MAST30027 Assignment 4

Name: Lang (Ron) Chen

Student ID: 1181506

Subject: MAST30027

Assignment number: 4

Tutorial time and tutor: Tues 11:00 Yidi Deng

1. a.

$$\mathbb{P}(\mu, \sigma^2, \vec{x}) \propto \mathbb{P}(\vec{x} \mid \mu, \sigma^2)\, \mathbb{P}(\mu, \sigma^2)$$
$$\propto \prod_{i=1}^{100} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}} \right) \times \frac{1}{\sigma^2}$$

$$\propto \left(\frac{1}{\sigma}\right)^{100} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2} \frac{1}{\sigma^2}$$
$$\propto \left(\frac{1}{\sigma^{102}}\right) e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2}$$

$$\mathbb{P}(\mu \mid \sigma^2, \vec{x}) = \mathbb{P}(\mu, \sigma^2, \vec{x}) / \mathbb{P}(\sigma^2, \vec{x})$$
$$\propto \mathbb{P}(\mu, \sigma^2, \vec{x}) \quad \text{as denom d.n. contain } \mu$$
$$\propto e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2}$$
$$\propto e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i^2 - 2x_i\mu + \mu^2)}$$
$$\propto e^{-\frac{1}{2\sigma^2}(100\mu^2 - 2\times\sqrt{100}\times\mu\sum_{i=1}^{100}x_i \times \frac{1}{10})}$$
$$\propto e^{-\frac{1}{2\sigma^2}\left(10\mu - \frac{\sum_{i=1}^{100}x_i}{10}\right)^2}$$
$$\propto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\left(10\mu - \frac{\sum x_i}{10}\right)^2}$$
$$\Rightarrow 10\mu \mid \vec{x}, \sigma^2 \sim N\left(\frac{\sum_{i=1}^{100}x_i}{10}, \sigma^2\right)$$
$$\Rightarrow \mu \mid \vec{x}, \sigma^2 \sim N\left(\frac{\sum_{i=1}^{100}x_i}{100}, \frac{\sigma^2}{100}\right)$$

$$\mathbb{P}(\sigma^2 \mid \mu, \vec{x}) \propto \mathbb{P}(\mu, \sigma^2, \vec{x})$$
$$\propto \left(\frac{1}{\sigma^2}\right)^{51} e^{-\left(\sum_{i=1}^{100}(x_i-\mu)^2\right)\frac{1}{2\sigma^2}}$$

$$\sigma^2 \mid \mu, \vec{x} \sim \text{Inverse Gamma}\left(50, \frac{\sum_{i=1}^{100}(x_i-\mu)^2}{2}\right)$$

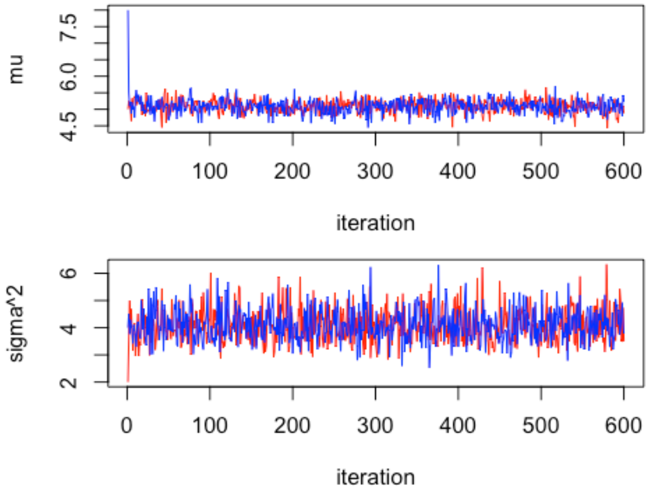
Setup

```
set.seed(30027)
setwd('/Users/tg.chenny/Desktop/1. University/1. Undergraduate/20. Modern Applied Statistics/Asmt/Asmt 4')

library(invgamma)

X = scan(file='assignment4.txt', what=double())
mean(X)

## [1] 5.089332

sqrt(var(X))

## [1] 1.998487
```

Q1

1b)

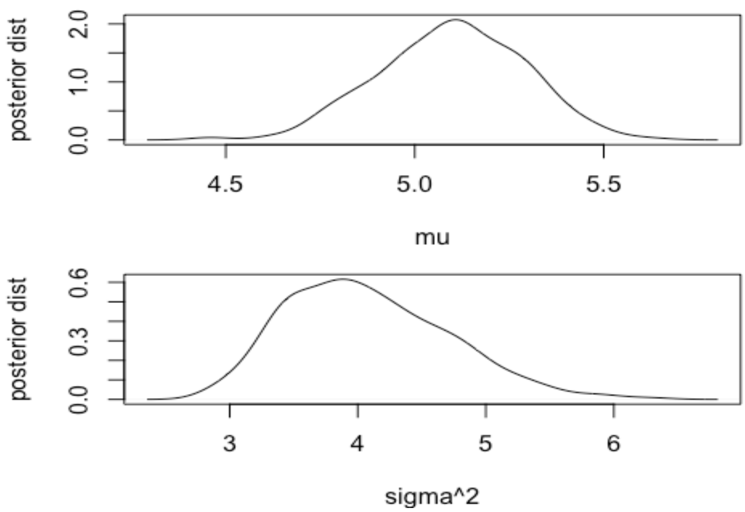
```
# gibbs function
gibbs.f2 = function(x, mu0, sigmasq0, N){
  mu.seq <- sigmasq.seq <- rep(-1, N)

  # set initial values
  mu.seq[1] <- mu0
  sigmasq.seq[1] <- sigmasq0

  # simulate new sample values
  for(j in 2:N){
    mu.seq[j] <- rnorm(1, mean(x), sqrt(sigmasq.seq[j-1]/100))
    sigmasq.seq[j] <- rinvgamma(1, 50, 0.5*sum((x-mu.seq[j])^2))
  }

  result = list(mu = mu.seq, sigmasq = sigmasq.seq)
  return(result)
}

# run simulations
N = 600
gibbsam1=gibbs.f2(X, 5, 2, N)
gibbsam2=gibbs.f2(X, 8, 4, N)


# run trace plots
par(mfrow=c(2,1), mar=c(4,4,1,1))
plot(1:N, gibbsam1$mu, type='l', col='red', ylim = c(min(gibbsam1$mu, gibbsam2$mu), max(gibbsam1$mu,
gibbsam2$mu)), xlab = 'iteration', ylab='mu')
points(1:N, gibbsam2$mu, type='l', col='blue')


plot(1:N, gibbsam1$sigmasq, type='l', col='red', ylim = c(min(gibbsam1$sigmasq, gibbsam2$sigmasq),
max(gibbsam1$sigmasq, gibbsam2$sigmasq)), xlab = 'iteration', ylab='sigma^2')
points(1:N, gibbsam2$sigmasq, type='l', col='blue')
```



Judging from the trace plot, the two chains starting at different initial values are mixing well.

1c)

```
# estimated distribution
## USE 50 BURNIN

par(mfrow=c(2,1), mar=c(4,4,1,1))
plot(density(gibbsam1$mu[-(1:50)]), ylab='posterior dist', xlab='mu', main='')
plot(density(gibbsam1$sigmasq[-(1:50)]), ylab='posterior dist', xlab='sigma^2', main='')
```




```
# get mean and 90% credible interval
print('MU')

## [1] "MU"

paste('mean: ', mean(gibbsam2$mu[-(1:50)]))

## [1] "mean:  5.07912864622427"

paste('90% CI:', quantile(gibbsam2$mu[-(1:50)], 0.05), quantile(gibbsam2$mu[-(1:50)], 0.95))

## [1] "90% CI: 4.7483665704439 5.40405779461706"
```

```r
print('')
## [1] ""
print('SIGMASQ')
## [1] "SIGMASQ"
paste('mean: ', mean(gibbsam2$sigmasq[-(1:50)]))
## [1] "mean:  4.10065869642716"
paste('90% CI:', quantile(gibbsam2$sigmasq[-(1:50)], 0.05), quantile(gibbsam2$sigmasq[-(1:50)], 0.95))
## [1] "90% CI: 3.21295925894037 5.13917091531868"
```

Using chain 2: the estimated posterior mean of mu is 5.07912; with 90% credible interval (4.7484, 5.4041); the estimated posterior mean of sigma^2 is 4.1007; with 90% credible interval (3.2130, 5.1392).

2. a. Theory only. Pls see below for R code

Get acceptance prob:

$$A = \min\left(1, \frac{\pi(\theta')\, q(\theta', \theta)}{\pi(\theta)\, q(\theta, \theta')}\right)$$

q:

$$\sigma^2_n \sim \text{gamma}(\text{shape} = 5\sigma^2_c, \text{rate} = 5)$$

$$\mu_n \sim N(\text{mean} = \mu_c, \text{Var} = \sigma^2_n)$$

$$q(\theta, \theta') = \text{dgamma} \times \text{dnorm}.$$

$$\log q(\theta, \theta') = \log(\text{dgamma}) + \log(\text{dnorm})$$

π:

$$\pi(\mu, \sigma^2) \propto \mathbb{P}(X \mid \mu, \sigma^2)\, \mathbb{P}(\mu, \sigma^2)$$

$$\propto \prod_{i=1}^{100} \mathbb{P}(x_i \mid \mu, \sigma^2)\, \frac{1}{\sigma^2}$$

$$\propto \frac{1}{\sigma^{102}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{100} (x_i - \mu)^2}$$

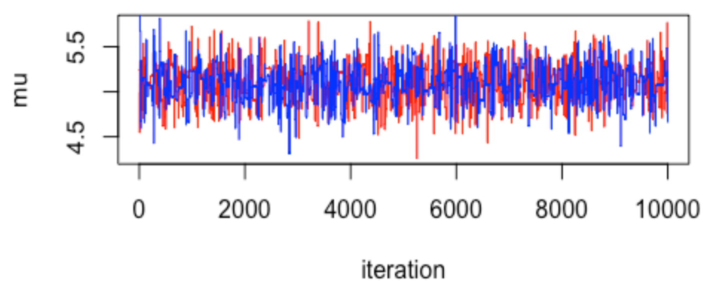
$$\log \pi(\mu, \sigma^2) = -51 \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{100} (x_i - \mu)^2$$

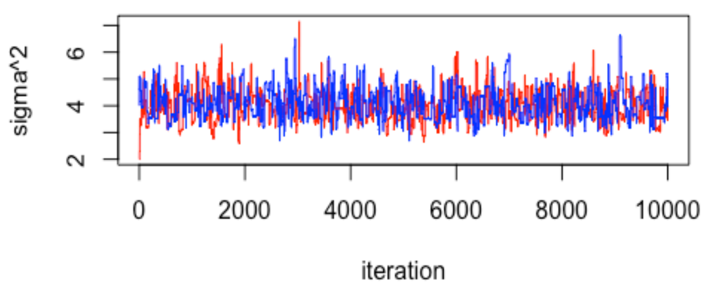
Q2

2a

```r
# MH algorithm function
run_metropolis_MCMC = function(startvalue, iterations){

  # initiate chain and set initial values
  chain = array(dim=c(iterations+1, 2))
  chain[1,] = startvalue

  for (i in 1:iterations){

    # get proposal
    proposal = proposalfunction(chain[i,])

    # compute acceptance probability
    probab = exp( log_posterior(proposal) + log_q(proposal, chain[i,]) - (log_posterior(chain[i,]) +
log_q(chain[i,], proposal)) )

    # decide whether to accept

    if (runif(1) < probab){

      chain[i+1,] = proposal

    } else {

      chain[i+1,] = chain[i,]

    }
  }

  return(chain)
}

# proposal function (derivation pls see previous)
proposalfunction = function(param){
  sigmasq = rgamma(1, 5*param[2], 5)
  return(c(rnorm(1, param[1], sqrt(sigmasq)), sigmasq))
}

# log posterior function (derivation please see previous)
log_posterior = function(param){
  return (-51*log(param[2])-(1/(2*param[2]))*sum((X-param[1])^2))
}

# log q function (derivation please see previous)
log_q = function(param, proposal){
  return(dnorm(proposal[1], mean=param[1], sd=sqrt(proposal[2]), log=T) + dgamma(proposal[2], shape =
5*param[2], rate = 5, log=T))
}

proposalfunction <- function(param){
   mu = param[1]
   sigma2 = param[2]
   new.sigma2 = rgamma(1, 5*sigma2, 5)
   new.mu = rnorm(1, mu, sqrt(new.sigma2))
   return(as.vector(c(new.mu, new.sigma2)))
 }
# log prob of proposal function
 proposal.prob <- function(old.param, new.param){
   mu = old.param[1]
   sigma2 = old.param[2]
   new.mu = new.param[1]
   new.sigma2 = new.param[2]
    return(dnorm(new.mu, mean = mu, sd = sqrt(sigma2), log = T)+
```

```
            dgamma(new.sigma2, shape = 5*sigma2, rate = 5, log=T))
}

# actually run the MH simulation
startvalue1 = c(5, 2)
chain1 = run_metropolis_MCMC(startvalue1, 10000)
startvalue2 = c(8, 4)
chain2 = run_metropolis_MCMC(startvalue2, 10000)

burnIn=5000
acceptance1=1-mean(duplicated(chain1[-(1:burnIn),]))
acceptance2=1-mean(duplicated(chain2[-(1:burnIn),]))

acceptance1

## [1] 0.089982

acceptance2

## [1] 0.0789842

# run trace plots
par(mfrow=c(2,1), mar=c(4,4,1,1))
plot(1:10001, chain1[,1], type='l', col='red', xlab = 'iteration', ylab='mu')
points(1:10001, chain2[,1], type='l', col='blue')


plot(1:10001, chain1[,2], type='l', col='red', xlab = 'iteration', ylab='sigma^2')
points(1:10001, chain2[,2], type='l', col='blue')
```

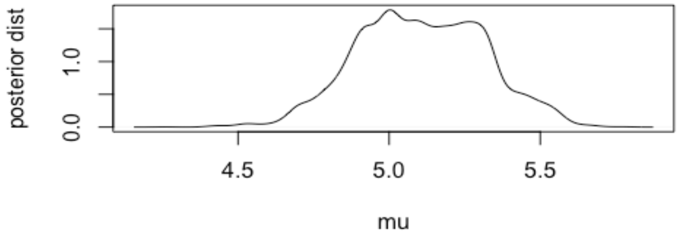


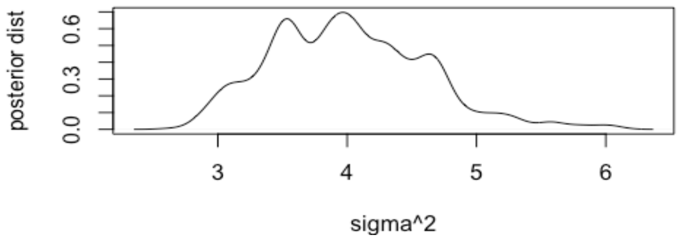


Judging from the trace plot, the two chains starting at different initial values are mixing well.

2b

```
# estimated distribution
par(mfrow=c(2,1), mar=c(4,4,1,1))
plot(density(chain1[-(1:5000),1]), ylab='posterior dist', xlab='mu', main='')
plot(density(chain1[-(1:5000),2]), ylab='posterior dist', xlab='sigma^2', main='')
```






```r
# get mean and 90% credible interval
print('MU')

## [1] "MU"

paste('mean: ', mean(chain1[-c(1:5000),1]))

## [1] "mean:  5.10476145548351"

paste('90% CI:', quantile(chain1[-c(1:5000),1], 0.05), quantile(chain1[-c(1:5000),1], 0.95))

## [1] "90% CI: 4.76604775616876 5.46044818448603"

print('')

## [1] ""

print('SIGMASQ')

## [1] "SIGMASQ"

paste('mean: ', mean(chain1[-c(1:5000),2]))

## [1] "mean:  4.01140153726327"

paste('90% CI:', quantile(chain1[-c(1:5000),2], 0.05), quantile(chain1[-c(1:5000),2], 0.95))

## [1] "90% CI: 3.07241992449456 5.09909700840007"
```

Using chain 1: the estimated posterior mean of mu is 5.1048; with 90% credible interval (4.76605, 5.4604); the estimated posterior mean of sigma^2 is 4.0114; with 90% credible interval (3.0724, 5.0991).

3.a. $x_i \sim N(\mu,\sigma^2)$ for $i=1,\dots,100$

$P(\mu,\sigma^2) = P(\mu|\sigma^2)\,P(\sigma^2)$

$\mu|\sigma^2 \sim N(\mu_0,\sigma^2) \quad \sigma^2 \sim \text{Inv Gam}(a_0,b_0)$

suppose $q(\mu,\sigma^2)=q(\mu)\,q(\sigma^2)$

$$
\begin{aligned}
\log q^*(\mu) &= E_{\sigma^2}[\log P(\mu,\sigma^2,x_1,\dots,x_n)] + \text{const}\\
&= E_{\sigma^2}\Big[\log \prod_{i=1}^{100} P(x_i|\mu,\sigma^2) + \log P(\mu|\sigma^2) + \log(\sigma^2)\Big] + \text{const}\\
&= E_{\sigma^2}\Big[\log\Big(\prod_{i=1}^{100}\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2}\Big) + \log\Big(\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma}\right)^2}\Big) + \log\Big(\frac{b_0^{a_0}}{\Gamma(a_0)}\sigma^{-2a_0-2}e^{-\frac{b_0}{\sigma^2}}\Big)\Big] + \text{const}\\
&= E_{\sigma^2}\Big[\cancel{\log\frac{100}{\sqrt{2\pi}\,\sigma}} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 + \cancel{\log\Big(\frac{1}{\sigma\sqrt{2\pi}}\Big)} - \frac{1}{2\sigma^2}(\mu-\mu_0)^2 + \cancel{\log\Big(\frac{b_0^{a_0}}{\Gamma(a_0)}\Big)} + \cancel{\log\sigma^{-2a_0-2}} - \cancel{\frac{b_0}{\sigma^2}}\Big] + \text{const}\\
&= E_{\sigma^2}\Big[-\frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 - \frac{1}{2\sigma^2}(\mu-\mu_0)^2\Big] + \text{const}\\
&= -\frac{E(\frac{1}{\sigma^2})}{2}\Big[\sum_{i=1}^{100}(x_i-\mu)^2 + (\mu-\mu_0)^2\Big] + \text{const}\\
&= -\frac{E(\frac{1}{\sigma^2})}{2}\Big[\sum_{i=1}^{100}(x_i-\mu)^2 + (\mu-\mu_0)^2\Big] + \text{const}\\
&= -\frac{E(\frac{1}{\sigma^2})}{2}\Big[\sum_{i=1}^{100}(\cancel{x_i^2}-2x_i\mu+\mu^2) + (\mu^2-2\mu\mu_0+\cancel{\mu_0^2})\Big] + \text{const}\\
&= -\frac{E(\frac{1}{\sigma^2})}{2}\Big[101\mu^2 - 2\Big(\sum_{i=1}^{100}x_i+\mu_0\Big)\mu\Big] + \text{const}
\end{aligned}
$$

We see $q^*(\mu)$ is pdf of $N(\mu^*,\sigma^{2*})$

$\mu = \dfrac{\sum_{i=1}^{100}x_i+\mu_0}{101}$; $\sigma^{2*} = \dfrac{1}{101\,E(\frac{1}{\sigma^2})}$

where $E(\frac{1}{\sigma^2}) = \dfrac{a^*}{b^*}$

$$
\begin{aligned}
\log q_{\sigma^2}(\sigma^2) &= E_\mu[\log P(\mu,\tau,x_1,\dots,x_n) + \log P(\sigma^2)] + \text{const}\\
&= E_\mu\Big[\log \prod_{i=1}^{100} P(x_i|\mu,\sigma^2) + \log P(\mu|\sigma^2) + \log(\sigma^2)\Big] + \text{const}\\
&= E_\mu\Big[\log\Big(\prod_{i=1}^{100}\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2}\Big) + \log\Big(\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma}\right)^2}\Big) + \log\Big(\frac{b_0^{a_0}}{\Gamma(a_0)}\sigma^{-2a_0-2}e^{-\frac{b_0}{\sigma^2}}\Big)\Big] + \text{const}\\
&= E_\mu\Big[\cancel{100\log\frac{1}{\sqrt{2\pi}}} + 50\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 + \log\frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2}(\mu-\mu_0)^2 + \cancel{\log\frac{b_0^{a_0}}{\Gamma(a_0)}}\\
&\qquad + (a_0+1)\log\frac{1}{\sigma^2} - \frac{b_0}{\sigma^2}\Big] + \text{const}\\
&= 50\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}E_\mu[(x_i-\mu)^2] + \frac{1}{2}\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}E_\mu[(\mu-\mu_0)^2] + (a_0+1)\log\frac{1}{\sigma^2} - \frac{b_0}{\sigma^2} + \text{const}\\
&= (50.5+(a_0+1))\log\frac{1}{\sigma^2} - \Big[\frac{1}{2}\sum_{i=1}^{100}E_\mu[(x_i-\mu)^2] + \frac{1}{2}E_\mu[(\mu-\mu_0)^2] + b_0\Big]\frac{1}{\sigma^2} + \text{const}\\
&= \log {\sigma^2}^{-(50.5+(a_0+1))} - \Big[\frac{1}{2}\sum_{i=1}^{100}E_\mu[(x_i-\mu)^2] + \frac{1}{2}E_\mu[(\mu-\mu_0)^2] + b_0\Big]\frac{1}{\sigma^2} + \text{const}
\end{aligned}
$$

$-a^*-1 = -(50.5+(a_0+1))$

$-a^* = -(50.5+(a_0+1))+1$

$a^* = 50.5+(a_0+1)-1$

We see $q^*(\sigma^2)$ is pdf of Inv-Gamma(a^*, b^*)

$a^* = 50.5 + (a_0+1) - 1$

$\quad = 50.5 + a_0$

$b^* = \frac{1}{2}\sum_{i=1}^{100}E_\mu[(x_i-\mu)^2] + \frac{1}{2}E_\mu[(\mu-\mu_0)^2] + b_0$

where $E_\mu[(x_i-\mu)^2] = E_\mu(\mu^2) - 2x_iE_\mu(\mu) + x_i^2$

$\quad = \sigma^{2*} + (\mu^*)^2 - 2x_i\mu^* + x_i^2$

$E_\mu[(\mu-\mu_0)^2] = \sigma^{2*} + \mu^{*2} - 2\mu_0\mu^* + \mu_0^2$

b. $\mathrm{ELBO}(q_\mu^*(\mu), q_{\sigma^2}^*(\sigma^2)) = \mathrm{ELBO}(\mu^*, \sigma^{2*}, a^*, b^*)$

$$= E_{\mu,\sigma^2}(\log P(\log P(x_1, \dots, x_n, \mu, \tau) - \log(q_\mu^*(\mu)\, q_{\sigma^2}^*(\sigma^2))]$$

$$= \underbrace{E_{\mu,\sigma^2}[(\log \prod_{i=1}^{100} P(x_i \mid \mu, \sigma^2) P(\mu \mid \sigma^2) P(\sigma^2)]}_{①} - \underbrace{E_{\mu,\sigma^2}[\log q_\mu^*(\mu)]}_{②} - \underbrace{E_{\mu,\sigma^2}[\log q_{\sigma^2}^*(\sigma^2)]}_{③}$$

② $E_{\mu,\sigma^2}[\log q_\mu^*(\mu)] = E_{\mu,\sigma^2}\left[-\frac{1}{2}\log \sigma^{2*} - \frac{(\mu-\mu^*)^2}{2\sigma^{2*}}\right] + \text{const}$

$$= -\frac{1}{2}\log \sigma^{2*} - \frac{E[(\mu-\mu^*)^2]}{2\sigma^{2*}} + \text{const}$$

$$= -\frac{1}{2}\log \sigma^{2*} - \frac{\cancel{\sigma^{2*}}}{2\cancel{\sigma^{2*}}} + \text{const}$$

$$= -\frac{1}{2}\log \sigma^{2*} + \text{const}$$

③ $E_{\mu,\sigma^2}[\log q_{\sigma^2}^*(\sigma^2)] = E_{\mu,\sigma^2}\left[a^* \log b^* - \log \Gamma(a^*) + (a^*+1)\log \frac{1}{\sigma^2} - b^* \frac{1}{\sigma^2}\right]$

$$= a^* \log b^* - \log\Gamma(a^*) - (a^*+1) E_{\sigma^2}[\log \sigma^2] - b^* E_{\sigma^2}\left[\frac{1}{\sigma^2}\right]$$

$$= a^* \log b^* - \log\Gamma(a^*) - (a^*+1)(\log b^* - \psi(a^*)) - \cancel{b^*}\frac{a^*}{\cancel{b^*}}$$

$$= a^* \log b^* - \log\Gamma(a^*) - (a^*+1)(\log b^* - \psi(a^*)) - a^*$$

where $\psi(a^*) = \frac{d}{da^*}\log\Gamma(a^*)$

① $E_{\mu,\sigma^2}[(\log \prod_{i=1}^{100} P(x_i \mid \mu, \sigma^2) P(\mu \mid \sigma^2) P(\sigma^2)]$

$$= E_{\mu,\sigma^2}\left[\log \prod_{i=1}^{100} P(x_i \mid \mu, \sigma^2) P(\mu \mid \sigma^2) P(\sigma^2)\right]$$

$$= E_{\mu,\sigma^2}\left[50 \log \frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 + \frac{1}{2}\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}(\mu-\mu_0) + \log P(\sigma^2)\right] + \text{const}$$

$$= \underbrace{E_{\mu,\sigma^2}\left[50 \log \frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 + \frac{1}{2}\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}(\mu-\mu_0)^2\right]}_{④} + \underbrace{E_{\sigma^2}[\log P(\sigma^2)]}_{⑤} + \text{const}$$

$$= -50s(\log b^* - \psi(a^*)) - \frac{1}{2}\frac{a^*}{b^*}\left(\sum_{i=1}^{100}(\sigma^{2*} + \mu^{*2} - 2x_i\mu^* + x_i^2) + \sigma^{2*} + \mu^{*2} - 2\mu_0\mu^* + \mu_0^2\right)$$

$$+ (a_0-1)[\log b^* - \psi(a^*)] - b_0 \frac{a^*}{b^*} + \text{const}$$

④ $E_{\mu,\sigma^2}\left[50 \log \frac{1}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{100}(x_i-\mu)^2 + \frac{1}{2}\log\frac{1}{\sigma^2} - \frac{1}{2\sigma^2}(\mu-\mu_0)^2\right]$

$$= -50 E_{\sigma^2}[\log \sigma^2] - \frac{1}{2}E_{\sigma^2}\left[\frac{1}{\sigma^2}\right]\sum_{i=1}^{100} E_\mu[(x_i-\mu)^2] - \frac{1}{2}E_{\sigma^2}[\log \sigma^2] - \frac{1}{2}E_{\sigma^2}\left[\frac{1}{\sigma^2}\right]E_\mu[(\mu-\mu_0)^2]]$$

$$= -50s(\log b^* - \psi(a^*)) - \frac{1}{2}\frac{a^*}{b^*}\left(\underbrace{\sum_{i=1}^{100}(\sigma^{2*} + \mu^{*2} - 2x_i\mu^* + x_i^2)}_{A} + \underbrace{\sigma^{2*} + \mu^{*2} - 2\mu_0\mu^* + \mu_0^2}_{B}\right)$$

⑤ $E_{\sigma^2}[\log P(\sigma^2)] = E_{\sigma^2}\left[(-a_0-1)\log(\sigma^2) - b_0\frac{1}{\sigma^2}\right] + \text{const}$

$$= (a_0-1)E_{\sigma^2}[\log \sigma^2] - b_0 E_{\sigma^2}\left[\frac{1}{\sigma^2}\right]] + \text{const}$$

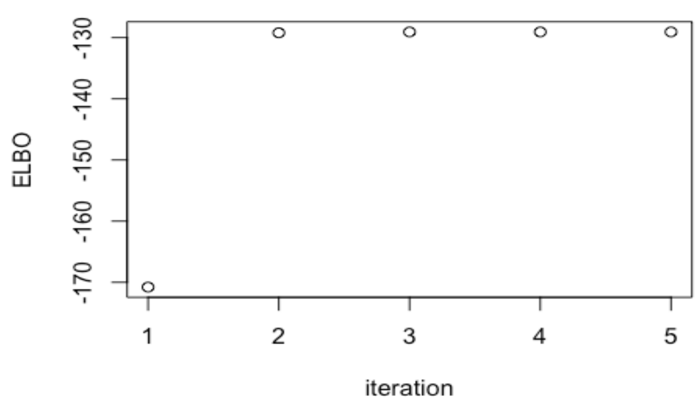
$$= (a_0-1)[\log b^* - \psi(a^*)] - b_0\frac{a^*}{b^*} + \text{const}$$

Q3

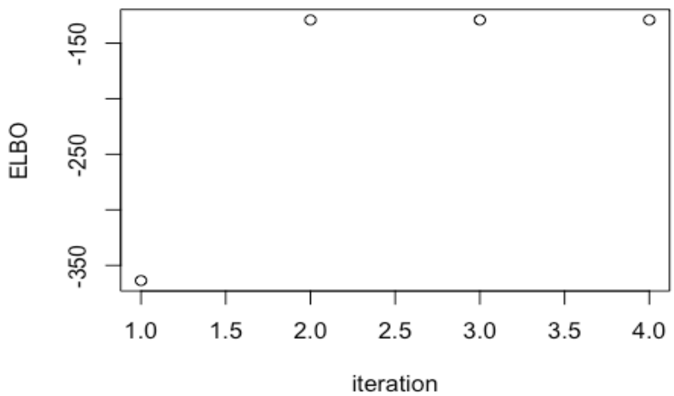
3c

```
cavi.normal = function(X, mu0, a0, b0, mu.vi.init, sigmasq.vi.init, a.vi.init, b.vi.init, epsilon = 1e-5,
max.iter = 100) {

  # initiate variables
  mu.vi = mu.vi.init
  sigmasq.vi = sigmasq.vi.init
  a.vi = a.vi.init
  b.vi = b.vi.init

  # create list to store the values from each iteration, for better endgame visualisation
  elbo = c()
  mu.vi.list = sigmasq.vi.list = a.vi.list = b.vi.list = c()

  # compute ELBO using initial values of mu*, sigmasq*, a*, b*
  Elogq.mu = -0.5 * log(sigmasq.vi)
  Elogq.sigmasq = a.vi * log(b.vi) - log(gamma(a.vi)) - (a.vi+1)*(log(b.vi) - digamma(a.vi)) - a.vi

  A = sigmasq.vi + mu.vi^2 -2*X*mu.vi + X*X
  B = sigmasq.vi + mu.vi^2 - 2*mu0*mu.vi + mu0^2

  Elogp.x.mu.tau = -50.5*(log(b.vi) - digamma(a.vi)) - (a.vi/(2*b.vi))*(sum(A)+B) + (-a0-1)*(log(b.vi)-
digamma(a.vi)) - b0*a.vi/b.vi

  elbo = c(elbo, Elogp.x.mu.tau-Elogq.sigmasq-Elogq.mu)
  mu.vi.list = c(mu.vi.list, mu.vi)
  sigmasq.vi.list = c(sigmasq.vi.list, sigmasq.vi)
  a.vi.list = c(a.vi.list, a.vi)
  b.vi.list = c(b.vi.list, b.vi)


  # initialise values for iteration
  delta.elbo = 1
  n.iter = 1

  while((delta.elbo > epsilon) & (n.iter <= max.iter)){

    # update mu.vi, sigma.vi, a.vi and b.vi
    mu.vi = (sum(X) + mu0)/101
    sigmasq.vi = 1/(101*(a.vi/b.vi))

    a.vi = 50.5+(a0+1)-1
    A = sigmasq.vi + mu.vi^2 -2*X*mu.vi + X*X
    B = sigmasq.vi + mu.vi^2 - 2*mu0*mu.vi + mu0^2
    b.vi = 0.5*sum(A) + 0.5*B + b0

    # compute ELBO using current values of mu*, sigmasq*, a*, b*

    Elogq.mu = -0.5 * log(sigmasq.vi)
    Elogq.sigmasq = a.vi * log(b.vi) - log(gamma(a.vi)) - (a.vi+1)*(log(b.vi) - digamma(a.vi)) - a.vi

    Elogp.x.mu.tau = -50.5*(log(b.vi)-digamma(a.vi)) - (a.vi/(2*b.vi))*(sum(A)+B) + (-a0-1)*(log(b.vi)-
digamma(a.vi))-b0*a.vi/b.vi

    elbo = c(elbo, Elogp.x.mu.tau-Elogq.sigmasq-Elogq.mu)
    mu.vi.list = c(mu.vi.list, mu.vi)
    sigmasq.vi.list = c(sigmasq.vi.list, sigmasq.vi)
    a.vi.list = c(a.vi.list, a.vi)
    b.vi.list = c(b.vi.list, b.vi)

    # calculate results of this loop
    delta.elbo = elbo[length(elbo)] - elbo[length(elbo)-1]
    n.iter = n.iter + 1
```

```
  }

  return(list(elbo=elbo, mu.vi.list=mu.vi.list, sigmasq.vi.list=sigmasq.vi.list, a.vi.list=a.vi.list,
b.vi.list=b.vi.list))
}

# set prior parameters
mu0=0
a0=2
b0=2

# first run
cavi1 = cavi.normal(X, mu0=mu0, a0=a0, b0=b0, mu.vi.init=5, sigmasq.vi.init=1, a.vi.init=1, b.vi.init=8,
epsilon=1e-5, max.iter=100)

cavi.res=cavi1
cavi.res$elbo

## [1] -170.7847 -129.2141 -129.0734 -129.0734 -129.0734

plot(cavi.res$elbo, ylab = 'ELBO', xlab='iteration')
```




```
print(paste("mu* and sigmasq* = (",
            round(cavi.res$mu.vi.list[length(cavi.res$mu.vi.list)],2), ", ",
            round(cavi.res$sigmasq.vi.list[length(cavi.res$sigmasq.vi.list)],2), ")", sep=""))

## [1] "mu* and sigmasq* = (5.04, 0.04)"

print(paste("a* and b* = (",
            round(cavi.res$a.vi.list[length(cavi.res$a.vi.list)],2), ", ",
            round(cavi.res$b.vi.list[length(cavi.res$b.vi.list)],2), ")", sep=""))

## [1] "a* and b* = (52.5, 214.57)"

# second run
cavi2 = cavi.normal(X, mu0=mu0, a0=a0, b0=b0, mu.vi.init=2, sigmasq.vi.init=4, a.vi.init=2, b.vi.init=6,
epsilon=1e-5, max.iter=100)

cavi.res=cavi2
cavi.res$elbo

## [1] -363.6567 -129.0948 -129.0734 -129.0734

plot(cavi.res$elbo, ylab = 'ELBO', xlab='iteration')
```




```
print(paste("mu* and sigmasq* = (",
            round(cavi.res$mu.vi.list[length(cavi.res$mu.vi.list)],2), ", ",
            round(cavi.res$sigmasq.vi.list[length(cavi.res$sigmasq.vi.list)],2), ")", sep=""))

## [1] "mu* and sigmasq* = (5.04, 0.04)"

print(paste("a* and b* = (",
            round(cavi.res$a.vi.list[length(cavi.res$a.vi.list)],2), ", ",
            round(cavi.res$b.vi.list[length(cavi.res$b.vi.list)],2), ")", sep=""))

## [1] "a* and b* = (52.5, 214.57)"
```

Both CAVI runs have the same ELBO value at -129.0734, so the highet ELBO q*(mu) ~ N(5.04, 0.04); q*(sigmasq) ~ InvGamma(52.5, 214.57)