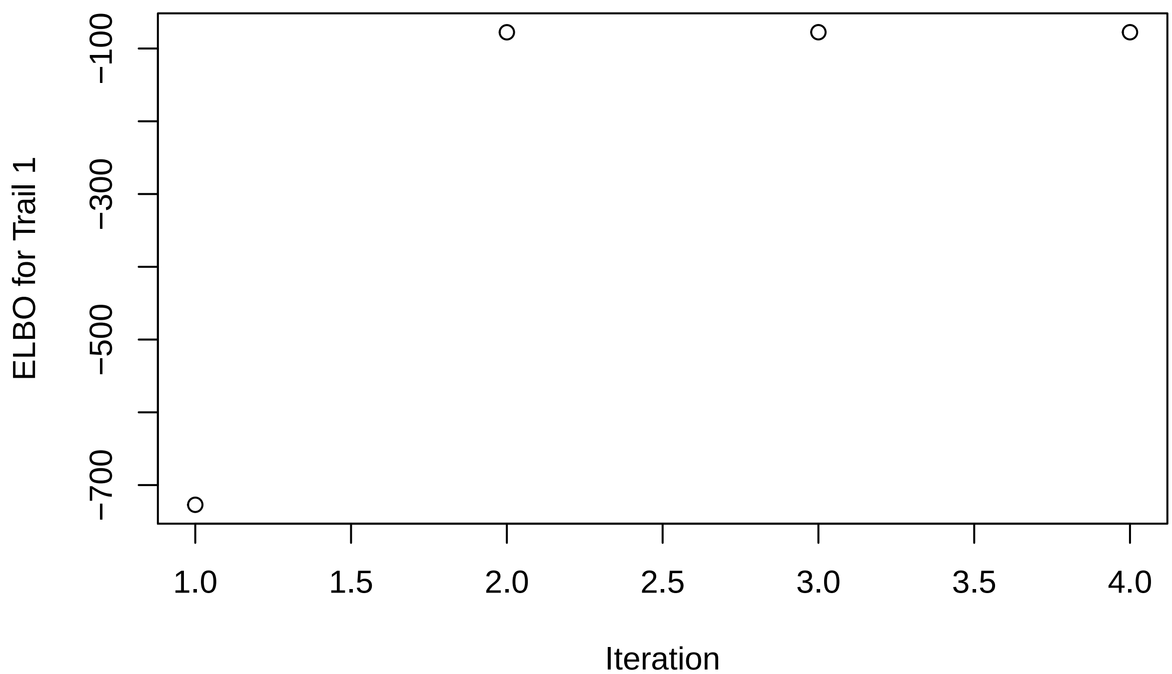
Running CAVI algorithm with two initial value for $\mu1$ and $\mu2$. We will use same two initial value as question 1 and 2.

```r
#Running CAVI algorithm Mu1=0, Mu2=0
CAVI_T1 = CAVI_A(X,Y,0,0)

#Running CAVI algorithm Mu1=2, Mu2=-1
CAVI_T2 = CAVI_A(X,Y,2,-1)
```

```r
#Printing ELBO Value for Trail 1
CAVI_T1$elbo
```

```
## [1] -727.08254  -77.70894  -77.63253  -77.63253
```

```r
#Plotting ELBO value
plot(CAVI_T1$elbo, ylab='ELBO for Trail 1', xlab='Iteration')
```



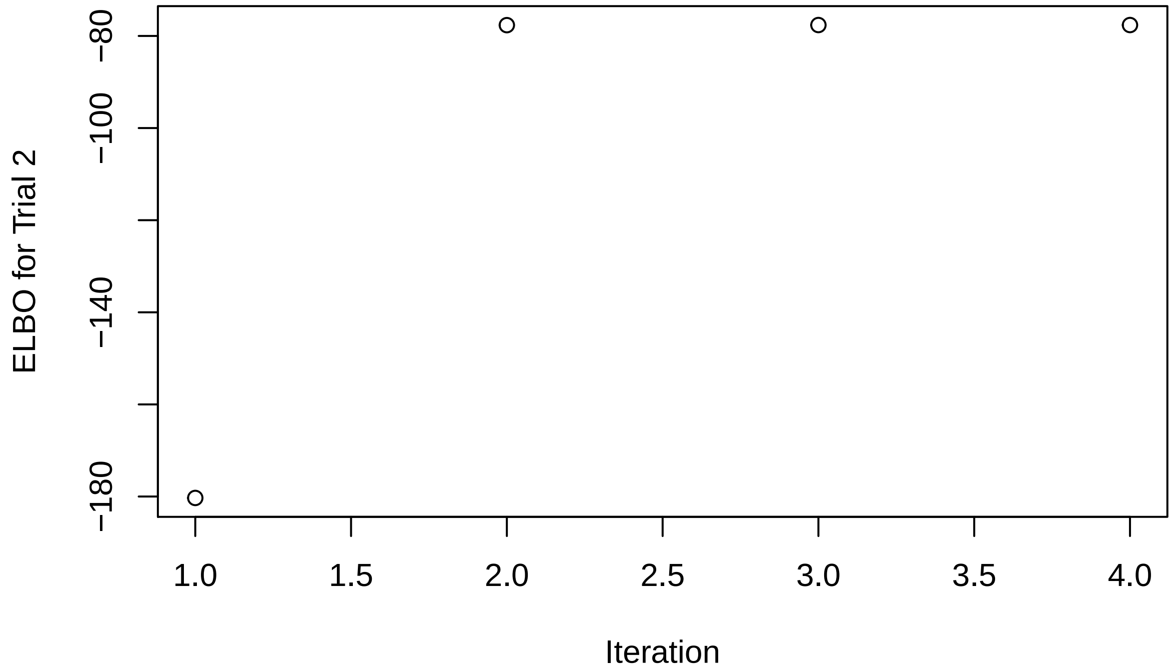

```r
#Pasting the final value for mu1 and mu2
print(paste("mu1 and mu2 = (",
            round(CAVI_T1$mu1.list[length(CAVI_T1$mu1.list)],2),",",
            round(CAVI_T1$mu2.list[length(CAVI_T1$mu2.list)],2), ")", sep=""))
```

```
## [1] "mu1 and mu2 = (3.14,-1.98)"
```

```r
#Printing ELBO Value for Trail 1
CAVI_T2$elbo
```

```
## [1] -180.31088  -77.65133  -77.63253  -77.63253
```

```r
#Plotting ELBO value
plot(CAVI_T2$elbo, ylab='ELBO for Trial 2', xlab='Iteration')
```




```r
#Pasting the final value for mu1 and mu2
print(paste("mu1 and mu2 = (",
            round(CAVI_T2$mu1.list[length(CAVI_T2$mu1.list)],2),",",
            round(CAVI_T2$mu2.list[length(CAVI_T2$mu2.list)],2), ")", sep=""))
```

```
## [1] "mu1 and mu2 = (3.14,-1.98)"
```

Using the CAVI algorithm we get $q^*_{\mu_1} \sim N\left(3.14, \frac{1}{103}\right)$ and $q^*_{\mu_2} \sim N\left(-1.98, \frac{1}{78}\right)$.