Compulsory HWA2, Bayesian Statistics

1. (1p) Consider the posterior $(\beta, \sigma^2) \mid x \sim \mathrm{NIG}(a_n, b_n, \mu_n, \lambda_n^{-1})$, where β is 1×1 and σ^2 is also 1×1 . Suppose that the loss of the wrong decision is

	Truth	
Decision	H_0	H_1
H_0	0	L_1
H_1	L_0	0

such that $L_0 + L_1 > 0$.

(a) Can you find the optimal Bayes test for the hypothesis $H_0 : \beta \leq 0$ and $\sigma^2 \geq 1$ versus $H_1 : H_0$ does not hold? If so, derive the Bayes test. Otherwise state the reason. If no closed form expression can be obtained, simplify the expressions as much as you can.
Solution: The optimal Bayes test is

$$\phi(x) = \begin{cases} 1, & \text{if } \mathrm{P}(H_0 \mid x) < \frac{L_1}{L_0+L_1}, \\ 0, & \text{if } \mathrm{P}(H_0 \mid x) \geq \frac{L_1}{L_0+L_1}. \end{cases}$$

The posterior probability of H_0 can be obtained by

$$\begin{aligned} \mathrm{P}(H_0 \mid x) = \mathrm{P}(\beta \leq 0 \text{ and } \sigma^2 \geq 1 \mid x) &= \int_1^{\infty} \int_{-\infty}^{0} \pi(\beta \mid \sigma^2, x)\, \pi(\sigma^2 \mid x)\, d\beta d\sigma^2 \\ &= \int_1^{\infty} \Phi\left(-\frac{\mu}{\sigma/\sqrt{\lambda_n}}\right) \pi(\sigma^2 \mid x)\, d\sigma^2, \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution of $N(0,1)$ and $\sigma^2 \mid x \sim \mathrm{InvGamma}(a_n, b_n)$.

(b) Can you find the optimal Bayes test for the hypothesis $H_0 : \beta \leq 0$ versus $H_1 : \beta > 0$? If so, derive the Bayes test. Otherwise state the reason. If no closed form expression can be obtained, simplify the expressions as much as you can.
Solution: The optimal Bayes test is

$$\phi(x) = \begin{cases} 1, & \text{if } \mathrm{P}(H_0 \mid x) < \frac{L_1}{L_0+L_1}, \\ 0, & \text{if } \mathrm{P}(H_0 \mid x) \geq \frac{L_1}{L_0+L_1}. \end{cases}$$

The posterior probability if H_0 can be obtained by

$$\mathrm{P}(H_0 \mid x) = \mathrm{P}(\beta \leq 0 \mid x) = \int_{-\infty}^{0} \pi(\beta \mid x)\, d\beta,$$

where $\beta \mid x \sim t_{2a_n}\left(\mu_n, \frac{b_n}{a_n}\lambda_n^{-1}\right)$.

2. (1p) Suppose that we have an iid sample $(x_1, ..., x_n)$ from $N(\mu, \sigma^2)$, where both μ and σ are unknown. The parameter is $\theta = (\mu, \sigma^2)$. We are interested in testing

$$H_0 : \mu = 0 \quad \text{versus} \quad H_1 : \mu \neq 0.$$

We consider the improper priors $\pi_0(\sigma^2) = \sigma^{-2}$ and $\pi_1(\mu, \sigma^2) = \sigma^{-2}$.

(a) Find the minimal training sample size.
Solution: H_1 has two parameters whereas H_0 has only one parameter. The posterior under H_1 satisfies

$$\begin{aligned}
\pi(\mu, \sigma^2 \mid x) &\propto \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}\right] \cdot \sigma^{-2} \\
&\propto \underbrace{\frac{1}{(\sigma^2)^{1/2}} \exp\left\{-\frac{(\bar{x}-\mu)^2}{2\sigma^2/n}\right\}}_{\text{normal for any } n \geq 1} \cdot \underbrace{\frac{1}{(\sigma^2)^{(n+1)/2}} \exp\left\{-\frac{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}{2\sigma^2}\right\}}_{\text{need } n \geq 2},
\end{aligned}$$

where the second part needs $n \geq 2$ in order for $\sum_{i=1}^{n} x_i^2 - n\bar{x}^2 > 0$. Hence, the minimal training sample size is 2.

(b) Derive the intrinsic Bayes factor.
Solution: Let's say we take x_1 and x_2 as the minimal training sample. We first need to find out $\pi_0(\sigma^2 \mid x_1, x_2)$ and $\pi_1(\mu, \sigma^2 \mid x_1, x_2)$ under H_0 and H_1, respectively. Under H_0,

$$\begin{aligned}
\pi_0(\sigma^2 \mid x_1, x_2) &\propto \left[\prod_{i=1}^{2} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{x_i^2}{2\sigma^2}\right\}\right] \cdot \sigma^{-2} \\
&\propto \frac{1}{(\sigma^2)^2} \exp\left\{-\frac{x_1^2 + x_2^2}{2\sigma^2}\right\} \sim \text{InvGamma}\left(1, \frac{1}{2}(x_1^2 + x_2^2)\right).
\end{aligned}$$

Hence,

$$\pi_0(\sigma^2 \mid x_1, x_2) = \frac{\frac{1}{2}(x_1^2 + x_2^2)}{\Gamma(1)} \frac{1}{(\sigma^2)^2} \exp\left\{-\frac{x_1^2 + x_2^2}{2\sigma^2}\right\} = \frac{x_1^2 + x_2^2}{2(\sigma^2)^2} \exp\left\{-\frac{x_1^2 + x_2^2}{2\sigma^2}\right\}.$$

Under H_1,

$$\begin{aligned}
\pi_1(\mu, \sigma^2 \mid x) &= \frac{1}{\sqrt{2\pi\sigma^2/2}} \exp\left\{-\frac{\left(\mu - \frac{x_1+x_2}{2}\right)^2}{2\sigma^2/2}\right\} \times \left[\frac{1}{2}\left(x_1^2 + x_2^2 - 2\left(\frac{x_1+x_2}{2}\right)^2\right)\right]^{1/2} \\
&\quad \frac{1}{\Gamma(1/2)} \frac{1}{(\sigma^2)^{3/2}} \exp\left\{-\frac{1}{\sigma^2} \times \frac{1}{2}\left(x_1^2 + x_2^2 - 2\left(\frac{x_1+x_2}{2}\right)^2\right)\right\} \\
&= \frac{1}{\pi(\sigma^2)^2} \left[\frac{1}{2}\left(x_1^2 + x_2^2 - 2\left(\frac{x_1+x_2}{2}\right)^2\right)\right]^{1/2} \\
&\quad \times \exp\left\{-\frac{\left(\mu - \frac{x_1+x_2}{2}\right)^2}{\sigma^2} - \frac{1}{2\sigma^2}\left[x_1^2 + x_2^2 - 2\left(\frac{x_1+x_2}{2}\right)^2\right]\right\}
\end{aligned}$$

where

$$\begin{aligned}
\mu \mid \sigma^2, x &\sim N\left(\frac{x_1+x_2}{2}, \frac{\sigma^2}{2}\right), \\
\sigma^2 \mid x &\sim \text{InvGamma}\left(\frac{1}{2}, \frac{1}{2}\left[x_1^2 + x_2^2 - 2\left(\frac{x_1+x_2}{2}\right)^2\right]\right).
\end{aligned}$$

We will use these distributions as priors for the rest of data. Hence,

$$
\begin{aligned}
m_0\left(x_{-\ell}\right) &= \int_0^{\infty}\left[\prod_{i=3}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{x_i^2}{2\sigma^2}\right\}\right]\cdot\frac{x_1^2+x_2^2}{2\left(\sigma^2\right)^2}\exp\left\{-\frac{x_1^2+x_2^2}{2\sigma^2}\right\}d\sigma^2 \\
&= \frac{x_1^2+x_2^2}{2\left(2\pi\right)^{(n-2)/2}}\int_0^{\infty}\frac{1}{\left(\sigma^2\right)^{n/2+1}}\exp\left\{-\frac{\sum_{i=1}^{n}x_i^2}{2\sigma^2}\right\}d\sigma^2 \\
&= \frac{x_1^2+x_2^2}{\pi^{n/2-1}}\frac{\Gamma\left(n/2\right)}{\left(\sum_{i=1}^{n}x_i^2\right)^{n/2}},
\end{aligned}
$$

and

$$
\begin{aligned}
m_1\left(x_{-\ell}\right) &= \int_0^{\infty}\int_{-\infty}^{\infty} f\left(x_3,...,x_n\mid\mu,\sigma^2\right)\pi_1\left(\mu\mid\sigma^2,x\right)\pi_1\left(\sigma^2\mid x\right)d\mu d\sigma^2 \\
&= \int_0^{\infty}\frac{\sqrt{2}}{2^{n/2-1}\pi^{n/2-1}\left(\sigma^2\right)^{n/2-1}\sqrt{n}}\exp\left\{-\frac{\sum_{i=3}^{n}x_i^2-n\left(\bar{X}\right)^2}{2\sigma^2}-\frac{\left(x_1+x_2\right)^2}{4\sigma^2}\right\}\pi_1\left(\sigma^2\mid x\right)d\sigma^2 \\
&= \frac{\left[\left(x_1^2+x_2^2-2\left(\frac{x_1+x_2}{2}\right)^2\right)\right]^{1/2}}{2^{n/2-1}\pi^{n/2-1}\sqrt{n}\Gamma\left(1/2\right)}\int_0^{\infty}\frac{1}{\left(\sigma^2\right)^{n/2+1/2}}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}x_i^2-n\left(\bar{X}\right)^2\right]\right\}d\sigma^2 \\
&= \frac{\left[\left(x_1^2+x_2^2-2\left(\frac{x_1+x_2}{2}\right)^2\right)\right]^{1/2}}{2^{n/2-1}\pi^{n/2-1}\sqrt{n}\Gamma\left(1/2\right)}\times\frac{\Gamma\left(n/2-1/2\right)}{\left(\frac{1}{2}\left[\sum_{i=1}^{n}x_i^2-n\left(\bar{X}\right)^2\right]\right)^{n/2-1/2}} \\
&= \frac{\sqrt{\left(x_1-x_2\right)^2}}{\pi^{(n-1)/2}\sqrt{n}}\frac{\Gamma\left(\frac{n-1}{2}\right)}{\left(\sum_{i=1}^{n}x_i^2-n\bar{x}^2\right)^{(n-1)/2}}.
\end{aligned}
$$

where

$$
\begin{aligned}
&\int_{-\infty}^{\infty} f\left(x_3,...,x_n\mid\mu,\sigma^2\right)\pi_1\left(\mu\mid\sigma^2,x\right)d\mu \\
&= \int_{-\infty}^{\infty}\frac{1}{2^{n/2-1}\pi^{n/2-1}\left(\sigma^2\right)^{n/2-1}}\exp\left\{-\frac{\sum_{i=3}^{n}\left(x_i-\mu\right)^2}{2\sigma^2}\right\}\frac{1}{\sqrt{2\pi\sigma^2/2}}\exp\left\{-\frac{\left(\mu-\frac{x_1+x_2}{2}\right)^2}{2\sigma^2/2}\right\}d\mu \\
&= \frac{1}{2^{n/2-1}\pi^{n/2-1}\left(\sigma^2\right)^{n/2-1}}\frac{1}{\sqrt{2\pi\sigma^2/2}}\exp\left\{-\frac{\sum_{i=3}^{n}x_i^2-n\left(\bar{X}\right)^2}{2\sigma^2}-\frac{\left(x_1+x_2\right)^2}{4\sigma^2}\right\}\int_{-\infty}^{\infty}\exp\left\{-\frac{\left(\mu-\bar{X}\right)^2}{2\sigma^2/n}\right\}d\mu \\
&= \frac{\sqrt{2}}{2^{n/2-1}\pi^{n/2-1}\left(\sigma^2\right)^{n/2-1}\sqrt{n}}\exp\left\{-\frac{\sum_{i=3}^{n}x_i^2-n\left(\bar{X}\right)^2}{2\sigma^2}-\frac{\left(x_1+x_2\right)^2}{4\sigma^2}\right\}.
\end{aligned}
$$

Thus, the intrinsic Bayes factor is

$$
\begin{aligned}
B_{01}^{I} &= \frac{\int\limits_{\Theta_0} f_0\left(x_{-(l)} \mid \theta\right) \pi_0\left(\theta \mid x_{(l)}\right) d\theta}{\int\limits_{\Theta_1} f_1\left(x_{-(l)} \mid \theta\right) \pi_1\left(\theta \mid x_{(l)}\right) d\theta} \\
&= \frac{\frac{x_1^2+x_2^2}{\pi^{n/2-1}} \frac{\Gamma(n/2)}{\left(\sum_{i=1}^{n} x_i^2\right)^{n/2}}}{\frac{\sqrt{(x_1-x_2)^2}}{\pi^{(n-1)/2}\sqrt{n}} \frac{\Gamma\left(\frac{n-1}{2}\right)}{\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2\right)^{(n-1)/2}}} \\
&= \sqrt{n\pi} \frac{\Gamma\left(n/2\right) \left[\sum_{i=1}^{n} x_i^2 - n\bar{x}^2\right]^{(n-1)/2}}{\Gamma\left((n-1)/2\right) \left[\sum_{i=1}^{n} x_i^2\right]^{n/2}} \times \frac{x_1^2 + x_2^2}{\sqrt{\left(x_1 - x_2\right)^2}}.
\end{aligned}
$$

3. (3p) Suppose that we observe an iid sample X_1, ..., X_n from $\mathrm{Exp}\left(\theta\right)$ with mean $1/\theta$. We let the prior of θ be $\mathrm{Gamma}\left(a_0, b_0\right)$.

   (a) Derive BIC of this model.
   Solution: The log-likelihood is

$$
\log f\left(x \mid \theta\right) = n \log \theta - \theta \sum_{i=1}^{n} x_i.
$$

   Note that

$$
\frac{d \log f\left(x \mid \theta\right)}{d\theta} = \frac{n}{\theta} - \sum_{i=1}^{n} x_i.
$$

   The MLE is $\hat{\theta} = n / \sum_{i=1}^{n} x_i = 1/\bar{X}$, where $\bar{x} = n^{-1} \sum_{i=1}^{n} x_i$. With $p = 1$, we get

$$
\begin{aligned}
\mathrm{BIC} &= -2 \max_{\theta} \log f\left(x \mid \theta\right) + \log\left(n\right) \\
&= -2 \left[n \log\left(\frac{1}{\bar{x}}\right) - \frac{1}{\bar{x}} \sum_{i=1}^{n} x_i \right] + \log\left(n\right).
\end{aligned}
$$

   (b) Derive DIC of this model. You may need the result that if $X \sim \mathrm{Gamma}\left(a, b\right)$, then $\mathrm{E}\left[X\right] = a/b$ and $\mathrm{E}\left[\log X\right] = \psi\left(a\right) - \log\left(b\right)$, where $\psi\left(\cdot\right)$ is a digamma function.
   Solution: Ignoring the constant, the deviance is

$$
D\left(\theta\right) = -2 \log f\left(x \mid \theta\right) = -2n \log \theta + 2\theta \sum_{i=1}^{n} x_i.
$$

   The posterior distribution satisfies

$$
\pi\left(\theta \mid x\right) \propto \theta^{n} \exp\left\{-\theta \sum_{i=1}^{n} x_i\right\} \cdot \theta^{a_0 - 1} \exp\left\{-b_0 \theta\right\} \sim \mathrm{Gamma}\left(a_0 + n, b_0 + \sum_{i=1}^{n} x_i\right).
$$

   Hence, $\mathrm{E}\left[\theta \mid x\right] = \frac{a_0+n}{b_0+\sum_{i=1}^{n} x_i}$ and

$$
D\left(\mathrm{E}\left[\theta \mid x\right]\right) = -2n \log\left(\frac{a_0 + n}{b_0 + \sum_{i=1}^{n} x_i}\right) + 2 \frac{a_0 + n}{b_0 + \sum_{i=1}^{n} x_i} \sum_{i=1}^{n} x_i.
$$

The posterior expectation of deviance is

$$\begin{aligned}
\mathrm{E}\left[D(\theta) \mid x\right] &= -2n\mathrm{E}\left[\log\theta \mid x\right] + 2\mathrm{E}\left[\theta \mid x\right]\sum_{i=1}^{n} x_i \\
&= -2n\left[\psi\left(a_0+n\right) - \log\left(b_0 + \sum_{i=1}^{n} x_i\right)\right] + 2\frac{a_0+n}{b_0+\sum_{i=1}^{n} x_i}\sum_{i=1}^{n} x_i.
\end{aligned}$$

Then,

$$\mathrm{DIC} = 2\mathrm{E}\left[D(\theta) \mid x\right] - D\left(\mathrm{E}\left[\theta \mid x\right]\right),$$

where both terms are presented above.

(c) Suppose that we want to approximate the posterior density by an exponential density $\mathrm{Exp}(\lambda)$ with mean λ^{-1}. Find the corresponding ELBO. Can you also find λ that maximizes ELBO?

Solution: Let $q(\theta \mid x)$ be a exponential density of θ. Then,

$$\begin{aligned}
\mathrm{ELBO}(q) &= \int q(\theta \mid x)\log\left[\frac{p(\theta, x)}{q(\theta \mid x)}\right] d\theta \\
&= \int q(\theta \mid x)\log\left[\frac{\theta^n \exp\left\{-\theta\sum_{i=1}^{n} x_i\right\} \cdot \frac{b_0^{a_0}}{\Gamma(a_0)}\theta^{a_0-1}\exp\left\{-b_0\theta\right\}}{\lambda\exp\left(-\lambda\theta\right)}\right] d\theta \\
&= \mathrm{E}\left[n\log\theta - \theta\sum_{i=1}^{n} x_i + \log\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right) + (a_0-1)\log\theta - b_0\theta\right] - \mathrm{E}\left[\log(\lambda) - \lambda\theta\right]
\end{aligned}$$

where the expectation is taken with respect to $\theta \sim \mathrm{Exp}(\lambda)$. We can simplify the expectations to

$$\mathrm{ELBO}(q) = (n+a_0-1)\,\psi(1) - (n+a_0)\log\lambda - \lambda^{-1}\left(\sum_{i=1}^{n} x_i + b_0\right) + \log\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right) + 1.$$

Note that

$$\frac{d\mathrm{ELBO}(q)}{d\lambda} = \frac{n+a_0}{\lambda} + \lambda^{-2}\left(\sum_{i=1}^{n} x_i + b_0\right).$$

The stationary point is

$$\lambda = \frac{\sum_{i=1}^{n} x_i + b_0}{n+a_0}.$$

The second derivative is

$$\frac{d^2\mathrm{ELBO}(q)}{d\lambda^2} = -\frac{n+a_0}{\lambda^2} - 2\lambda^{-3}\left(\sum_{i=1}^{n} x_i + b_0\right) = -\frac{\lambda(n+a_0) + 2\left(\sum_{i=1}^{n} x_i + b_0\right)}{\lambda^2} < 0.$$

Hence, we just let $\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i + b_0}{n+a_0}$.

4. (4p) Consider the data set in HWA2.csv. We want to model the data by the regression model $y = \alpha + \beta x + \epsilon$, where $\epsilon \sim N\left(0, \sigma^2\right)$. The sample size is $n = 100$. Consider the independent t priors for α and β, and the exponential prior $\mathrm{Exp}(1)$ for σ. That is, $\alpha \sim t(3)$, $\beta \sim t(3)$, and $\sigma \sim \mathrm{Exp}(1)$.

(a) Fit such normal linear model by MCMC in software of your choice. For simplicity, you can use the default priors and proposal distribution, if any. Choose the length of the Markov chain such that the length is at least $10,000$ after discarding the burn in period. What are the posterior means of the regression coefficients?
Solution: We fit the model in the R package rstanarm.

```
library(rstanarm)
Fit <- stan_glm(y ~ x, family = gaussian(), data = HWA2, iter = 5000,
                warmup = 2500, refresh = 0)
```

If I use the rstanarm package, we have already discarded the burn-in period. The posterior mean can be computed by

```
PostDraw <- as.matrix(Fit) # Posterior draws
colMeans(PostDraw)

## (Intercept)           x       sigma
##   0.9895569   0.6887367   0.5463672
```

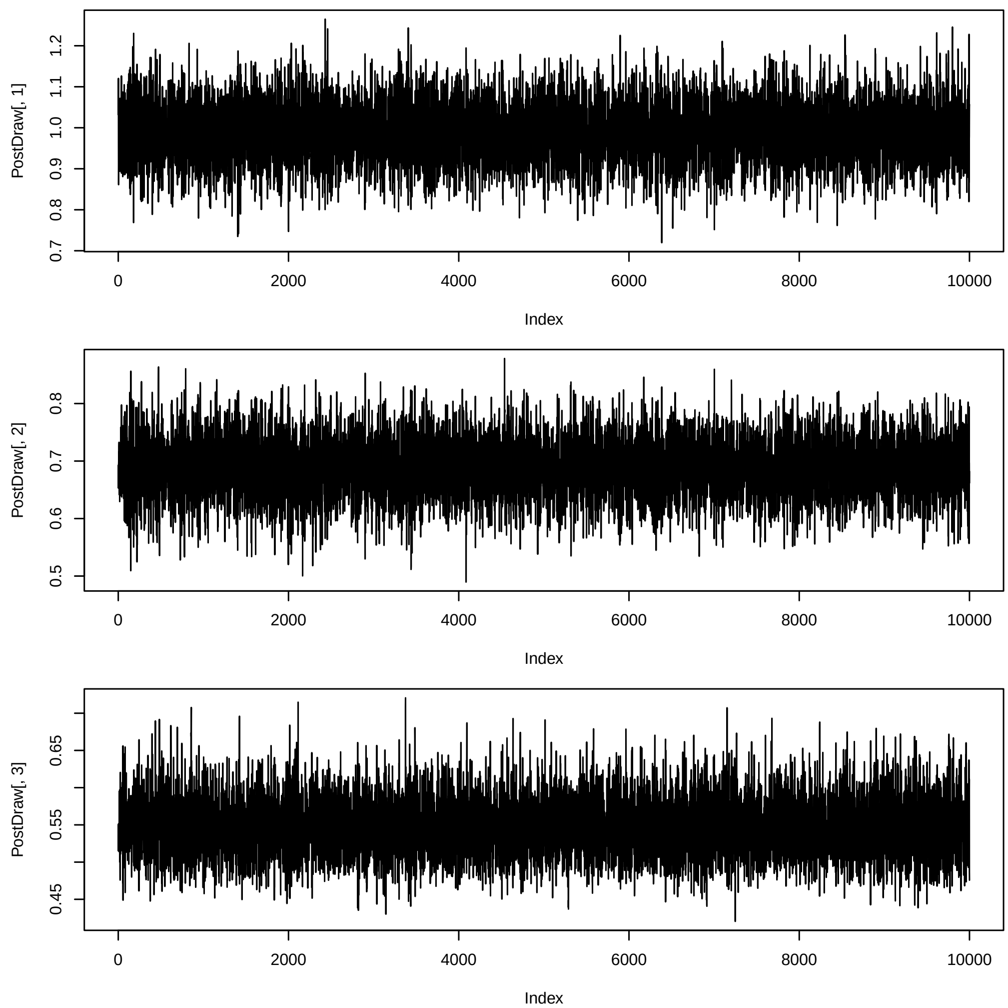
(b) What are the 95% credible intervals of the slope β?
Solution: The credible interval can be obtained by

```
posterior_interval(Fit, prob = 0.95)

##                  2.5%     97.5%
## (Intercept) 0.8486343 1.1315757
## x           0.5884533 0.7869861
## sigma       0.4764085 0.6308347
```

(c) Investigate also whether the Markov chains are converged.
Solution: We can investigate convergence visually.

```
par(mfrow = c(3, 1), mar = c(4.1, 4.1, 1, 1))
plot(PostDraw[, 1], type = "l")
plot(PostDraw[, 2], type = "l")
plot(PostDraw[, 3], type = "l")
```



They demonstrate good mixing. We can also look at summary statistics

```
summary(Fit)
```

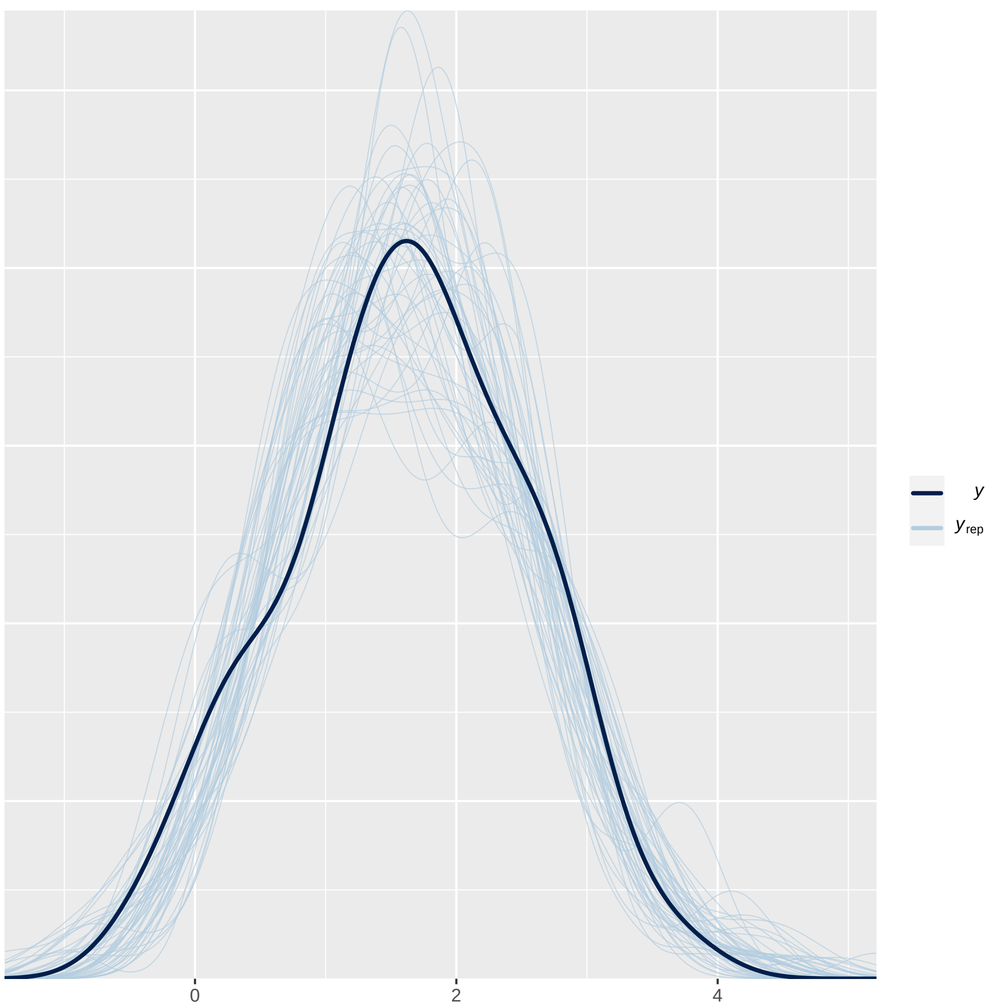
```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  algorithm:    sampling
##  sample:       10000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 100
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) 1.0    0.1  0.9   1.0   1.1
## x           0.7    0.1  0.6   0.7   0.8
## sigma       0.5    0.0  0.5   0.5   0.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
```

```
## mean_PPD 1.6    0.1  1.5   1.6   1.7
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome vari
##
## MCMC diagnostics
##              mcse Rhat n_eff
## (Intercept)   0.0  1.0  10396
## x             0.0  1.0   9238
## sigma         0.0  1.0   9695
## mean_PPD      0.0  1.0   9778
## log-posterior 0.0  1.0   4268
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effe
```

The $\hat{R}$'s are all 1, indicating convergence.

(d) Perform posterior predictive check. What conclusions can be draw?
Solution: We perform posterior predictive check using

```
pp_check(Fit)
```



It is seen that the simulated data are close to the observed data.

(e) Suppose that we have observed a new observation with $\tilde{x} = 2.3$. Approximate the mean

of predictive distribution. Write your own script for this task.
Solution: The predictive density is

$$f(\tilde{y} \mid y, \tilde{x}) \quad = \quad \int f(\tilde{y} \mid \tilde{x}, \theta) \, \pi(\theta \mid y) \, d\theta.$$

We can sample $\theta \sim \pi(\theta \mid y)$ in MCMC and then sample $\tilde{y} \mid \tilde{x}, \theta$. Hence, we can approximate the mean of predictive distribution by

$$\mathrm{E}(\tilde{y} \mid y, \tilde{x}) \quad = \quad \frac{1}{L} \sum_{j=1}^{L} \tilde{y}^{(l)},$$

where $\theta^{(l)}$ is the lth iteration in MCMC. Then, we perform sampling

```
ytilde <- rep(NA, nrow(PostDraw))
for(j in 1 : nrow(PostDraw)){
      ytilde[j] <- rnorm(n = 1, mean = PostDraw[j, 1]+ PostDraw[j, 2] * 2.3,
                         sd = PostDraw[j, 3])
}
```

The approximated mean is

```
mean(ytilde)

## [1] 2.572996
```

(f) Suppose that we have observed a new observation with $\tilde{x} = 2.3$. Approximate the predictive density function. Write your own script for this task.
Solution: We can approximate the predictive distribution by

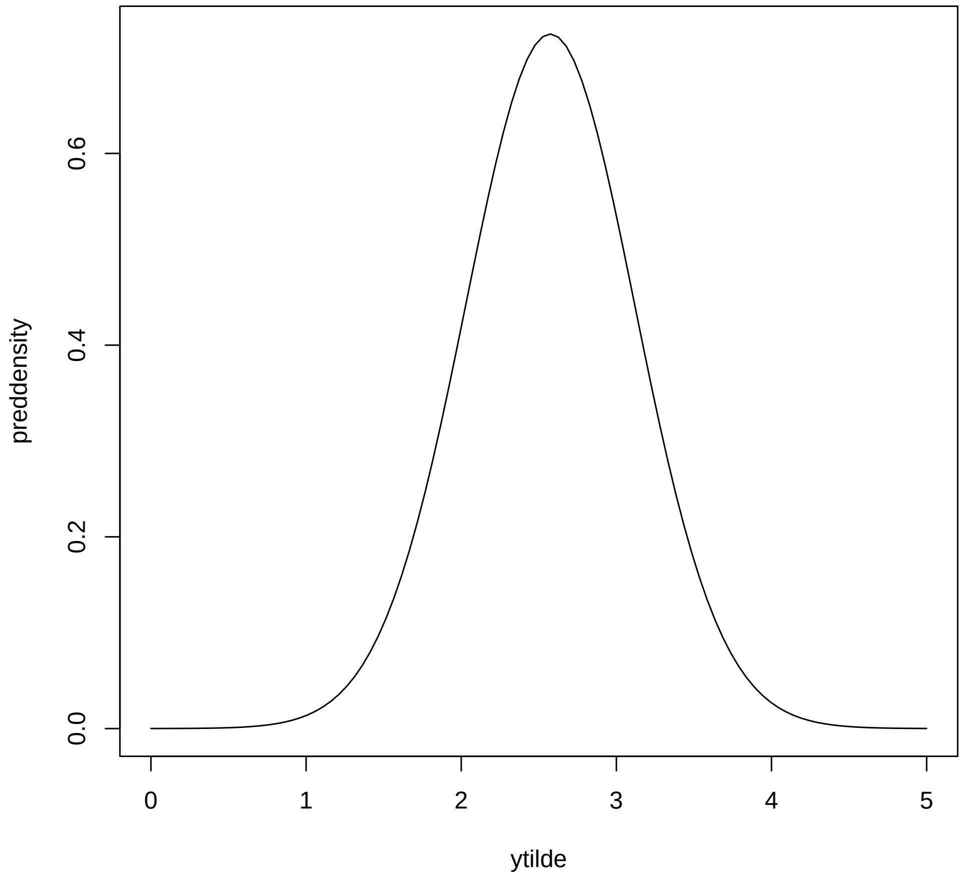
$$f(\tilde{y} \mid y, \tilde{x}) \quad = \quad \frac{1}{L} \sum_{j=1}^{L} f\left(\tilde{y} \mid \tilde{x}, \theta^{(l)}\right),$$

where $\theta^{(l)}$ is the lth iteration in MCMC. We can pick a grid of $\tilde{y}$ and evaluate $f(\tilde{y} \mid y, \tilde{x})$ for each value.

```
## Define likelihood
likelihood <- function(y, x, alpha, beta, sigma, log){
   out <- dnorm(y, mean = alpha + beta * x, sd = sigma, log = log)
   if(log == TRUE){
      return(sum(out))
   } else {
      return(prod(out))
   }
}
## Approximate predictive density
ytilde <- seq(0, 5, length.out = 1e02)
preddensity <- rep(0, 1e02)
for(i in 1 : 1e02){
   for(j in 1 : nrow(PostDraw)){
      L <- likelihood(y = ytilde[i], x = 2.3, alpha = PostDraw[j, 1],
                      beta = PostDraw[j, 2], sigma = PostDraw[j, 3], log = FALSE)
      preddensity[i] <- preddensity[i] + L
   }
}
preddensity <- preddensity / nrow(PostDraw)
```

We can then plot the predictive distribution as

```
plot(ytilde, preddensity, type = "l")
```



An alternative approach is that we have simulated $\tilde{y}$ in task (e). Hence, we can obtain the histogram and use it as an approximation to the density.

(g) Approximate DIC of the fitted model. Write your own script for this task.
Solution: We need to approximate

$$\begin{aligned} \mathrm{E}\left[D\left(\theta\right) \mid x\right] = \int D\left(\theta\right) \pi\left(\theta \mid x\right) d\theta &\approx \frac{1}{T} \sum_{t=1}^{T} D\left(\theta_t\right), \\ \int \theta \pi\left(\theta \mid x\right) d\theta &\approx \frac{1}{T} \sum_{t=1}^{T} \theta_t. \end{aligned}$$

Ignoring the constant term, the deviance of the model is

$$D\left(\theta\right) = -2 \log f\left(x \mid \theta\right) = n \log\left(2\pi\right) + 2n \log \sigma + \frac{1}{\sigma^2} \sum_{i=1}^{n} \left(y_i - \alpha - \beta x_i\right)^2.$$

Hence, we can approximate $\mathrm{E}\left[D\left(\theta\right) \mid x\right]$ by

```r
deviance <- function(alpha, beta, sigma){
   n * log(2 * pi) + 2 * n * log(sigma) +
    sum((HWA2$y - alpha - beta * HWA2$x) ^ 2) / (sigma ^ 2)
}
n <- nrow(HWA2)# Sample size
Dtheta <- rep(NA, nrow(PostDraw))
for(j in 1 : nrow(PostDraw)){
   Dtheta[j] <- deviance(PostDraw[j, 1], PostDraw[j, 2], PostDraw[j, 3])
}
Mean_Dtheta <- mean(Dtheta)
```

We can approximate $\int \theta\pi\left(\theta \mid x\right) d\theta$ by

```r
Mean_theta <- colMeans(PostDraw) # Column means
```

Then, we can approximate DIC as follows.

```r
pD <- Mean_Dtheta - deviance(Mean_theta[1], Mean_theta[2], Mean_theta[3])
DIC <- Mean_Dtheta + pD
DIC

##    sigma
## 164.8233
```

5. (1p) This task is quite demanding. But it allows you to have more feelings about the MCMC algorithm for the above regression model. Consider still the data in HWA2.csv.

   (a) Write your own Metropolis-Hastings algorithm to sample from the posterior distribution of (α, β, σ). For simplicity, you can sample α, β, and $\log\sigma$ from independent normal distribution. Let the length of Markov Chain be $10,000$ after the burn-in period.
   Solution: We consider t priors and exponential prior. We consider the proposal distributions

$$
\begin{aligned}
\alpha^{(t+1)} \mid \alpha^{(t)} &\sim N\left(\alpha^{(t)}, \tau_1^2\right), \\
\beta^{(t+1)} \mid \beta^{(t)} &\sim N\left(\beta^{(t)}, \tau_1^2\right), \\
\log\sigma^{(t+1)} \mid \sigma^{(t)} &\sim N\left(\log\sigma^{(t)} - \frac{1}{2}\tau_2^2, \tau_2^2\right).
\end{aligned}
$$

The log-normal distribution for $\sigma^{(t+1)}$ ensures that $\mathrm{E}\left[\sigma^{(t+1)} \mid \sigma^{(t)}\right] = \sigma^{(t)}$.
The Metropolis-Hasting algorithm is

```r
## Define likelihood
likelihood <- function(y, x, alpha, beta, sigma, log){
   out <- dnorm(y, mean = alpha + beta * x, sd = sigma, log = log)
   if(log == TRUE){
      return(sum(out))
   } else {
      return(prod(out))
   }
}
## Define independent prior
prior <- function(alpha, beta, sigma, log = FALSE){
   out <- c(dt(alpha, df = 3, log = log), dt(beta, df = 3, log = log),
            dexp(sigma, rate = 1, log = log))
```

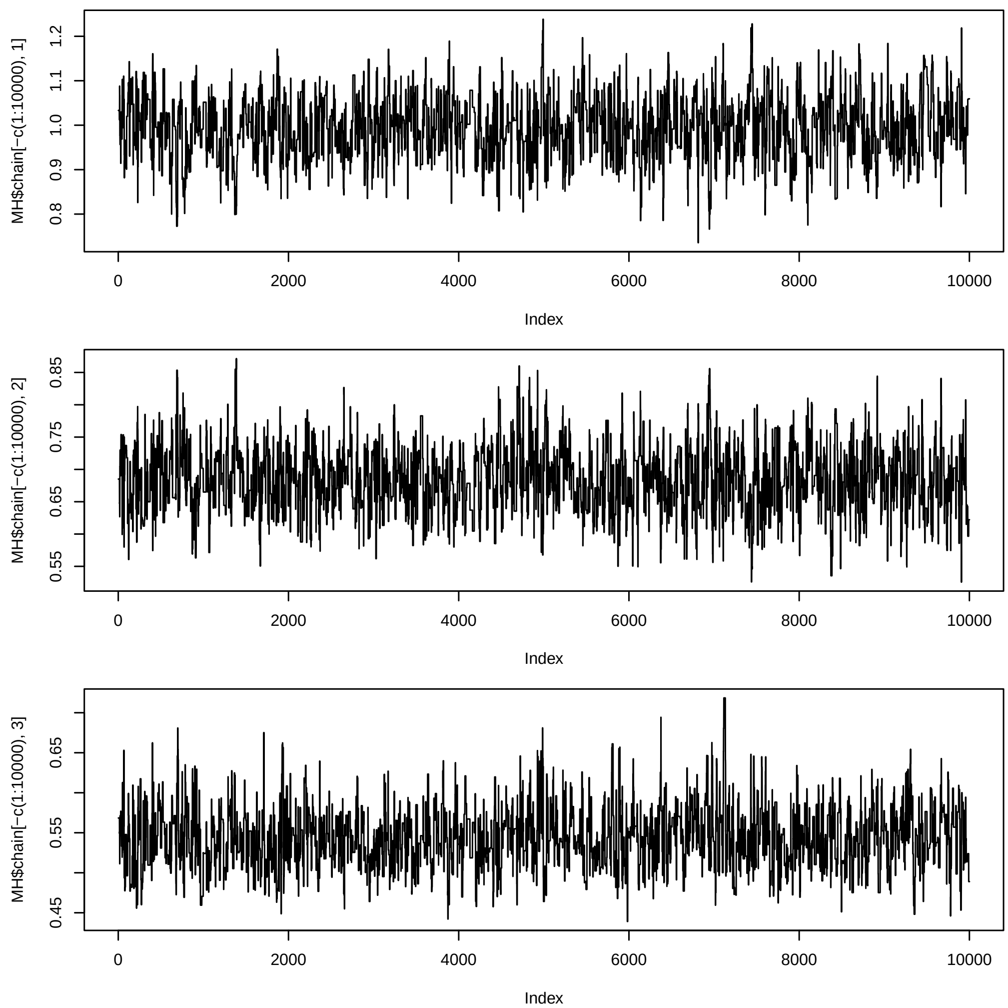
```
  if(log == TRUE){
    return(sum(out))
  } else {
    return(prod(out))
  }
}
## MCMC algorithm
MHalgorithm <- function(L, initial, sd, y, x) {
  # L is the length of the Markov chain, including the burn-in period
  # initial is the initial state
  # sd is the standard deviation of the proposal distribution for alpha, beta, and sigma
  chain <- rbind(c(initial), matrix(NA, L, length(initial)))
  accept <- 0
  reject <- 0
  for(t in 1 : L){
    # Propose a candidate
    alpha.prop <- rnorm(n = 1, mean = chain[t, 1], sd = sd[1])
    beta.prop <- rnorm(n = 1, mean = chain[t, 2], sd = sd[1])
    sigma.prop <- rlnorm(n = 1, meanlog = log(chain[t, 3]) - 0.5 * sd[2] ^ 2,
                         sdlog = sd[2])
    # Calculate the ratio
    log.numerator <- likelihood(y = y, x = x, alpha = alpha.prop, beta = beta.prop,
                                sigma = sigma.prop, log = TRUE) +
          prior(alpha = alpha.prop, beta = beta.prop, sigma = sigma.prop, log = TRUE) +
          dnorm(x = chain[t, 1], mean = alpha.prop, sd = sd[1], log = TRUE) +
          dnorm(x = chain[t, 2], mean = beta.prop, sd = sd[1], log = TRUE) +
          dlnorm(x = chain[t, 3], meanlog = log(sigma.prop) - 0.5 * sd[2] ^ 2,
                 sdlog = sd[2], log = TRUE)
    log.denominator <- likelihood(y = y, x = x, alpha = chain[t, 1], beta = chain[t, 2],
                                  sigma = chain[t, 3], log = TRUE) +
          prior(alpha = chain[t, 1], beta = chain[t, 2], sigma = chain[t, 3],
                log = TRUE) +
          dnorm(x = alpha.prop, mean = chain[t, 1], sd = sd[1], log = TRUE) +
          dnorm(x = beta.prop, mean = chain[t, 2], sd = sd[1], log = TRUE) +
          dlnorm(x = sigma.prop, meanlog = log(chain[t, 3]) - 0.5 * sd[2] ^ 2,
                sdlog = sd[2], log = TRUE)
    r <- exp(log.numerator - log.denominator)
    # Generate U(0, 1)
    u <- runif(1, 0, 1)
    # Update
    if(u <= r) {
      chain[t + 1, ] <- c(alpha.prop, beta.prop, sigma.prop)
      accept <- accept + 1
    } else {
      reject <- reject + 1
    chain[t + 1, ] <- chain[t, ]
    }
  }
  list(chain = chain, accept = accept, reject = reject)
}
```

We can execute the Metropolis-Hasting algorithm.

```r
HWA2 <- read.csv("HWA2.csv")
MH <- MHalgorithm(L = 20000, initial = c(1, 1, 1), sd = c(0.1, 0.1),
                  y = HWA2$y, x = HWA2$x)
```

The trace plot is

```r
par(mfrow = c(3, 1), mar = c(4.1, 4.1, 1, 1))
plot(MH$chain[-c(1 : 10000), 1], type = "l")
plot(MH$chain[-c(1 : 10000), 2], type = "l")
plot(MH$chain[-c(1 : 10000), 3], type = "l")
```



(b) Write your own Hamiltonian Monte Carlo algorithm to sample from the posterior distribution of (α, β, σ).
Solution: We need to compute a few gradients in HMC. The posterior is

$$\pi(\theta \mid y) \quad \propto \quad \exp\left\{-\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2\right\} \cdot \pi(\alpha)\,\pi(\beta)\,\pi(\sigma),$$

where

$$
\begin{aligned}
\pi(\alpha) &= \frac{\Gamma(2)}{\sqrt{3\pi}\Gamma(1.5)}\left(1+\frac{\alpha^2}{3}\right)^{-2}, \\
\pi(\beta) &= \frac{\Gamma(2)}{\sqrt{3\pi}\Gamma(1.5)}\left(1+\frac{\beta^2}{3}\right)^{-2}, \\
\pi(\sigma) &= \exp\{-\sigma\}.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\frac{\partial \log \pi(\theta \mid y)}{\partial \alpha} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\alpha-\beta x_i)-\frac{4\alpha}{3+\alpha^2}, \\
\frac{\partial \log \pi(\theta \mid y)}{\partial \beta} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\alpha-\beta x_i)\,x_i-\frac{4\beta}{3+\beta^2}, \\
\frac{\partial \log \pi(\theta \mid y)}{\partial \sigma} &= -\frac{n}{\sigma}+\frac{1}{\sigma^3}\sum_{i=1}^{n}(y_i-\alpha-\beta x_i)^2-1.
\end{aligned}
$$

We can program these functions as

```
posterior <- function(y, x, alpha, beta, sigma, log) {
   out <- c(likelihood(y, x, alpha, beta, sigma, log), prior(alpha, beta, sigma, log))
   if(log == TRUE){
      return(sum(out))
   } else {
      return(prod(out))
   }
}
grad_logposterior <- function(y, x, alpha, beta, sigma){
   out <- matrix(NA, 3, 1)
   error <- y - alpha - beta * x
   ## Gradient: alpha
   out[1, 1] <- sum(error) / (sigma ^ 2) - 4 * alpha / (3 + alpha ^ 2)
   ## Gradient: beta
   out[2, 1] <- sum(error * x) / (sigma ^ 2) - 4 * beta / (3 + beta ^ 2)
   ## Gradient: sigma
   n <- length(y)
   out[3, 1] <- -n / sigma + sum(error ^ 2) / (sigma ^ 3) - 1
   out
}
```

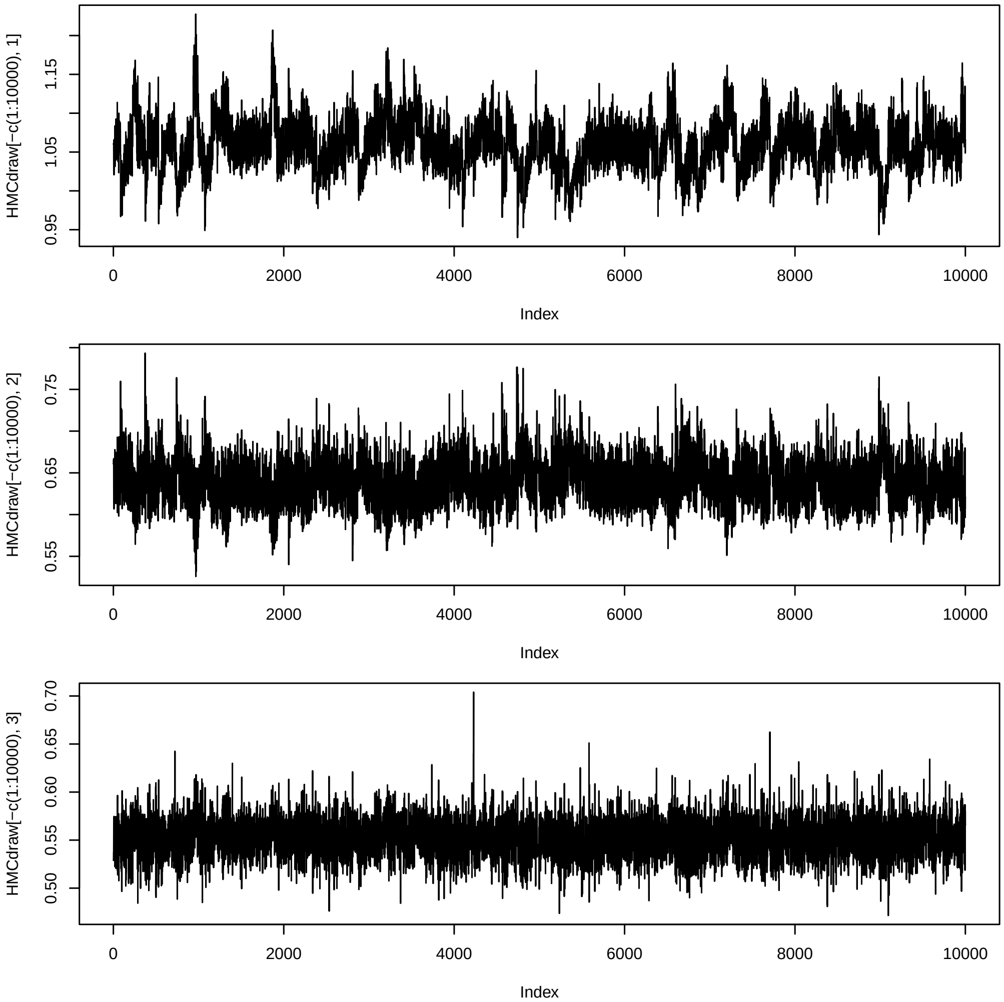
The code for HMC is

```
HMC <- function (n, initial, m, epsilon, L, y, x) {
   # n is the length of the Markov chain, including burn in
   # initial is the initial state
   chain <- rbind(c(initial), matrix(NA, n, length(initial)))
   for(t in 1 : n){
      ## Draw momentum from normal
      phi0 <- rnorm(1, mean = 0, sd = sqrt(m))
      #-----------------------------------#
      # Leapfrog to update theta
      theta <- chain[t, ]
      # Current Hamiltonian
```

```
    H <- -posterior(y = y , x = x, alpha = chain[t, 1], beta = chain[t, 2],
          sigma = chain[t, 3], log = TRUE) -
          sum(dnorm(phi0, mean = 0, sd = sqrt(m), log = TRUE))
    # Make a half step for momentum at the beginning
    phi <- phi0 + epsilon * grad_logposterior(y = y , x = x, alpha = theta[1],
                                   beta = theta[2], sigma = theta[3]) / 2
    # Alternate full steps for position and momentum
    for (i in 1 : L){
      # Make a full step for the position
      theta <- theta + epsilon / m * phi
      # Make a full step for the momentum, except at end of trajectory
      if (i != L) {
        phi <- phi + epsilon * grad_logposterior(y = y , x = x, alpha = theta[1],
                                          beta = theta[2], sigma = theta[3])
      }
    }
    # Make a half step for momentum at the end.
    phi <- phi + epsilon * grad_logposterior(y = y , x = x, alpha = theta[1],
                                        beta = theta[2], sigma = theta[3]) / 2
    #-------------------------------------#
    # Negate momentum at end of trajectory to make the proposal symmetric
    phistar <- -phi
    # Proposed Hamiltonian
    Hstar <- -posterior(y = y , x = x, alpha = theta[1], beta = theta[2],
                   sigma = theta[3], log = TRUE) -
            sum(dnorm(phistar, mean = 0, sd = sqrt(m), log = TRUE))
    # Metropolis ratio
    ratio <- exp(H - Hstar)
    # Accept or reject the state at end of trajectory, returning either
    # the position at the end of the trajectory or the initial position
    if (runif(1) < ratio){
      chain[t + 1, ] <- theta
    } else {
      chain[t + 1, ] <- chain[t, ]
    }
  }
  chain
}
```

We can execute the code and obtain

```
HMCdraw <- HMC(n = 2e04, initial = c(1, 1, 1), m = c(1, 1, 1), epsilon = 0.05,
               L = 10, y = HWA2$y, x = HWA2$x)
## Trace plot
par(mfrow = c(3, 1), mar = c(4.1, 4.1, 1, 1))
plot(HMCdraw[-c(1 : 10000), 1], type = "l")
plot(HMCdraw[-c(1 : 10000), 2], type = "l")
plot(HMCdraw[-c(1 : 10000), 3], type = "l")
```


HMCdraw[−c(1:10000), 1]
0.95
1.05
1.15
0
2000
4000
6000
8000
10000
Index
HMCdraw[−c(1:10000), 2]
0.55
0.65
0.75
0
2000
4000
6000
8000
10000
Index
HMCdraw[−c(1:10000), 3]
0.50
0.55
0.60
0.65
0.70
0
2000
4000
6000
8000
10000
Index