

1

Ans:

(a) $X_1, \dots, X_N \sim \text{Poisson}(\lambda_i), i = 1, \dots, N$. $M \sim U\{1, \dots, N\}$, $\mu_1 \sim \text{Exp}(1)$ and $\mu_2 \sim \text{Exp}(1)$. Let $S_m = \sum_{i=1}^m X_i$, $T = S_N$ and $X = (X_1, \dots, X_N)$. Since,

$$\begin{aligned} \pi(M, \mu_1, \mu_2|X) &\propto f(X|M, \mu_1, \mu_2) \times \pi(M)\pi(\mu_1)\pi(\mu_2) \\ &\propto \mu_1^{S_m} e^{-(m+1)\mu_1} \mu_2^{T-S_m} e^{-(N-m+1)\mu_2}, \end{aligned}$$

and

$$\begin{aligned} m(X) &= \sum_{m=1}^N \int_0^\infty \int_0^\infty \mu_1^{S_m} e^{-(m+1)\mu_1} \mu_2^{T-S_m} e^{-(N-m+1)\mu_2} d\mu_1 d\mu_2 \\ &= \sum_{m=1}^N \frac{\Gamma(S_m + 1)}{(m + 1)^{S_m+1}} \frac{\Gamma(T - S_m + 1)}{(N - m + 1)^{T-S_m+1}}. \end{aligned}$$

Then $\pi(M, \mu_1, \mu_2|X) = \frac{1}{m(X)} \times \mu_1^{S_m} e^{-(m+1)\mu_1} \mu_2^{T-S_m} e^{-(N-m+1)\mu_2}$ where $M \in \{1, \dots, N\}$ and $\mu_1, \mu_2 \in (0, \infty)$.

Hence,

$$\begin{aligned} \pi(M = m|X) &= \int_0^\infty \int_0^\infty \pi(M, \mu_1, \mu_2|X) d\mu_1 d\mu_2 \\ &= \frac{1}{m(X)} \times \frac{\Gamma(S_m + 1)}{(m + 1)^{S_m+1}} \frac{\Gamma(T - S_m + 1)}{(N - m + 1)^{T-S_m+1}}. \end{aligned}$$

(b) The joint posterior distribution for (M, μ_1, μ_2) is $f(\theta|x) \propto \mu_1^{S_m} e^{-m\mu_1} \mu_2^{T-S_m} e^{-(N-m)\mu_2} \times g(\mu_1, \mu_2)$

So M, μ_1, μ_2 have conditional (posterior) distributions

$$f(M|\mu_1, \mu_2, x) \propto \mu_1^{S_m} e^{-m\mu_1} \mu_2^{-S_m} e^{m\mu_2} = \left(\frac{\mu_1}{\mu_2}\right)^{S_m} e^{m(\mu_2-\mu_1)} < \max(e^{N(\mu_2-\mu_1)}, \left(\frac{\mu_1}{\mu_2}\right)^T), M \in \{1, 2, \dots, N\}$$

$$f(\mu_1|M, \mu_2, x) \propto \mu_1^{S_m} e^{-m\mu_1} \times g(\mu_1, \mu_2), \mu_1 > 0$$

$$f(\mu_2|M, \mu_1, x) \propto \mu_2^{T-S_m} e^{-(N-m)\mu_2} \times g(\mu_1, \mu_2), \mu_2 > 0$$

If I know how to sample from these conditional distributions, then I could use the SSS algorithm.

Algorithm (SSS):

Step 1. Given initial value $(m^{(0)}, \mu_1^{(0)}, \mu_2^{(0)})$, and $j = 0$.

Step 2. Generate $M^{(j+1)}$ from $f(M|\mu_1^{(j)}, \mu_2^{(j)}, x)$.

Since the density $f(M|\mu_1, \mu_2, x)$ is bounded, I could use the rejection method with $M^* \sim \text{Unif}(1, \dots, N)$ to generate from it.

Step 3. Generate $\mu_1^{(j+1)}$ from $f(\mu_1|M^{(j+1)}, \mu_2^{(j)}, x)$.

Step 4. Generate $\mu_2^{(j+1)}$ from $f(\mu_2|M^{(j+1)}, \mu_1^{(j+1)}, x)$.

Since the form of $g(\mu_1, \mu_2)$ is unknown, I may choose 2 "standard distributions" like exponential which has the same or larger support for (μ_1, μ_2) to be the "proposal" distributions and then use Metropolis-Hastings algorithm to generate data from each of these conditional distributions.

Step 5. Set $\theta^{(j+1)} = (M^{(j+1)}, \mu_1^{(j+1)}, \mu_2^{(j+1)})$, and $j = j + 1$.

Repeat step 2 to 5 to get N_1 samples, and then for every N_2 samples to collect N samples.

2

Ans:

Let $\mathbf{X} = (X_1, \dots, X_5) \sim f_{\mathbf{X}} \propto h(\mathbf{x}) = \exp[-(x_1 - x_2)^2 - (x_2 - x_3)^2 - (x_3 - x_4)^2 - (x_4 - x_5)^2 - (x_5 - x_1)^2]$ and let $\mathbf{U} = (U_1, U_2, \dots, U_5)$ and $U_i \stackrel{iid}{\sim} U(0, 1)$. $P(\mathbf{U} = \mathbf{u}) = 1$ for $0 \leq u_i \leq 1, i = 1, \dots, 5$.

(a) Rejection algorithm

$0 < \frac{h(\mathbf{u})}{P(\mathbf{u})} = h(\mathbf{u}) \leq e^{(0)} = 1 \equiv M$, when $0 \leq u_i \leq 1, i = 1, \dots, 5$.

Algorithm:

Step 1. Generate $x_i^{(*)}, i = 1, \dots, 5$ from 5 independent Uniform(0, 1) .

Step 2. Generate $u \sim U(0, 1)$.

Step 3. If $\frac{h(\mathbf{x}^{(*)})}{P(\mathbf{x}^{(*)})} = h(\mathbf{x}^{(*)}) \geq u$, set $\mathbf{x} = \mathbf{x}^{(*)}$. Otherwise, return to Step 1.

Repeat step 1 to 3 to get N samples.

(b) Gibbs Sampler / Successive Substitution Sampling(SSS) algorithm

The conditional distribution of $x_1|X_2, \dots, X_5$ is

$$f(X_1|X_2, \dots, X_5) \propto h(x_1, \dots, x_5) \propto N\left(\frac{x_2 + x_5}{2}, \frac{1}{4}\right)I_{[0,1]}(x_1),$$

and $f(X_2|X_1, X_3, X_4, X_5)$, $f(X_3|X_1, X_2, X_4, X_5)$, $f(X_4|X_1, X_2, X_3, X_5)$, and $f(X_5|X_1, X_2, X_3, X_4)$ have similar conclusions.

Algorithm:

Step 1. Start at initial point $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_5^{(0)})$, $0 \leq x_i^{(0)} \leq 1$.

Step 2. For $j = 0, 1, 2, \dots, N$. Do

sample $x_1^{(j+1)}$ from $N\left(\frac{x_2^{(j)} + x_5^{(j)}}{2}, \frac{1}{4}\right)I_{[0,1]}(x_1)$.

sample $x_2^{(j+1)}$ from $N\left(\frac{x_1^{(j+1)} + x_3^{(j)}}{2}, \frac{1}{4}\right)I_{[0,1]}(x_2)$.

sample $x_3^{(j+1)}$ from $N\left(\frac{x_2^{(j+1)} + x_4^{(j)}}{2}, \frac{1}{4}\right)I_{[0,1]}(x_3)$.

sample $x_4^{(j+1)}$ from $N\left(\frac{x_3^{(j+1)} + x_5^{(j)}}{2}, \frac{1}{4}\right)I_{[0,1]}(x_4)$.

sample $x_5^{(j+1)}$ from $N\left(\frac{x_1^{(j+1)} + x_4^{(j+1)}}{2}, \frac{1}{4}\right)I_{[0,1]}(x_5)$.

Step 3. Set $\mathbf{x}^{(j+1)} = (x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_4^{(j+1)}, x_5^{(j+1)})$, and set $j = j + 1$.

Repeat step 2 to 3 to get N_1 samples, and thin for every N_2 samples to collect N samples. The $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ could be viewed as (dependent) sample from $f_{\mathbf{X}}$ and could be used to approximate properties of $h(\mathbf{x})$.

(c) Gibbs / M-H hybrid algorithm

$f(X_1|X_2, X_3, X_4, X_5) \propto \exp(-2[x_1 - \frac{x_2+x_5}{2}]^2)I_{[0,1]}(x_1)$ and let $J^{(j+1)}(x_i^{(*)}|x_i^{(j)}) = 1$, for $0 \leq x_i^{(*)} \leq 1$.

Algorithm:

Step 1. Start at initial point $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_5^{(0)})$, $0 \leq x_i^{(0)} \leq 1$.

Step 2. For $j = 0, 1, 2, \dots, N$. Do

- generate $x_i^{(*)}$ from $U(0, 1)$ for $i = 1, \dots, 5$.
- calculate $r_1^{(j+1)} = \frac{f(x_1^{(*)}|x_2^{(j)}, x_3^{(j)}, x_4^{(j)}, x_5^{(j)})}{f(x_1^{(j)}|x_2^{(j)}, x_3^{(j)}, x_4^{(j)}, x_5^{(j)})}$, then generate $Y_1^{(j)} \sim \text{Bernoulli}(\min(r_1^{(j+1)}, 1))$,
and set $x_1^{(j+1)} = Y_1^{(j)} x_1^{(*)} + (1 - Y_1^{(j)}) x_1^{(j)}$,
- calculate $r_2^{(j+1)} = \frac{f(x_2^{(*)}|x_1^{(j+1)}, x_3^{(j)}, x_4^{(j)}, x_5^{(j)})}{f(x_2^{(j)}|x_1^{(j+1)}, x_3^{(j)}, x_4^{(j)}, x_5^{(j)})}$, then generate $Y_2^{(j)} \sim \text{Bernoulli}(\min(r_2^{(j+1)}, 1))$,
and set $x_2^{(j+1)} = Y_2^{(j)} x_2^{(*)} + (1 - Y_2^{(j)}) x_2^{(j)}$.
- calculate $r_3^{(j+1)} = \frac{f(x_3^{(*)}|x_1^{(j+1)}, x_2^{(j+1)}, x_4^{(j)}, x_5^{(j)})}{f(x_3^{(j)}|x_1^{(j+1)}, x_2^{(j+1)}, x_4^{(j)}, x_5^{(j)})}$, then generate $Y_3^{(j)} \sim \text{Bernoulli}(\min(r_3^{(j+1)}, 1))$,
and set $x_3^{(j+1)} = Y_3^{(j)} x_3^{(*)} + (1 - Y_3^{(j)}) x_3^{(j)}$,
- calculate $r_4^{(j+1)} = \frac{f(x_4^{(*)}|x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_5^{(j)})}{f(x_4^{(j)}|x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_5^{(j)})}$, then generate $Y_4^{(j)} \sim \text{Bernoulli}(\min(r_4^{(j+1)}, 1))$,
and set $x_4^{(j+1)} = Y_4^{(j)} x_4^{(*)} + (1 - Y_4^{(j)}) x_4^{(j)}$,
- calculate $r_5^{(j+1)} = \frac{f(x_5^{(*)}|x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_4^{(j+1)})}{f(x_5^{(j)}|x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_4^{(j+1)})}$, then generate $Y_5^{(j)} \sim \text{Bernoulli}(\min(r_5^{(j+1)}, 1))$,
set $x_5^{(j+1)} = Y_5^{(j)} x_5^{(*)} + (1 - Y_5^{(j)}) x_5^{(j)}$.

Step 3. Set $\mathbf{x}^{(j+1)} = (x_1^{(j+1)}, x_2^{(j+1)}, x_3^{(j+1)}, x_4^{(j+1)}, x_5^{(j+1)})$, and set $j = j + 1$.

Repeat step 2 to 3 to get N_1 samples, and thin for every N_2 samples to collect N samples. The $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$ could be viewed as (dependent) sample from $f_{\mathbf{X}}$ and could be used to approximate properties of $h(\mathbf{x})$.

(d) Metropolis-Hastings algorithm

$$h(\mathbf{x}) = \exp(-(x_1 - x_2)^2 - (x_2 - x_3)^2 - (x_3 - x_4)^2 - (x_4 - x_5)^2 - (x_5 - x_1)^2) \text{ and } J^{(j+1)}(\mathbf{x}^{(*)}|\mathbf{x}^{(j)}) = 1$$

Algorithm:

Step 1. Start at initial point $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_5^{(0)})$ for $0 \leq x_i^{(0)} \leq 1$.

Step 2. For $j = 0, \dots, N$. Do

- select candidate $\mathbf{x}^{(*)} = (x_1^{(*)}, \dots, x_5^{(*)})$, $X_i^{(*)} \sim U(0, 1)$,
- calculate $r^{(j)} = \frac{h(\mathbf{x}^{(*)})/J^{(j)}(\mathbf{x}^{(*)}|\mathbf{x}^{(j-1)})}{h(\mathbf{x}^{(j-1)})/J^{(j)}(\mathbf{x}^{(j-1)}|\mathbf{x}^{(*)})} = \frac{h(\mathbf{x}^{(*)})}{h(\mathbf{x}^{(j-1)})}$,
- generate $Y^{(j)} \sim \text{Bernoulli}(\min(r^{(j)}, 1))$,
- take $\mathbf{x}^{(j)} = Y^{(j)} \mathbf{x}^{(*)} + (1 - Y^{(j)}) \mathbf{x}^{(j-1)}$, and set $j = j + 1$.

Repeat step 2 to get N_1 samples, and then for every N_2 samples to collect N samples. The $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$ could be viewed as (dependent) sample from $f_{\mathbf{X}}$ and could be used to approximate properties of $h(\mathbf{x})$.

3

Ans:

$X \sim N(50\mu, 50), Y \sim N(M\mu, M), X, Y \in \mathbb{R}$, and X, Y are independent. $\mu \sim N(0, (1000)^2), M - 1 \sim \text{Poisson}(100), \mu \in \mathbb{R}, M \in \mathbb{N}$, and M, μ are independent.

The posterior distribution of $M, \mu | X, Y$ is

$$f(M, \mu | X, Y) \propto f(X, Y | M, \mu) \times f(M, \mu) = f(X | \mu) \times f(Y | M, \mu) \times f(\mu) \times f(M)$$

$$\propto e^{-\frac{(x-50\mu)^2}{2 \times 50}} \times \frac{1}{\sqrt{m}} e^{-\frac{(y-m\mu)^2}{2m}} \times e^{-\frac{\mu^2}{2 \times 1000^2}} \times \frac{1}{(m-1)!} 100^{(m-1)}$$

Then the conditional distributions are

$$f(\mu | M, X, Y) \propto e^{-\frac{(x-50\mu)^2}{2 \times 50} - \frac{(y-m\mu)^2}{2m} - \frac{\mu^2}{2 \times 1000^2}} \propto N\left(\frac{y+x}{50+m+10^{-6}}, \frac{1}{50+m+10^{-6}}\right), \mu \in \mathbb{R} \text{ and}$$

$$f(M | \mu, X, Y) \propto \frac{1}{\sqrt{m(m-1)!}} e^{-\frac{(y-m\mu)^2}{2m}} \times 100^{(m-1)}$$

The first one is a normal density and the second one could be generated by the M-H algorithm with $J^{(j+1)}(M^{(*)} | M^{(j)}) = f(m) \propto \frac{1}{(m-1)!} 100^{(m-1)}$

Then $r = \exp\left(-\frac{(y-m^{(*)}\mu^{(*)})^2}{2m^{(*)}} + \frac{(y-m^{(j)}\mu^{(j)})^2}{2m^{(j)}}\right) \times \sqrt{\frac{m^{(j)}-1}{m^{(*)}}}$

Algorithm:

Step 1. Start at initial point $(M^{(0)}, \mu^{(0)}) \in \mathbb{N} \times \mathbb{R}$, and $j = 0$.

Step 2. Generate $\mu^{(j+1)} \sim N\left(\frac{y+x}{50+m^j+10^{-6}}, \frac{1}{50+m^j+10^{-6}}\right)$

Step 3. Generate $M^{(*)} \sim \text{Poisson}(100)$, and compute r .

Step 4. Generate $u_1 \sim U(0, 1)$. If $u_1 \leq \min(r, 1)$, then set $M^{(j+1)} = M^{(*)}$.

Otherwise, $M^{(j+1)} = M^{(j)}$, and set $j = j + 1$.

Repeat step 2 to 4 to get N samples. The posterior mean of M could be approximated by $\frac{1}{N} \sum_{i=1}^N M^{(i)}$.

4

Ans:

(a) A rejection algorithm.

Use $X^{(*)}, Y^{(*)} \stackrel{iid}{\sim} \text{Exp}(1)$, then $P(x, y) = e^{-x-y}$ for $0 \leq x$ and $0 \leq y$. and $0 < \frac{h(x, y)}{P(x, y)} = \sin^2(x)e^{-xy} \leq 1 \equiv M$.

Algorithm:

Step 1. Generate $u_1, u_2 \stackrel{iid}{\sim} U(0, 1)$, then $X^{(*)} = -\log(u_1)$ and $Y^{(*)} = -\log(u_2)$.

Step 2. Generate $u \sim U(0, 1)$. If $\frac{h(X^{(*)}, Y^{(*)})}{P(X^{(*)}, Y^{(*)})} \geq u$, set $(X, Y) = (X^{(*)}, Y^{(*)})$. Otherwise, return to step 1.

Repeat step 1 to 2 to collect N samples.

(b) A Gibbs algorithm

$h(x|y) = \frac{h(x,y)}{\int h(x,y)dx} \propto \sin^2(x)\exp(-(1+y)x) = h'(x|y), x > 0$. To update x using a rejection algorithm, let $P(x|y) = (1+y)e - (1+y)x$ which implies $\frac{h'(x|y)}{P(x|y)} \leq (1+y)^{-1} \equiv M$.
 $h(y|x) = \frac{h(x,y)}{\int h(x,y)dy} = (1+x)\exp(-(1+x)y) \sim \exp(1+x), y > 0$.

Algorithm:

- Step 1. Start at initial point $(X^{(0)}, Y^{(0)})$, and set $j = 0$.
- Step 2. Sample $u_1, u_2 \stackrel{iid}{\sim} U(0, 1)$, then set $X^{(*)} = -\frac{\log(u_1)}{(1+Y^j)}$.
- Step 3. If $\frac{h'(X^{(*)}|Y^{(j)})}{P(X^{(*)}|Y^{(j)})} \geq Mu_2$, then set $X^{(j+1)} = X^{(*)}$, else goto step 2.
- Step 4. Sample $u \sim U(0, 1)$, set $Y^{(j+1)} = -\frac{\log(u)}{1+X^{(j+1)}}$.
- Step 5. Set $j = j + 1$, goto step 2.

Repeat step 2 to 5 to get N_1 samples, and thin for every N_2 samples to collect N samples.

(c) A hybrid Gibbs/M-H algorithm

Let $J^{(j+1)}(x^{(*)}|x^{(j)}) = e^{-x^{(j)}}$.

Algorithm:

- Step 1. Start at initial point $(X^{(0)}, Y^{(0)})$, and set $j = 0$.
- Step 2. Sample $u \sim U(0, 1)$, then $X^{(*)} = -\log(u)$.
- Step 3. Calculate $r^{(j+1)} = \frac{h(x^{(*)}|(x^{(j)}, y^{(j)}))/J^{(j+1)}(x^{(*)}|x^{(j)})}{h(x^{(j)}|(x^{(j)}, y^{(j)}))/J^{(j+1)}(x^{(j)}|x^{(*)})} = \frac{\sin^2(x^{(*)})(\exp(-y^{(j)}(x^{(*)}-x^{(j)})))}{\sin^2(x^{(j)})}$.
- Step 4. Generate $Y^{(j+1)} \sim \text{Bernoulli}(\min(r^{(j+1)}, 1))$, Set $x^{(j+1)} = Y^{(j+1)}x^{*} + (1 - Y^{(j+1)})x^{(j)}$.
- Step 5. Sample $u \sim U(0, 1)$, set $Y^{(j+1)} = -\frac{\log(u)}{1+X^{(j+1)}}$.
- Step 6. Set $j = j + 1$, goto step 2.

Repeat step 2 to 6 to get N_1 samples, and thin for every N_2 samples to collect N samples.

(d) A M-H algorithm

Let $J^{(j+1)}(x^{(*)}, y^{(*)}|x^{(j)}, y^{(j)}) = e^{-x^{(j)}-y^{(j)}}$.

Algorithm:

- Step 1. Start at initial point $(X^{(0)}, Y^{(0)})$, and set $j = 0$.
- Step 2. Sample $u_1, u_2 \stackrel{iid}{\sim} U(0, 1)$, and set $X^{(*)} = -\log(u_1), Y^{(*)} = -\log(u_2)$.
- Step 3. Calculate $r^{(j+1)} = \frac{h(x^{(*)})/J^{(j+1)}(x^{(*)}|x^{(j)})}{h(x^{(j)})/J^{(j+1)}(x^{(j)}|x^{(*)})} = \frac{\sin^2(X^{(*)})\exp(-X^{(*)}Y^{(*)})}{\sin^2(X^{(j)})\exp(-X^{(j)}Y^{(j)})}$.
- Step 4. Generate $Y^{(j+1)} \sim \text{Bernoulli}(\min(r^{(j+1)}, 1))$.
- Step 5. Set $x^{(j+1)} = Y^{(j+1)}x^{(*)} + (1 - Y^{(j+1)})x^{(j)}$.

Repeat step 2 to 5 to get N_1 samples, and thin for every N_2 samples to collect N samples.

.....

5

Ans:

(a)

$X_1, X_2, \dots, X_5 \stackrel{iid}{\sim} N(\mu, \sigma^2)$. $\bar{x} = 3, s^2 = 0.5, t_{0.972,4} = 2.7765$.

95% two-sided confidence limits for μ : [2.1220, 3.8780].

95% two-sided confidence limits for σ^2 : [0.1795, 4.1287].

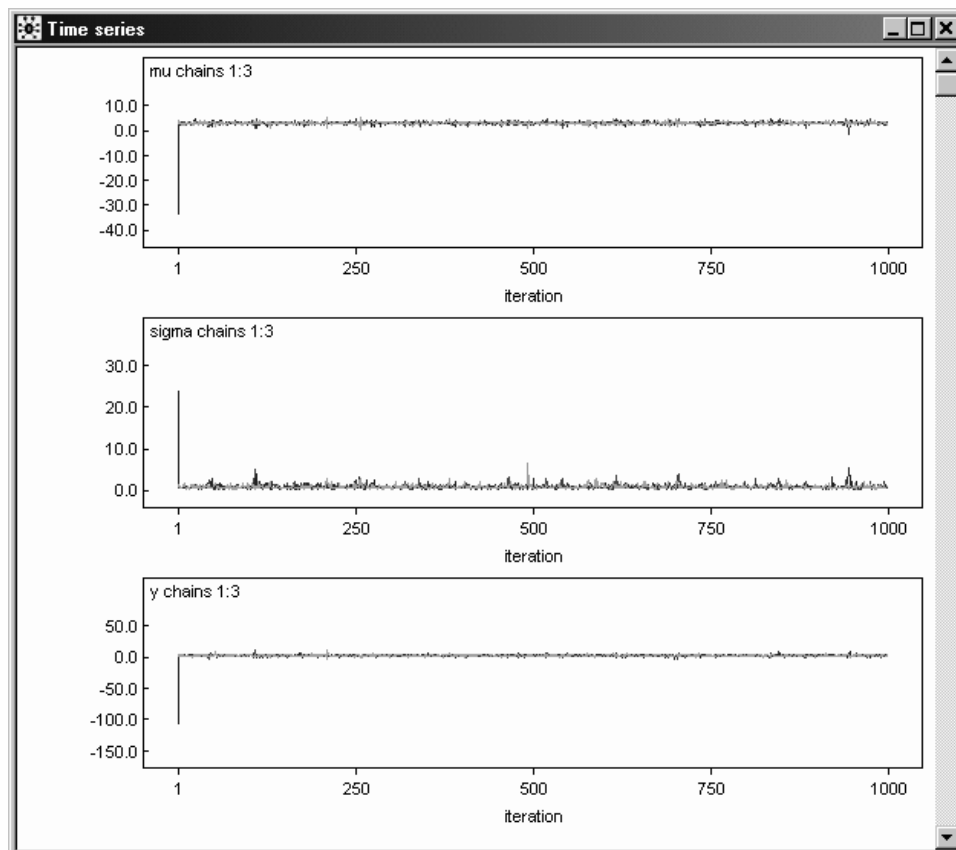
95% two-sided confidence limits for σ : [0.4237, 2.0319].

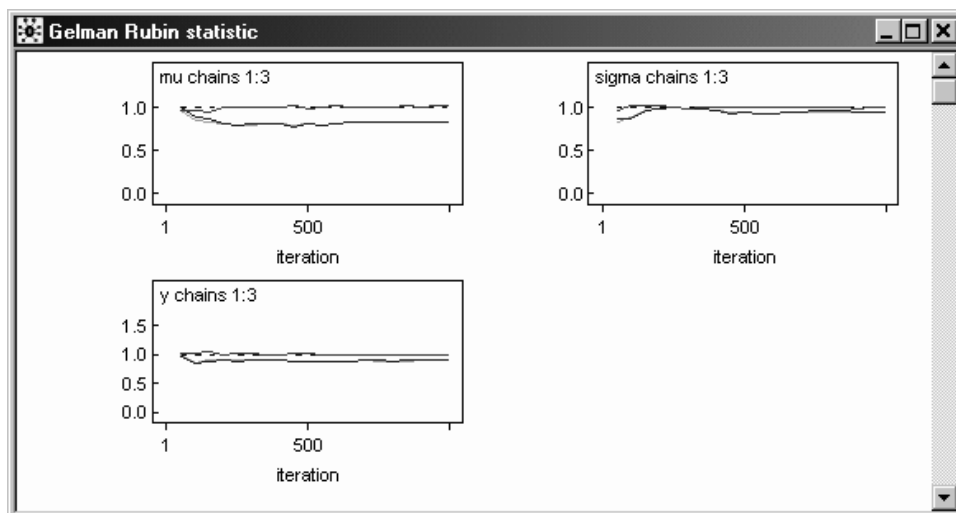
95% two-sided prediction limits for an additional value from this normal distribution: [0.8494, 5.1506].

(b) Bayes analyses

(i) The "history" plot and "bgr diag" plot of the first 1,000 iterations

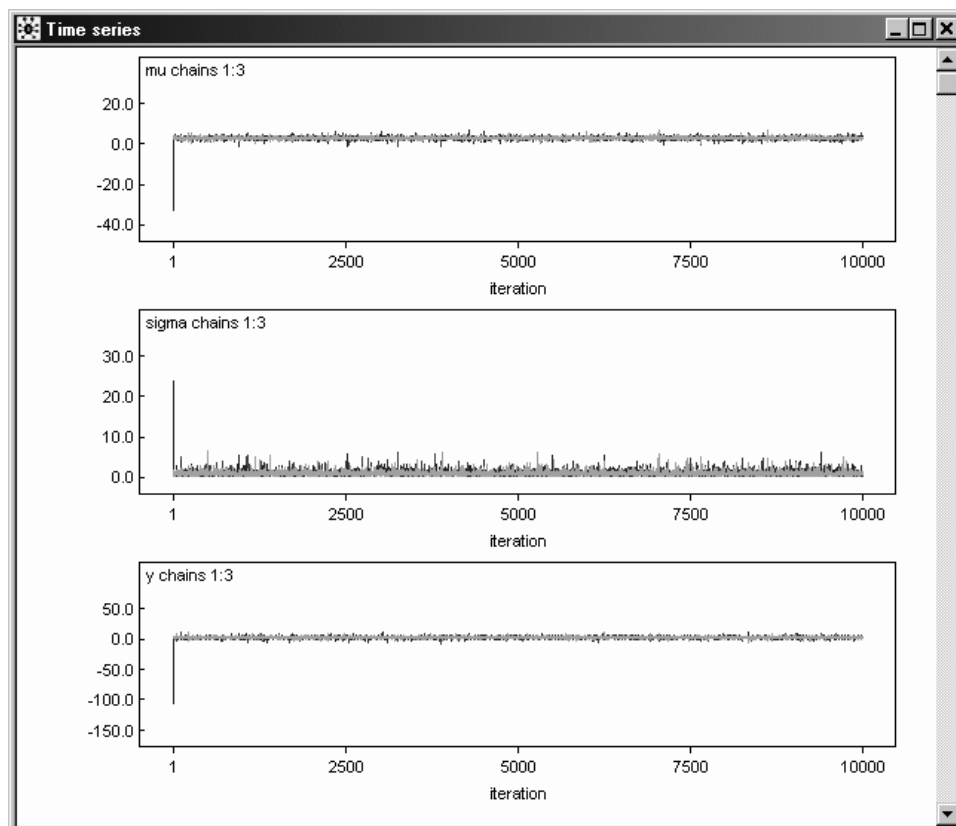
From the "history" plot, the simulated data seem to converge, but from the "bgr diag" plot shows that the chains do not appear yet adequate as a burn-in. (The red line is close to 1, but the blue and green lines are not stable, some chains may be not converge yet.)

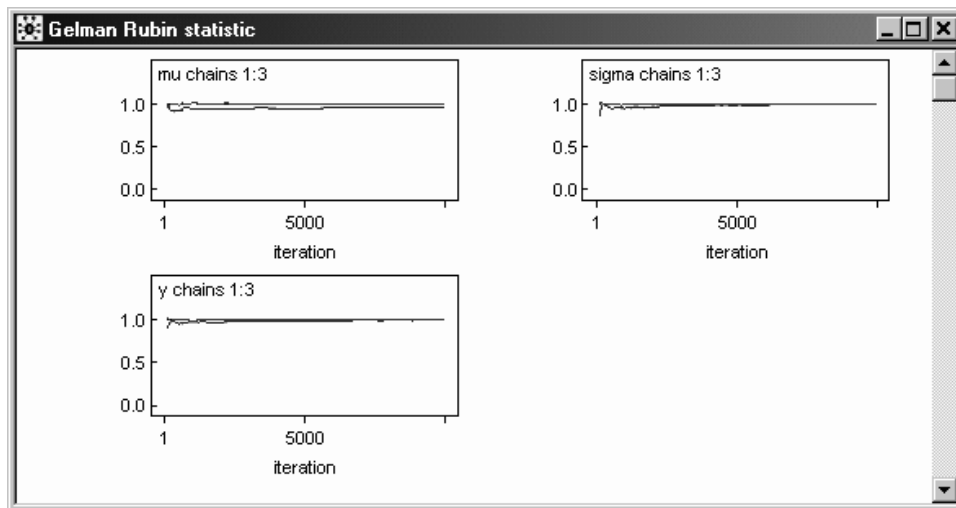




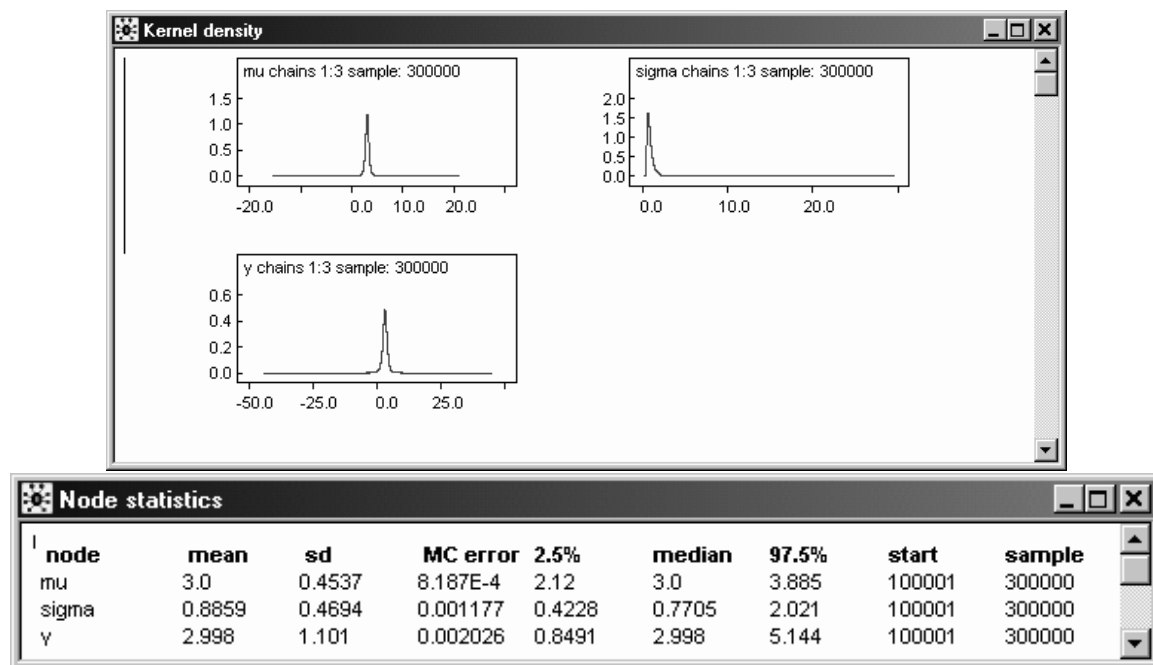
(ii) The "history" plot and "bgr diag" plot of the first 10,000 iterations

From the "history" plot, the simulated data seem to converge, and from the "bgr diag" plot shows that the chains have appeared adequate as a burn-in.





(iii) The "density" plot and "stats" results for samples from the 100,000 iterations after 100,000 updated and appered as a "burn-in".



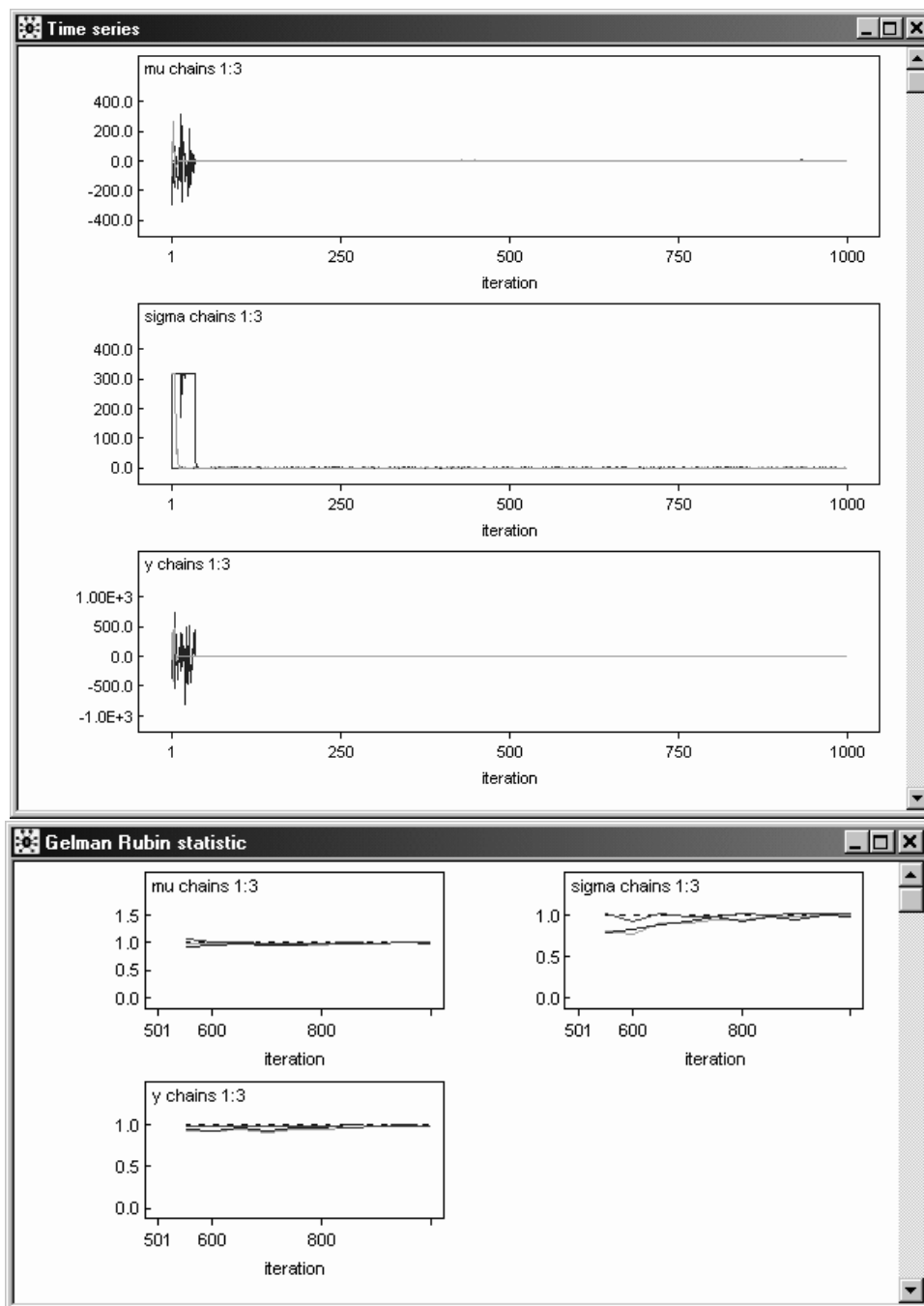
The 95% credible intervals are from the "stats" results compared with the 95% confidence intervals from (a). They are close.

	95% credible interval	95% confidence interval
μ	[2.12, 3.885]	[2.1220, 3.8780]
σ	[0.4228, 2.021]	[0.4237, 2.0319]
y	[0.8491, 5.144]	[0.8494, 5.1506]

(c) Bayes analyses

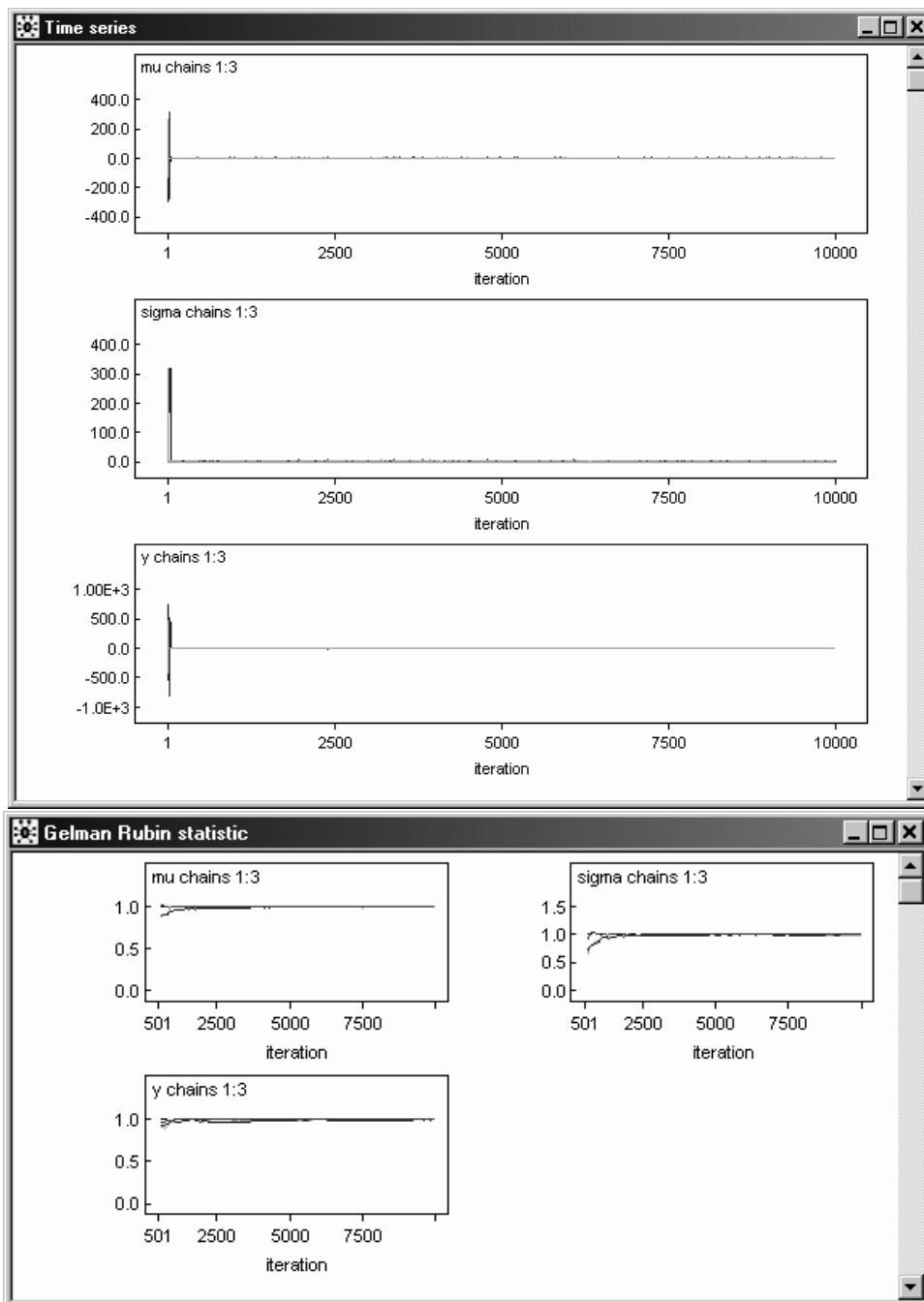
(i) The first 1,000 iterations

From the "history" plot, the simulated data seem to converge. From the "bgr diag" plot shows that the chains do not appear yet adequate as a burn-in, but it shows that they are in convergence. (The red line is close to 1, but the blue and green lines are not stable, some chains may be not converge yet.)

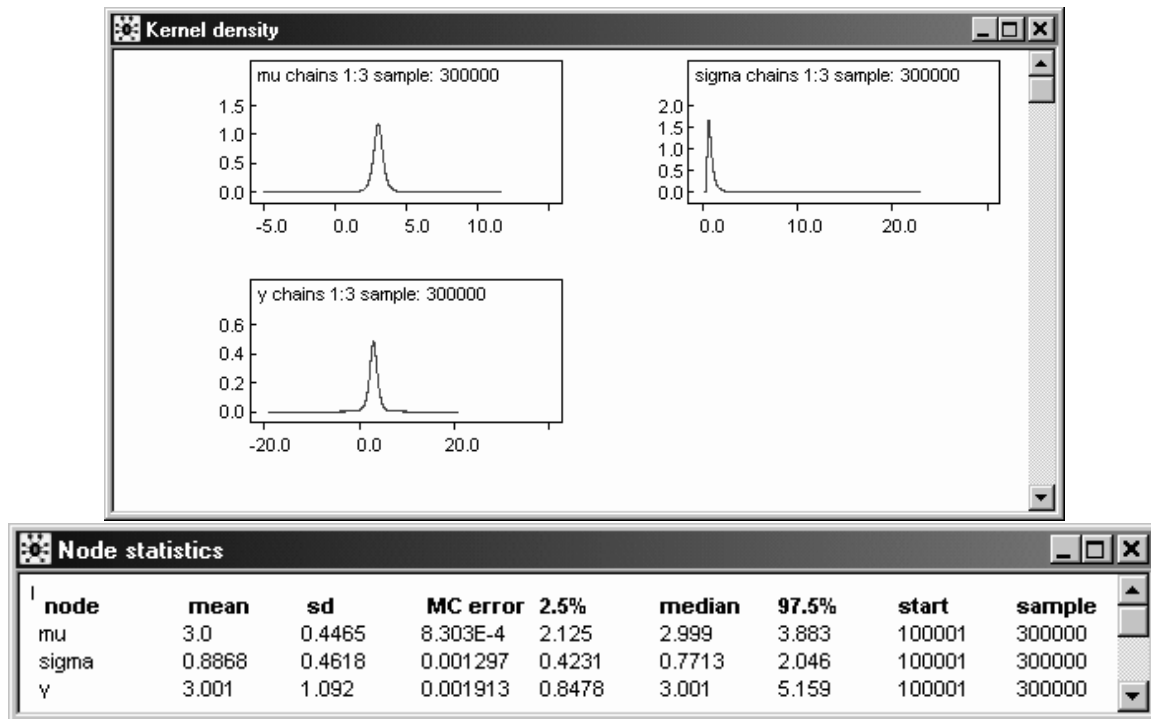


(ii) The first 10,000 iterations

From the "history" plot, the simulated data seem to converge, and from the "bgr diag" plot shows that the chains have appeared adequate as a burn-in.



(iii) The "density" plot and "stats" results for samples from the 100,000 iterations after 100,000 updated and appered as a "burn-in".



The 95% credible intervals are from the "stats" results compared with the 95% confidence intervals from (a). They are close.

	95% credible interval	95% confidence interval
μ	[2.125, 3.883]	[2.1220, 3.8780]
σ	[0.4231, 2.046]	[0.4237, 2.0319]
y	[0.8478, 5.159]	[0.8494, 5.1506]

(d) Using a "diffuse" proper prior for (μ, σ^2) , e.g. $\mu \sim N(0, 1/0.0001)$ and $\sigma^2 \sim G(1, 100)$, for inference of μ , σ and y , the results should be robust enough, and also close to parts (a), (b), and (c).

node	mean	sd	MCerror	2.5%	median	97.5%	start	sample
mu	2.998	0.4456	0.002769	2.106	2.998	3.881	100000	27000
sigma	0.8875	0.4635	0.00284	0.4231	0.7721	2.048	100000	27000
y	3.0	1.094	0.007197	0.8499	3.006	5.171	100000	27000

```

model {
for (i in 1:5) { x[i]~dnorm(mu, tau) }
y~dnorm(mu,tau)
mu~dnorm(0,0.0001)
tau~dgamma(0.01,0.01)
sigma<-sqrt(1/tau) }

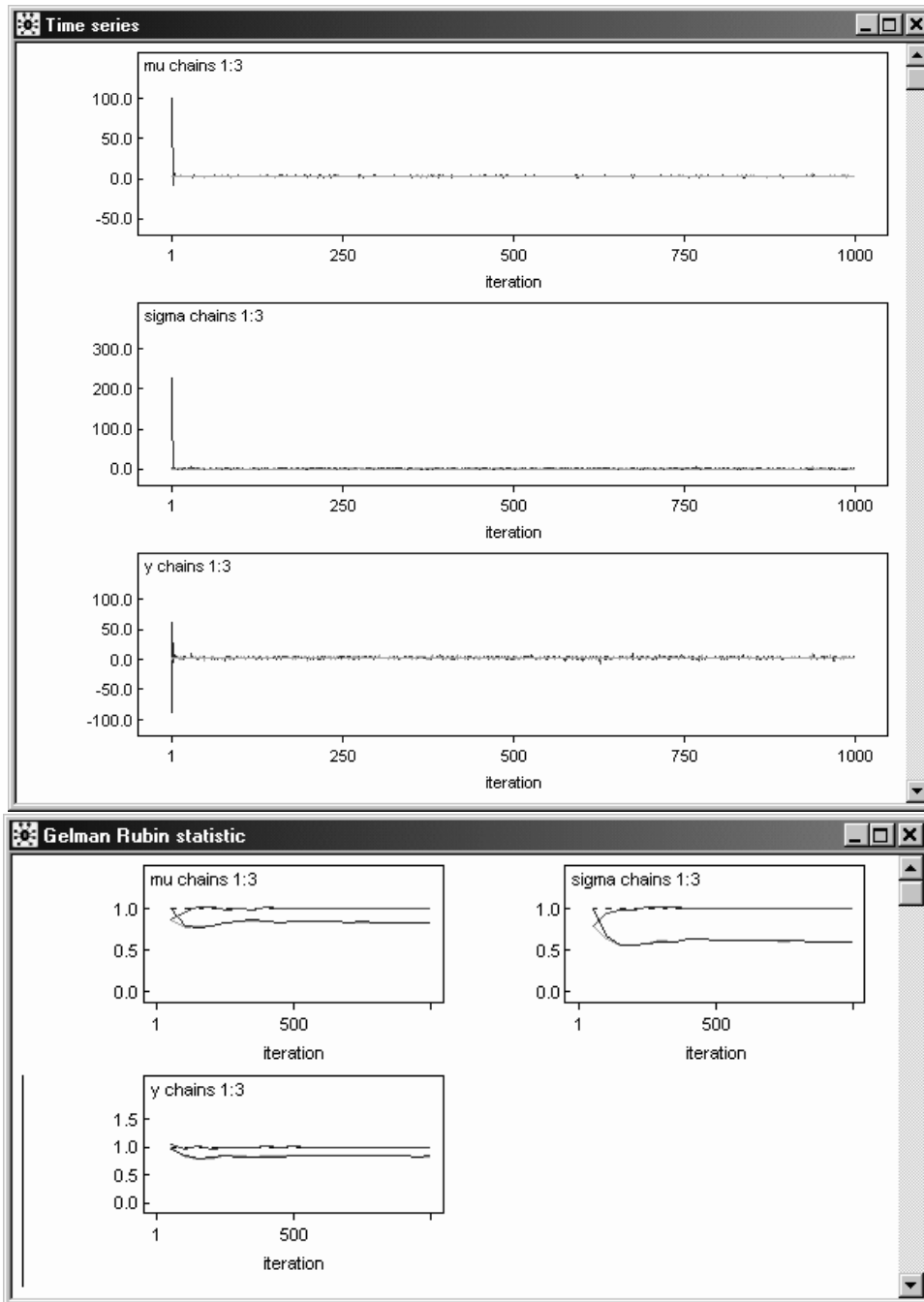
```

6 Censoring

Ans:

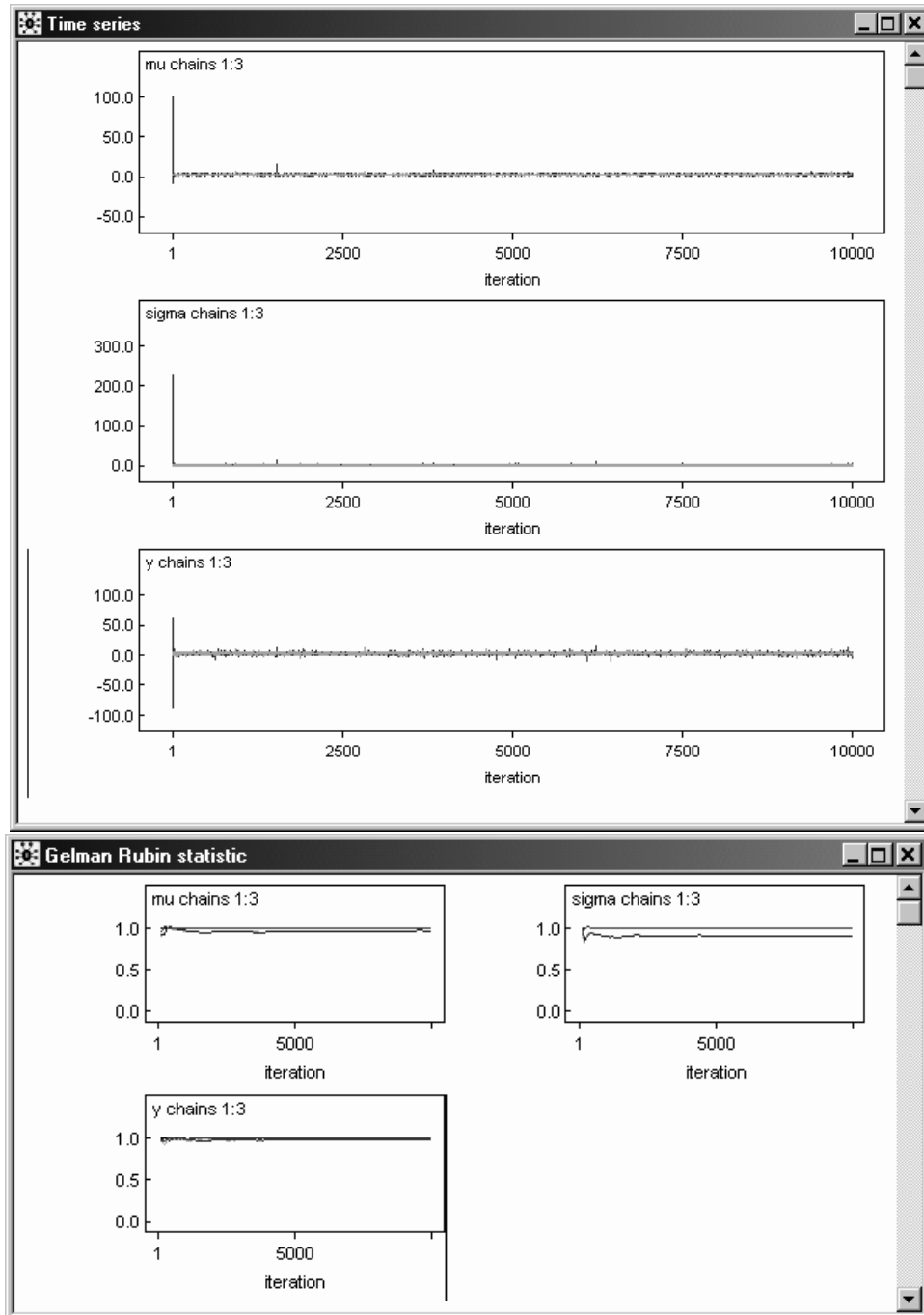
(i) The first 1,000 iterations

From the "history" plot, the simulated data seem to converge, but from the "bgr diag" plot shows that the chains do not appear yet adequate as a burn-in. (The red line is close to 1, but the blue and green lines are not stable, some chains may be not converge yet.)

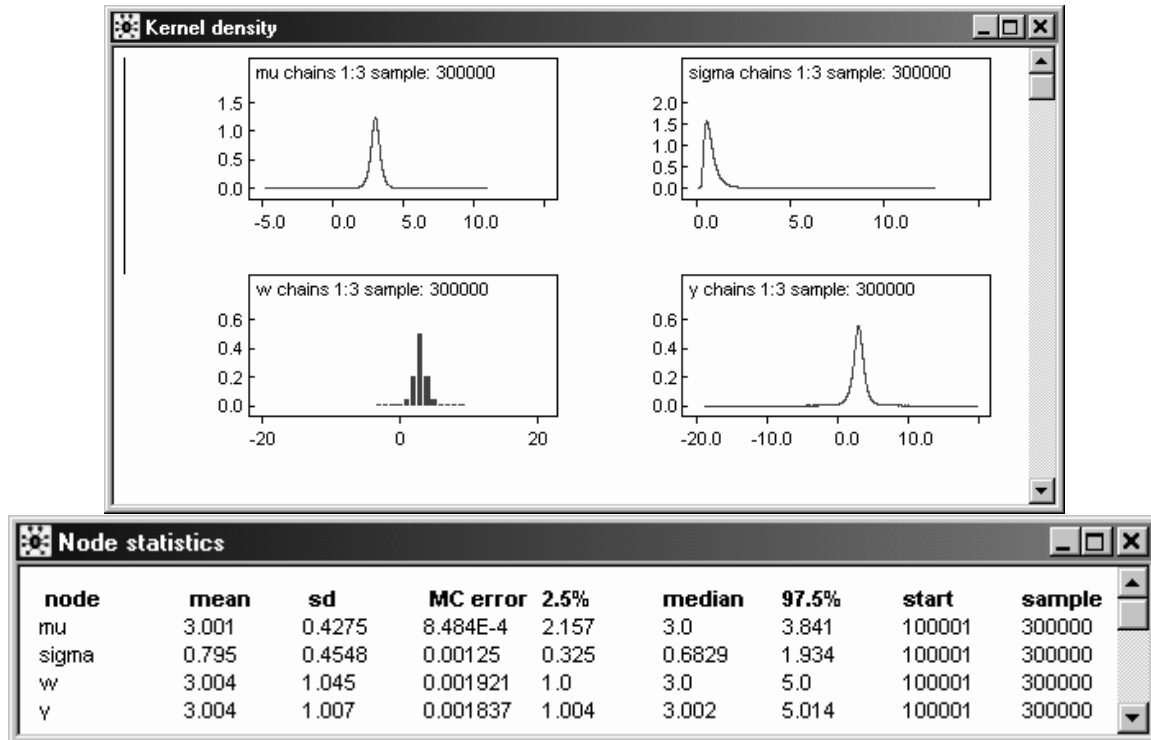


(ii) The first 10,000 iterations

From the "history" plot, the simulated data seem to converge, but from the "bgr diag" plot shows that the chains of σ do not appear yet adequate as a burn-in. (The red line is close to 1, but the blue and green lines are not stable, some chains may be not converge yet.)



(iii) The "density" plot and "stats" results for samples from the 100,000 iterations of each 3 chains after 100,000 updated and appered as a "burn-in".



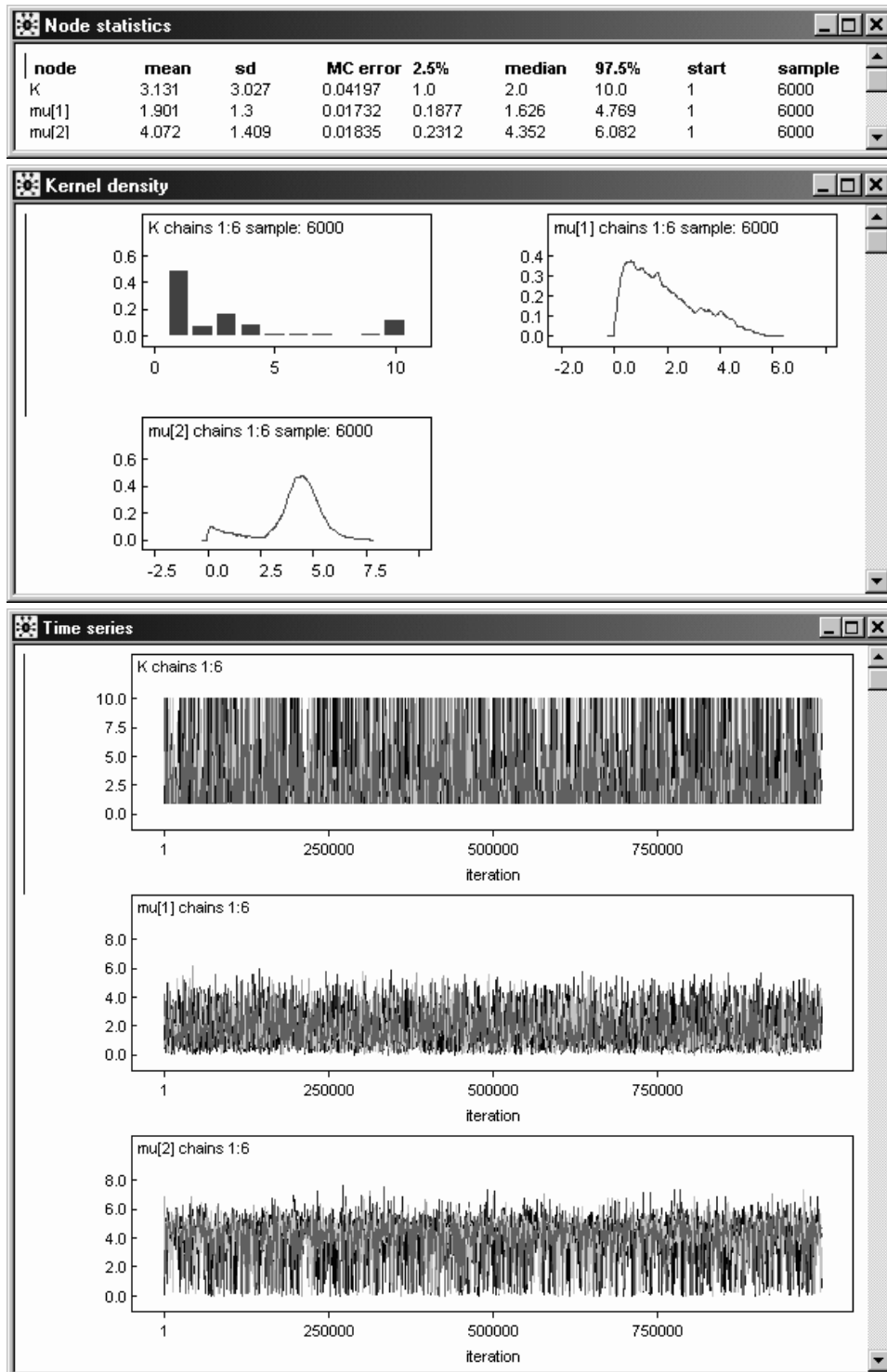
The 95% credible intervals are from the "stats" results compared with the 95% credible intervals from 5(b) and the 95% confidence intervals from (a). They are close, and it also shows that σ has smaller interval in this problem.

	95% credible interval	95% credible interval 5(b)	95% confidence interval
μ	[2.157, 3.841]	[2.12, 3.885]	[2.1220, 3.8780]
σ	[0.325, 1.934]	[0.4228, 2.021]	[0.4237, 2.0319]
y	[1.004, 5.014]	[0.8491, 5.144]	[0.8494, 5.1506]

7 Consider the context of Problem #1

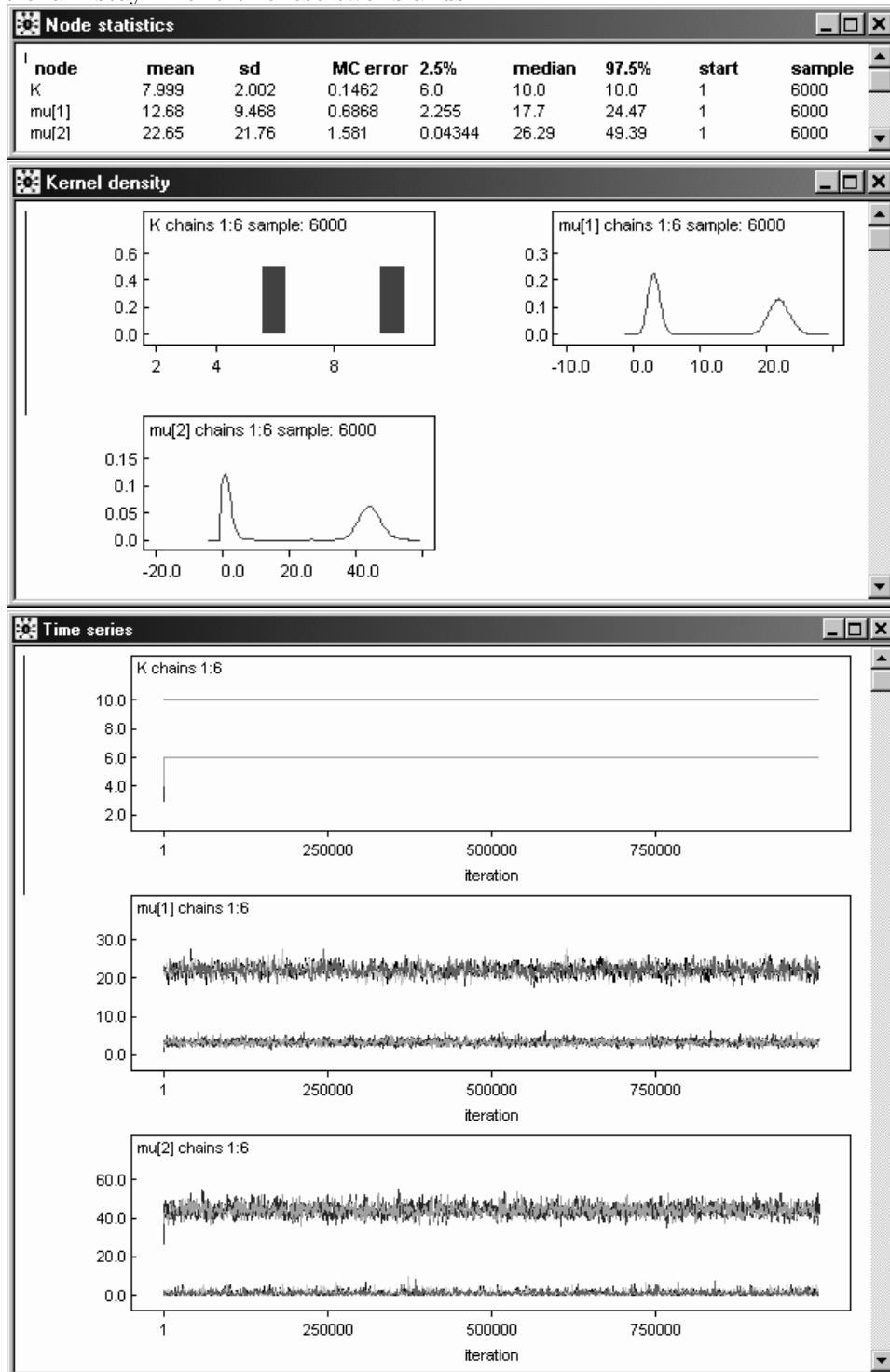
(a) Data set 1

Ans: Use 6 chains, update with 1 million iterations, and thin to every 1000th iteration. The estimated posterior mean for the changing position is $K = 3.131$, and the posterior mean of the rate λ are $\mu[1] = 1.901$ and $\mu[2] = 4.072$. The picks 1 and 10 in the histogram of K do not close to truth.

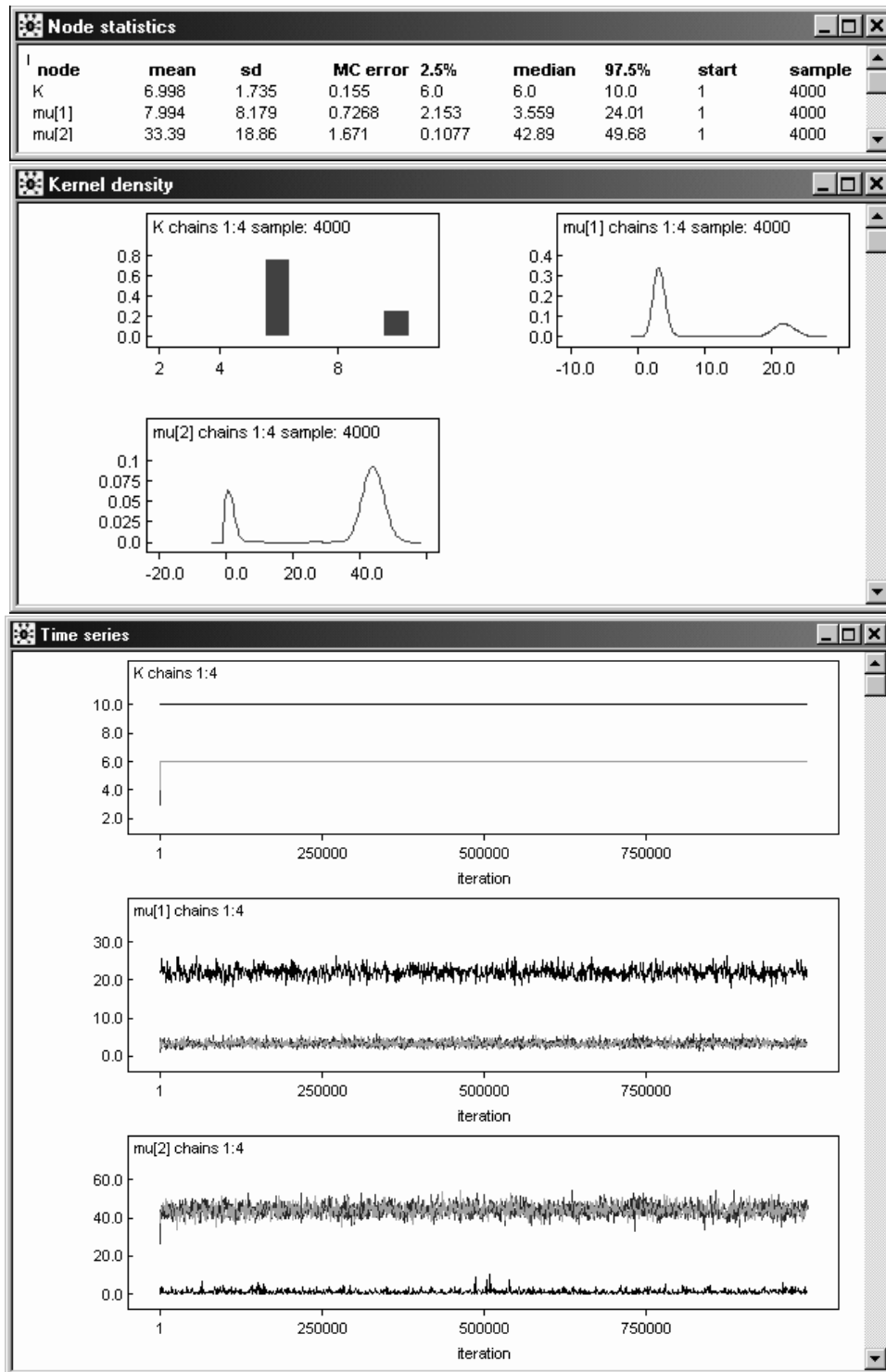


(b) Data set 2

Ans: Use 6 chains, update with 1 million iterations, and thin to every 1000th iteration. The estimated posterior mean for the changing position is $K = 7.999$ and the posterior mean of the rate λ are $\mu[1] = 12.66$ and $\mu[2] = 22.65$. In the histogram of K , the two pikes at 6 and 10 correspond to 2 picks in the distributions of μ 's. There seem to have 2 island in the joint distribution and the initial values of K will cause the whole chain stay in one of these two islands.



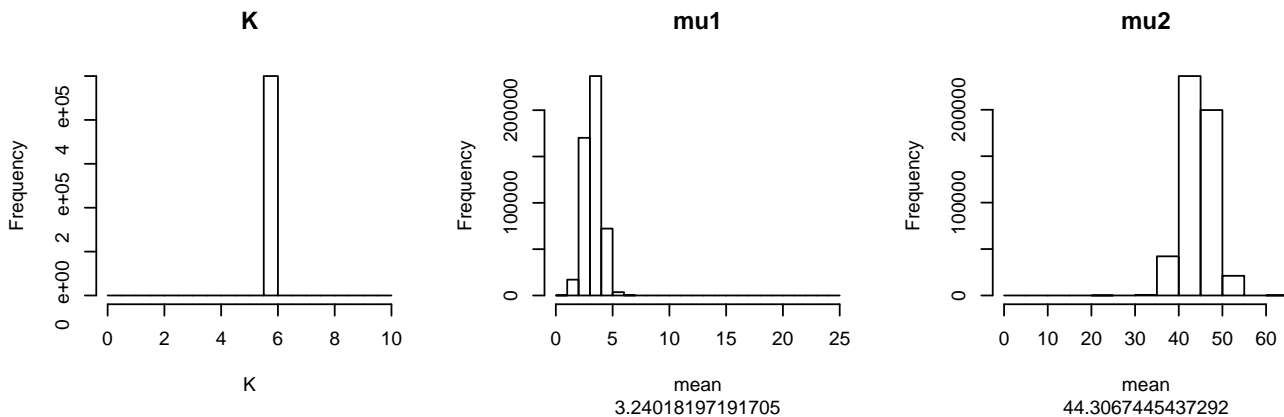
The stats, density and history results from WinBUGS. Use the first 4 chains, update with 1 million iterations, and thin to every 1000th iteration. The results of using the first 4 initial shows that the estimated distribution depends on the initial values. Both approximated distribution is not true.



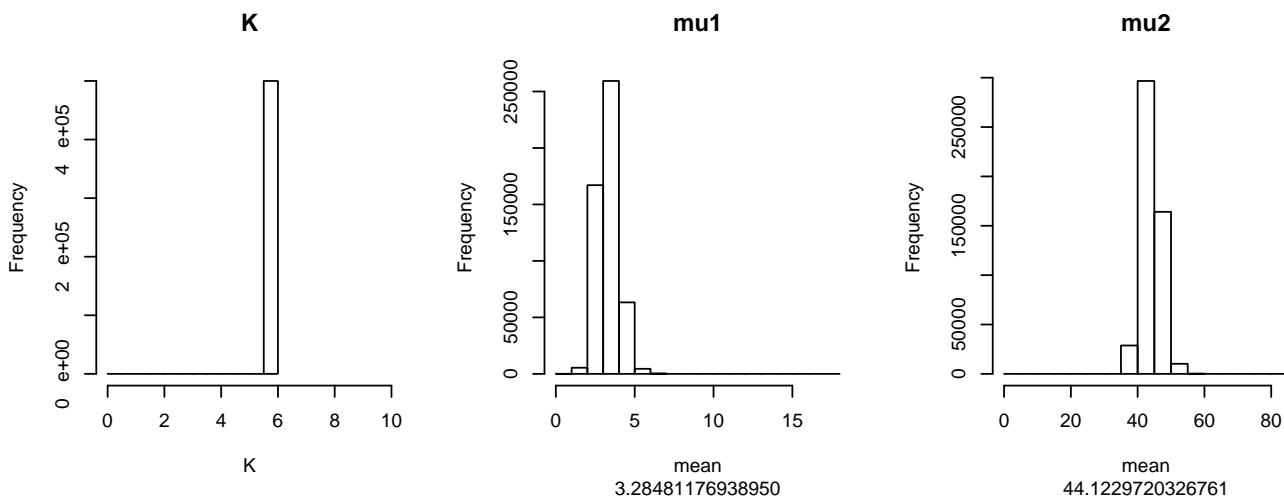
(c) Use a Metropolis-Hastings algorithm

The following are histograms for K , $\mu[1]$ and $\mu[2]$ computed from a M-H algorithm for data set 2 in part (b). They are closer to true values and the chains won't be stuck in an island.

For initial value (10, 25, 1)



For initial value (1, 25, 1)



The ratio of the posterior probability that $M=10$ to the posterior probability that $M=6$

And the ratio of the posterior probability that $M = 10$ to the posterior probability that $M = 6$ is

$$\begin{aligned} \frac{\pi(M = 10|X)}{\pi(M = 6|X)} &= \frac{\Gamma(S_{10} + 1)}{(10 + 1)^{S_{10}+1}} \frac{\Gamma(T - S_{10} + 1)}{(10 - 10 + 1)^{T-S_{10}+1}} \\ &= \frac{\Gamma(S_6 + 1)}{(6 + 1)^{S_6+1}} \frac{\Gamma(T - S_6 + 1)}{(10 - 6 + 1)^{T-S_6+1}} \\ &= \left(\frac{\Gamma(T + 1)}{11^{T+1}} \right) \bigg/ \left(\frac{\Gamma(S_6 + 1)}{7^{S_6+1}} \frac{\Gamma(T - S_6 + 1)}{5^{T-S_6+1}} \right) \end{aligned}$$

For part (a), $T = S_{10} = 44$, $S_6 = 22$ and the ratio is 9.4180. For part (b), $T = S_{10} = 242$, $S_6 = 22$ and the ratio is 6.561×10^{-49} .