# Markov chain Monte Carlo (MCMC)

Ricardo Ehlers
ehlers@icmc.usp.br

Departamento de Matemática Aplicada e Estatística
Universidade de São Paulo

Ch. 6-10 in Robert & Casella
Ch. 4-6 in Gamerman & Lopes
Ch. 11-12 in Gelman et. al.

# Introduction to Markov chains

**Definition**

A stochastic process $X_0, X_1, \ldots$ is a Markov chain if, given the present state, past and future states are independent.

In terms of probability,

$$P(X_{t+1} = y | X_t = x, x_{t-1}, \ldots, x_0) = P(X_{t+1} = y | X_t = x).$$

## Objective

Generate values from a distribution $\pi(\cdot)$ by simulating a Markov chain.

- The chain must be homogeneous, irreducible and ergodic with stationary distribution given by $\pi(\cdot)$.
- Given realizations $\{\mathbf{X}^{(t)}, t = 0, 1, \dots\}$ from such a Markov chain, then

$$X^{(t)} \quad \stackrel{t\to\infty}{\longrightarrow} \quad \pi(x)$$

$$\frac{1}{n}\sum_{t=1}^{n} g(X_i^{(t)}) \quad \stackrel{n\to\infty}{\longrightarrow} \quad \mathbb{E}_\pi(g(X_i))$$

- The chain is dependent by definition.
- The sample average of the simulated values is a consistent estimator of the theoretical mean.
- Irreducible chain: there is a positive probability of reaching any point from any other point in a finite number of iterations.
- Aperiodic chain: does not reach the same point with any fixed regularity.

We need to define a transition kernel for the Markov chain.

- In the discrete case, define $P(X_{t+1} = y | X_t = x)$.
- In the continuous case, if $K(\cdot|x)$ is the transition kernel define,

$$P(X \in A|x) = \int_A K(y|x)dy.$$

We shall consider Markov chains for which the transition kernel is the conditional density of $X_{t+1}|X_t$.

**Example.** A sequence of random variables $\{X_t, t = 0, 1, \dots\}$ is said to be a random walk if,

$$X_{t+1} = X_t + \epsilon_t,$$

with $\epsilon_t$ independent of $X_t, X_{t-1}, \dots$

Markovian property,

$$P(X_{t+1} \in A | x_0, x_1, \dots, x_t) = P(X_{t+1} \in A | x_t)$$

## Recurrence

In a Markov chain with finite state space, a state is said *recurrent* if the mean number of visits is infinite. Otherwise, the state is said *transient*.

- The chain is *recurrent* if the mean number of visits to any set $A$ é infinito.
- In MCMC we are interested in recurrent chains, which explore the whole state space.

## Harris Recurrence

If besides being recurrent the chain is also *Harris recurrent* then,

$$P(X_t \in A | x_0) = 1$$

So, starting at any initial point $(x_0)$ the stationary distribution is the same.

In practice MCMC methods require,

- Specifying the transition kernel of the chain.
- Choosing the initial value of the chain.
- Monitoring the convergence. How can we decide whether the chain has reached equilibrium?
- The resulting chain needs to be homogeneous, irreducible and aperiodic.

# Important Concepts

# Stationary distribution

**Definition**

A distribution $\pi$ is said to be a stationary distribution of a chain with transition probabilities $P(x, y)$ if,

$$\sum_x \pi(x)P(x, y) = \pi(x), \ \forall y.$$

Equivalently, in matrix notation $\pi P = \pi$.

The idea is that, once the chain reaches $\pi$ then all subsequent distributions are $\pi$ which is also known as the equilibrium distribution.

# Ergodicity

**Definition**

A chain is said to be geometrically ergodic if there exist a number $0 \leq \lambda < 1$ and a real integrable function $M(x)$ such that,

$$||P^n(x, \cdot) - \pi(\cdot)|| \leq M(x)\lambda^n, \ \forall x.$$

- The smallest $\lambda$ satisfying this condition is called the rate of convergence.
- If $M(x)$ assumes a very large value this might slow down convergence considerably.
- In particular If $M(x) = M$ the ergodicity is uniform.

## Ergodic theorem

For a real valued function $g(X)$, if the Markov chain is ergodic and $\mathbb{E}_\pi(g(X)) < \infty$ for the unique limiting distribution $\pi$ then,

$$\frac{1}{n}\sum_{t=1}^{n} g(X^{(t)}) \quad \stackrel{n\to\infty}{\longrightarrow} \quad \mathbb{E}_\pi(g(X))$$

almost surely.

- This provides a Markov chain equivalent of the law of large numbers.
- So, averages of chain values provide strongly consistent estimates in the limiting distribution $\pi$ despite their dependence.

# Inefficiency Factor (IF)

What is the variance of this estimator?

How are the correlations taken into account?

Let $\bar{g}_n = \sum_{t=1}^{n} g(X^{(t)})/n$ be a sequence of estimators of $\mathbb{E}_\pi(g(X))$ and denote its variance by

$$Var_\pi(\bar{g}_n) = \frac{\tau_n^2}{n}.$$

It can be shown that,

$$\tau_n^2 = \sigma^2 \left[1 + 2\sum_{k=1}^{n}\left(1 - \frac{k}{n}\right)\rho_k\right] \quad \overset{n\to\infty}{\longrightarrow} \quad \sigma^2\left(1 + 2\sum_{k=1}^{\infty}\rho_k\right) = \tau^2,$$

where $\sigma^2 = Var(g(X))$ and the term between parentheses is called inefficiency factor or integrated autocorrelation time.

# Effective Sample Size (ESS)

How far is the Markov chain sample from an independent sample?

**Definition**

The effective sample size is defined as,

$$n_{eff} = \frac{n}{1 + 2\sum_{k=1}^{\infty} \rho_k}.$$

Consequently,

$$Var_{\pi}(\bar{g}_n) = \frac{\sigma^2}{n_{eff}}.$$

Note also that under independent sampling $n_{eff} = n$.

# A Central Limit Theorem for Markov chains

If a Markov chain is uniformly (geometrically) ergodic and $E[g^2] < \infty$ then,

$$\frac{\bar{g}_n - \mathbb{E}_\pi(g(X))}{\tau/\sqrt{n}} \quad \xrightarrow{d} \quad N(0,1)$$

as $n \to \infty$.

# Reversibility

**Definition**

For a homogeneous Markov chain with transition probabilities $P(x, y)$ and stationary distribution $\pi$, the reversibility condition is usually written as,

$$\pi(x)P(x, y) = \pi(y)P(y, x), \ \forall x, y.$$

So, a system in equilibrium moves from $x$ to $y$ at the same rate as it moves from $y$ to $x$.

This is referred to as detailed balance equation.

# Simulation using Markov chains

# Gibbs Sampler

The Gibbs sampler is a Markov chain where the transitions between states are performed according to the complete conditional distributions.

For $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$,

$$\pi(\theta_i|\boldsymbol{\theta}_{-i}) = \frac{\pi(\boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta}) d\theta_i} \propto \pi(\boldsymbol{\theta}).$$

where,

$$\boldsymbol{\theta}_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_d)'.$$

- Instead of generating values from the joint distribution $\pi(\boldsymbol{\theta})$, the values are generated from these conditional distributions.
- $\pi(\theta_i|\boldsymbol{\theta}_{-i})$ is obtained considering only those terms in the joint distribution that do not depend on $\theta_i$.
- Is generating values from $\pi(\theta_i|\boldsymbol{\theta}_{-i})$, $i = 1, \ldots, d$ equivalent to generate values from $\pi(\boldsymbol{\theta})$ ?

1. initialize the iterations counter for the chain $t = 0$;
2. specify the initial values, $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_d^{(0)})'$;
3. obtain a new value of $\boldsymbol{\theta}^{(t)}$ from $\boldsymbol{\theta}^{(t-1)}$ by sequentially generating values,

$$\begin{aligned}
\theta_1^{(t)} &\sim \pi(\theta_1 | \theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)}) \\
\theta_2^{(t)} &\sim \pi(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)}) \\
&\vdots \\
\theta_d^{(t)} &\sim \pi(\theta_d | \theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_{d-1}^{(t)})
\end{aligned}$$

4. Change counter $t$ to $t + 1$ and return to step 3 until convergence.

- Generating values from the complete conditional distributions is enough to recover the joint distribution.
- The sequence $\boldsymbol{\theta}^{(t)}$, $t = 1, 2, \ldots$ is a Markov chain with invariant distribution $\pi(\boldsymbol{\theta})$.
- Each sequence $\theta_j^{(t)}$, $t = 1, 2, \ldots,$ $j = 1, \ldots, d$ is a Markov chain with invariant distribution $\pi_j(\theta_j)$.
- The components of $\boldsymbol{\theta}$ can be scalars, vectors or matrices.

**Example.** Suppose a random vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ has a joint density $\pi(\boldsymbol{\theta})$ which admits the following simplification,

$$\pi(\boldsymbol{\theta}) \propto p(\theta_1|\theta_2)p(\theta_2)p(\theta_3).$$

where $p(\theta_1|\theta_2)$, $p(\theta_2)$ and $p(\theta_3)$ are known.

The complete conditional densities are,

$$\begin{aligned} p(\theta_1|\theta_2, \theta_3) &\propto p(\theta_1|\theta_2) \\ p(\theta_2|\theta_1, \theta_3) &\propto p(\theta_1|\theta_2)p(\theta_2) \\ p(\theta_3|\theta_1, \theta_2) &\propto p(\theta_3) \end{aligned}$$

**Example.** Let $X$ and $Y$ continuous random variables with joint density function,

$$f(x,y) = kx^4 \exp(-x(2+y)),\ x > 0, y > 0.$$

The complete conditional densities are,

$$f(y|x) \propto kx^4 \exp(-x(2+y)) \propto \exp(-xy), y > 0.$$

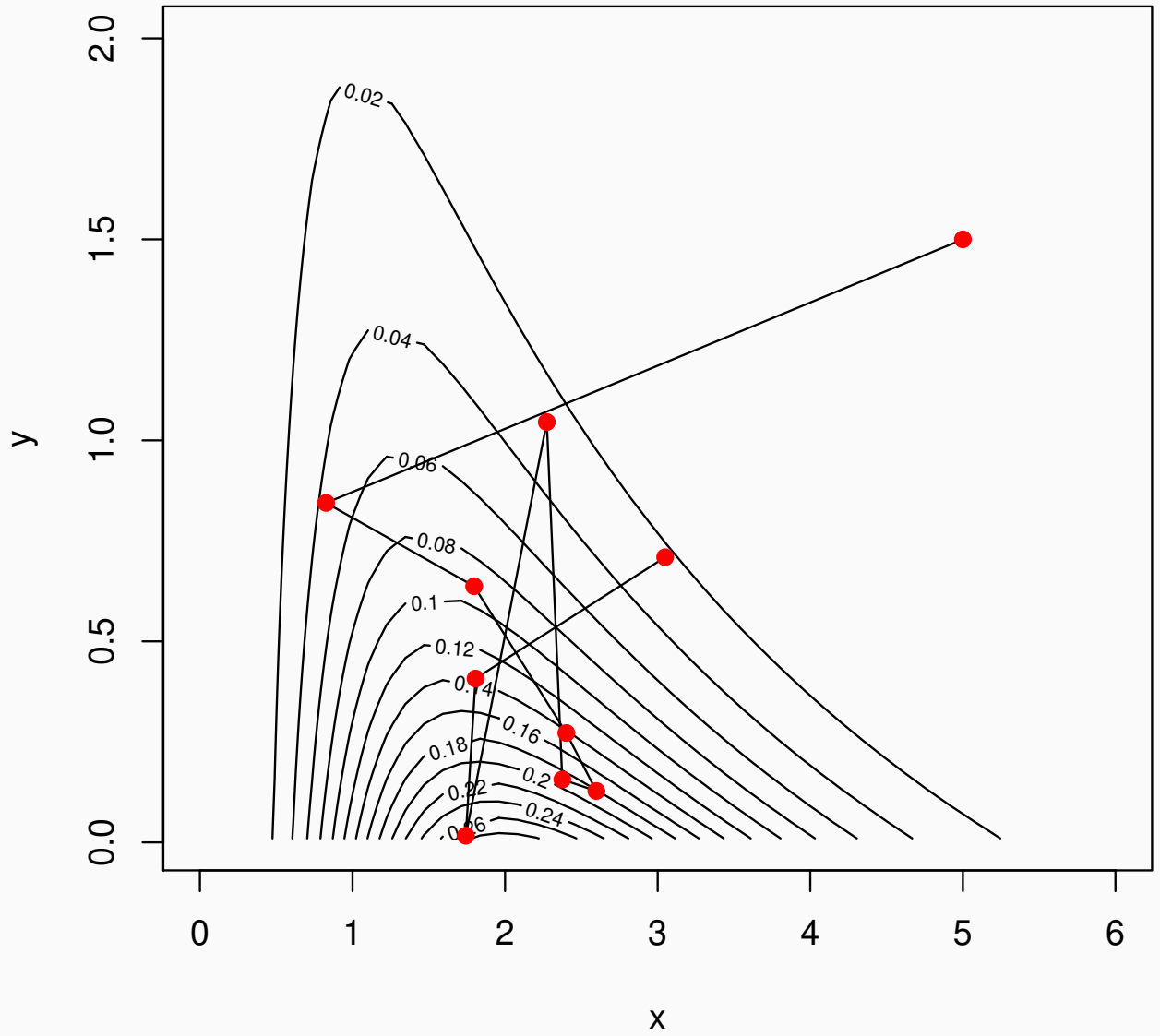
$$f(x|y) \propto kx^4 \exp(-x(2+y)) \propto x^4 \exp(-x(2+y)),\ x > 0.$$

Then,

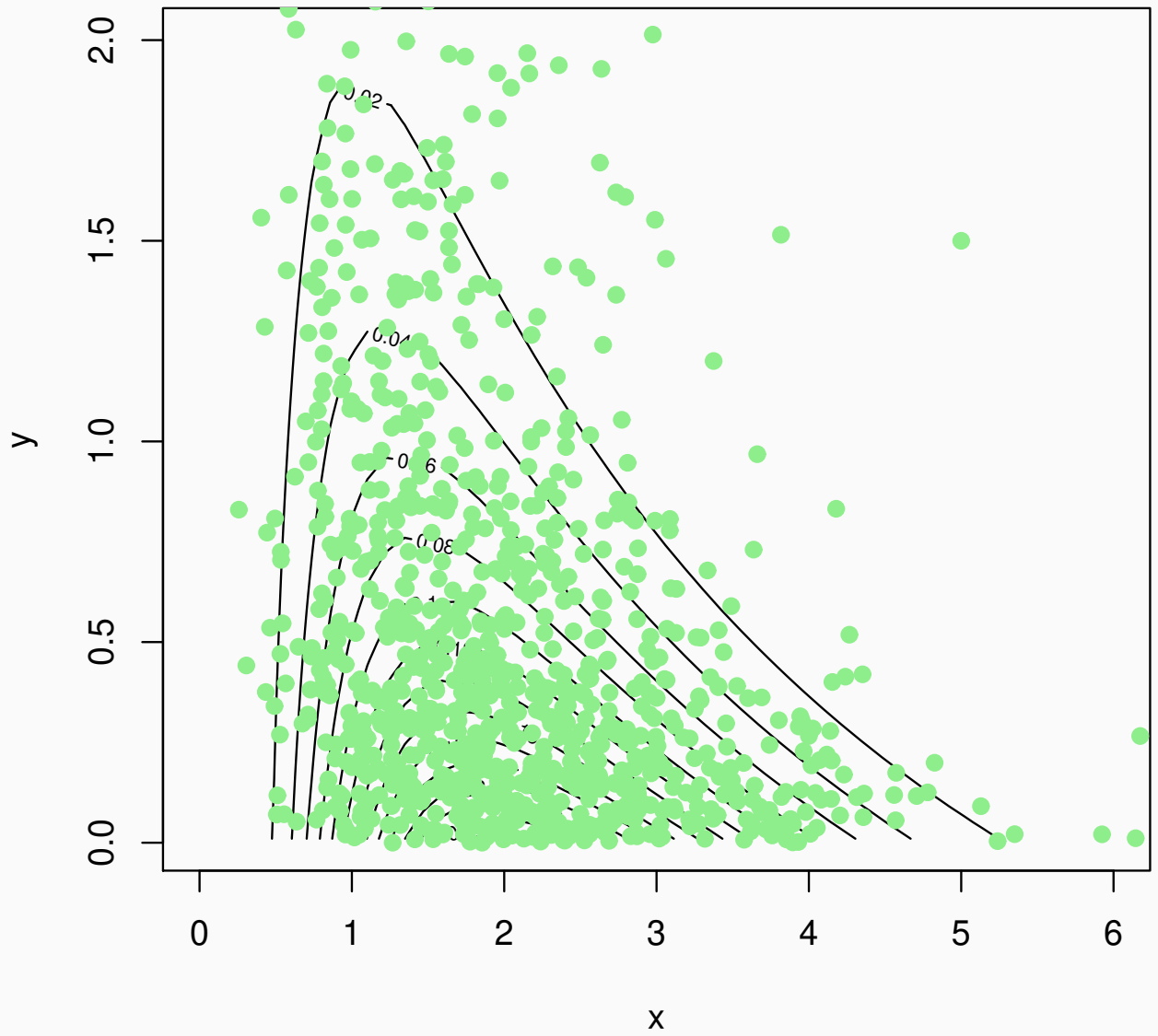
$$\begin{aligned} Y|X = x &\sim \text{Exp}(x), \\ X|Y = y &\sim \text{Gamma}(5, 2+y) \end{aligned}$$

```
> gibbs <- function(x0,y0,niter=1000) {
+     x = y = rep(0, niter)
+     x[1] = x0
+     y[1] = y0
+     for (i in 2:niter) {
+         x[i] = rgamma(1,5,2+y[i-1])
+         y[i] = rexp(1,x[i])
+     }
+     return(cbind(x,y))
+ }
```

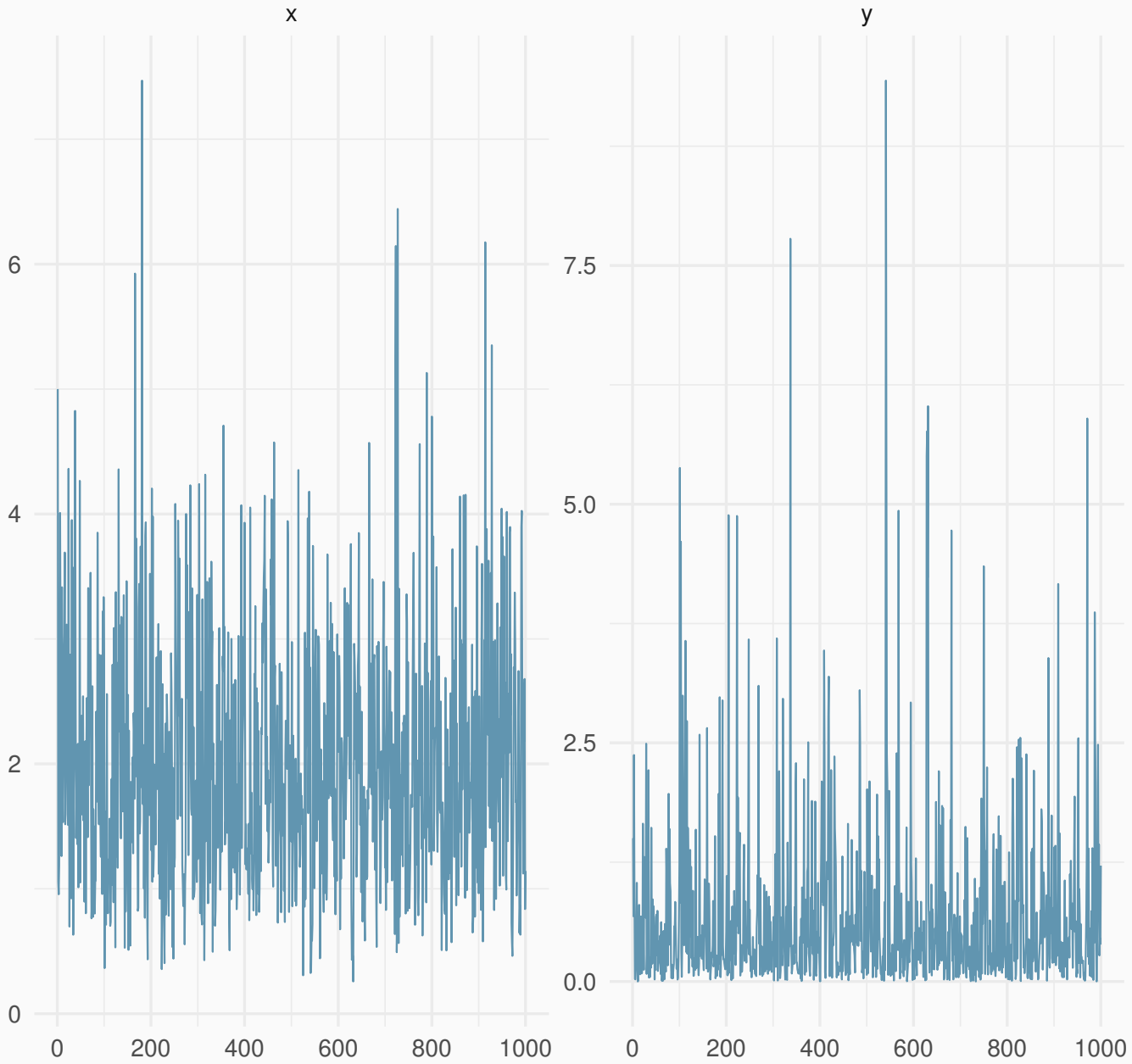
10 simulated values with starting point $x = 5$ and $y = 1.5$.

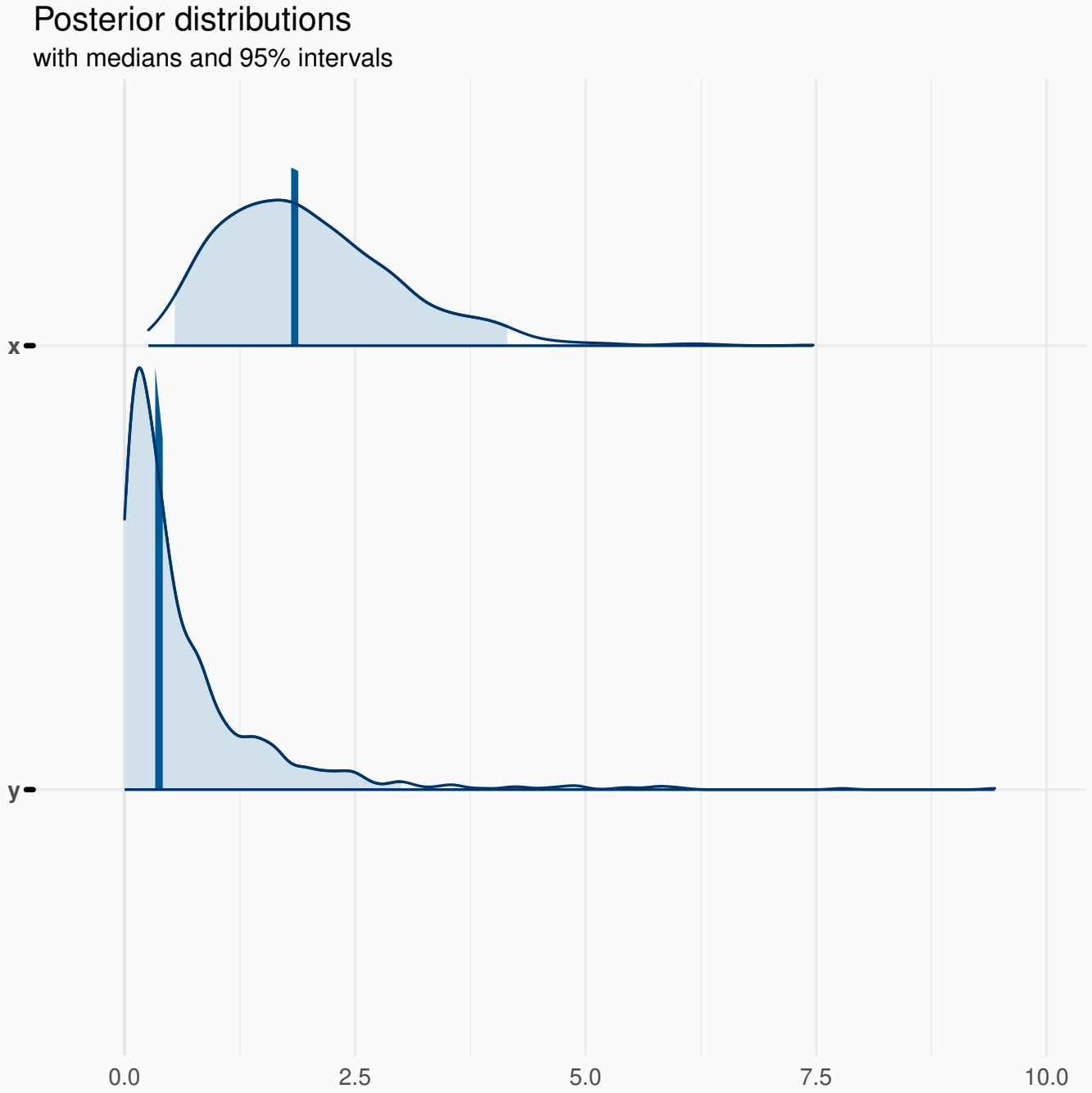


1000 simulated values with starting point $x = 5$ and $y = 1.5$.



Trace plots of MCMC draws




Posterior distributions
with medians and 95% intervals
x
y
0.0
2.5
5.0
7.5
10.0

**Example.** Let the random vector $(X, Y)$ with bivariate normal distribution,

$$(X, Y) \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$$

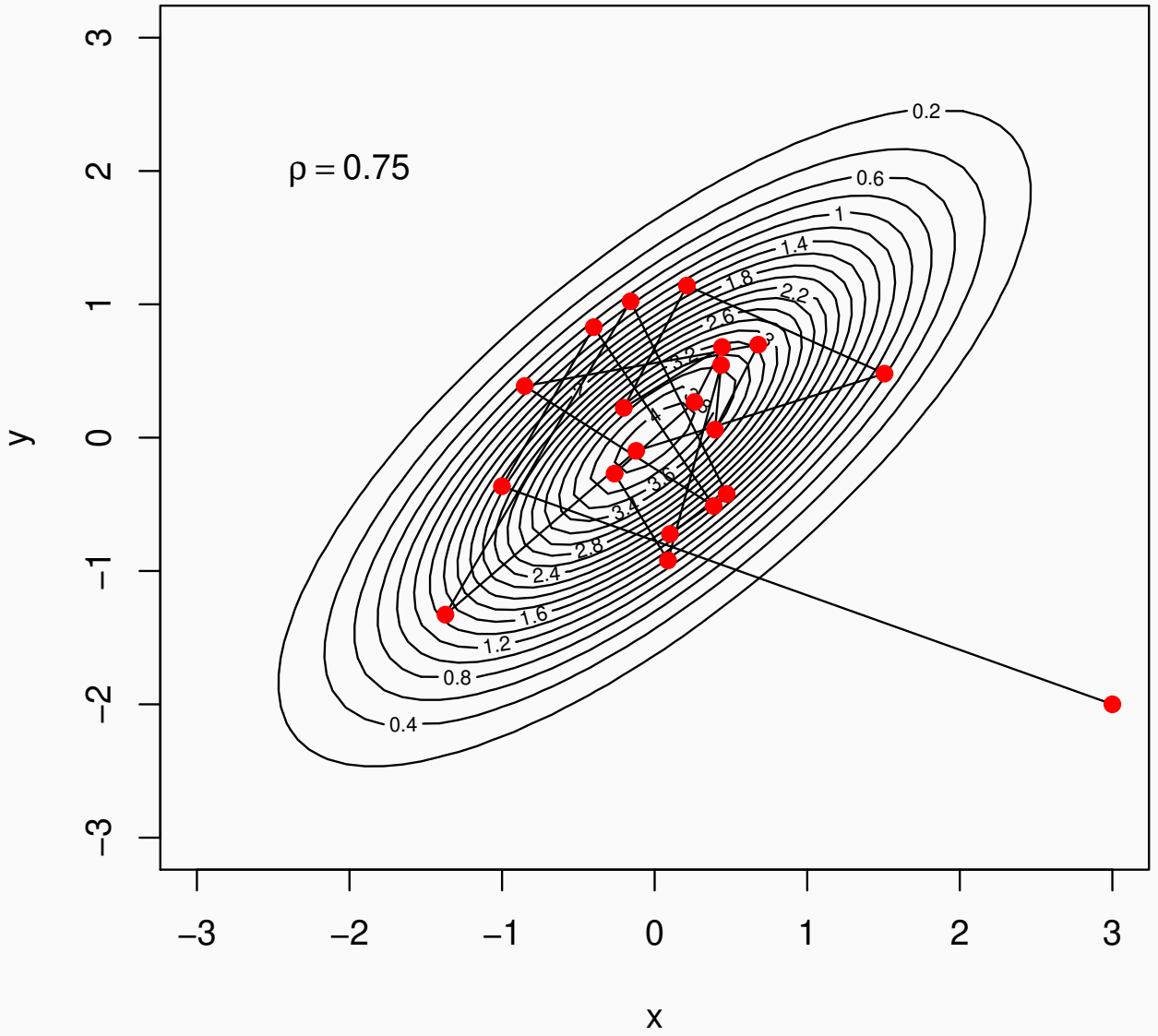
We can generate values of $(X, Y)$ using the complete conditional distributions, i.e. $X|Y = y$ and $Y|X = x$.

Given a value $y_t$ at iteration $t$,

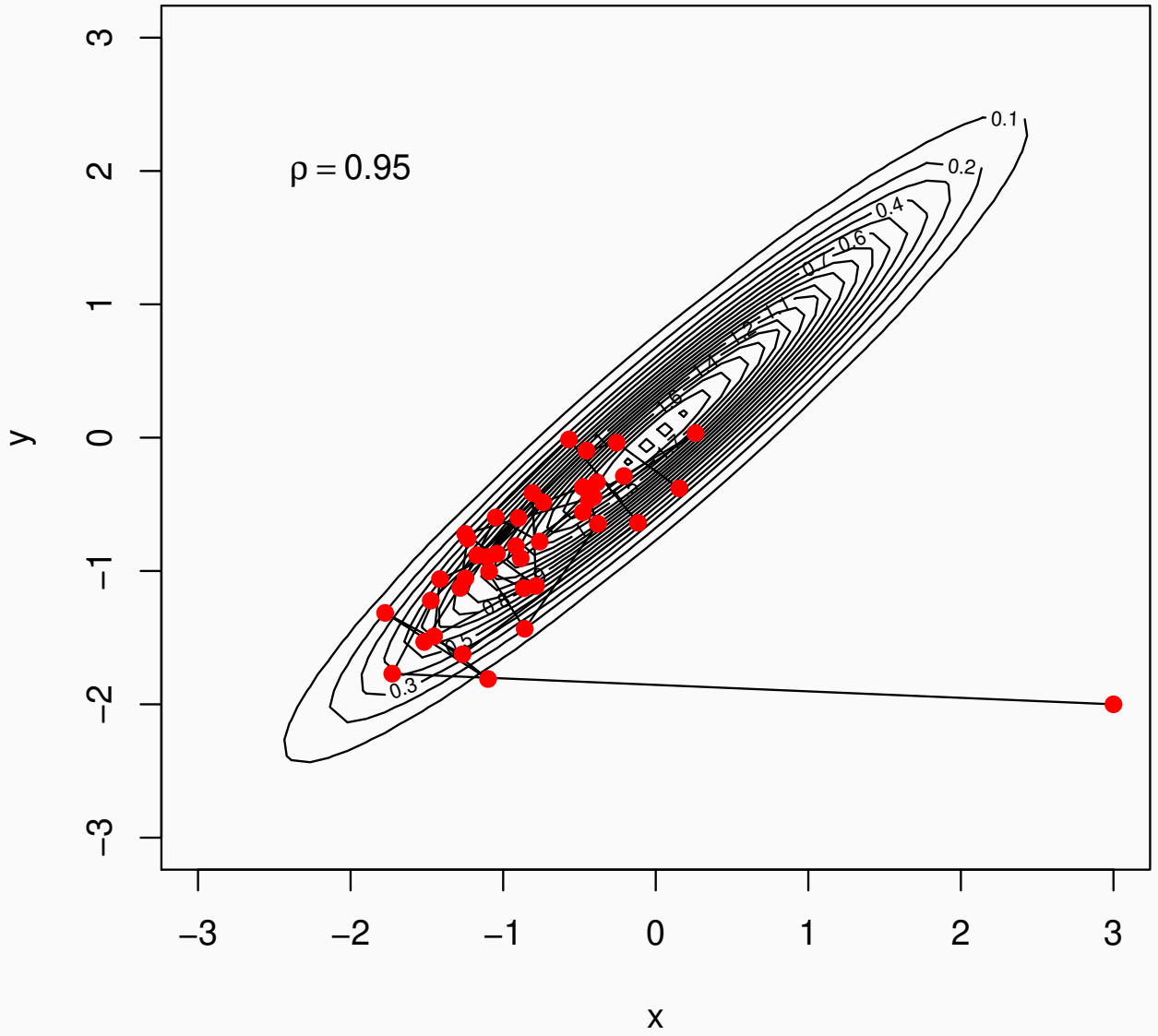
1. generate $x_{t+1}|y_t \sim N(\rho y_t, 1 - \rho^2)$,
2. generate $y_{t+1}|x_{t+1} \sim N(\rho x_{t+1}, 1 - \rho^2)$.

```
> gibbs <- function(x0,y0,niter=1000) {
+     x = y = rep(0, niter)
+     x[1] = x0
+     y[1] = y0
+     for (i in 2:niter) {
+         x[i] = rnorm(1,mean=rho*y[i-1],sd=sqrt(1-rho^2))
+         y[i] = rnorm(1,mean=rho*x[i],sd=sqrt(1-rho^2))
+     }
+     return(cbind(x,y))
+ }
```

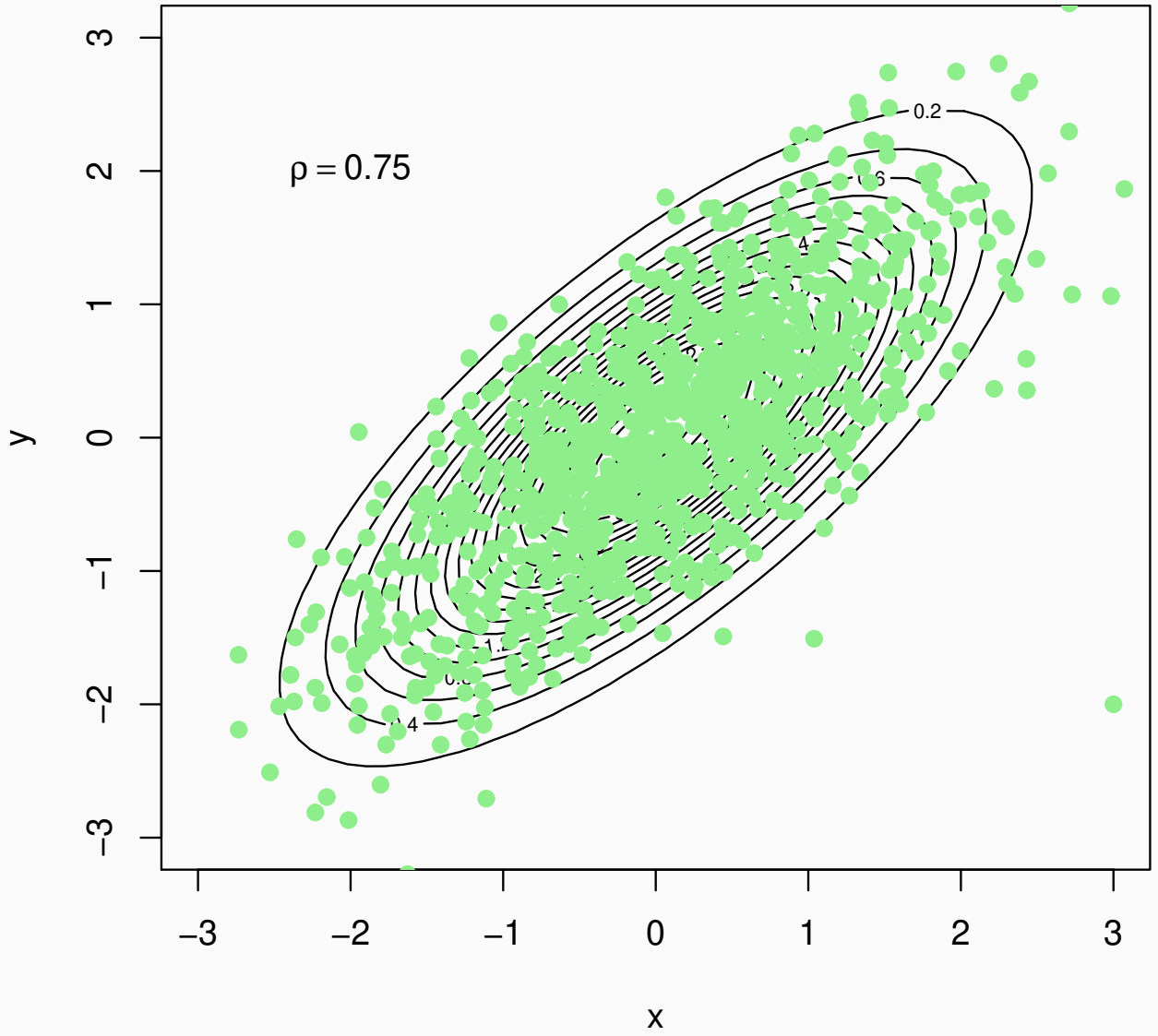
20 simulated values of $(X, Y)$.



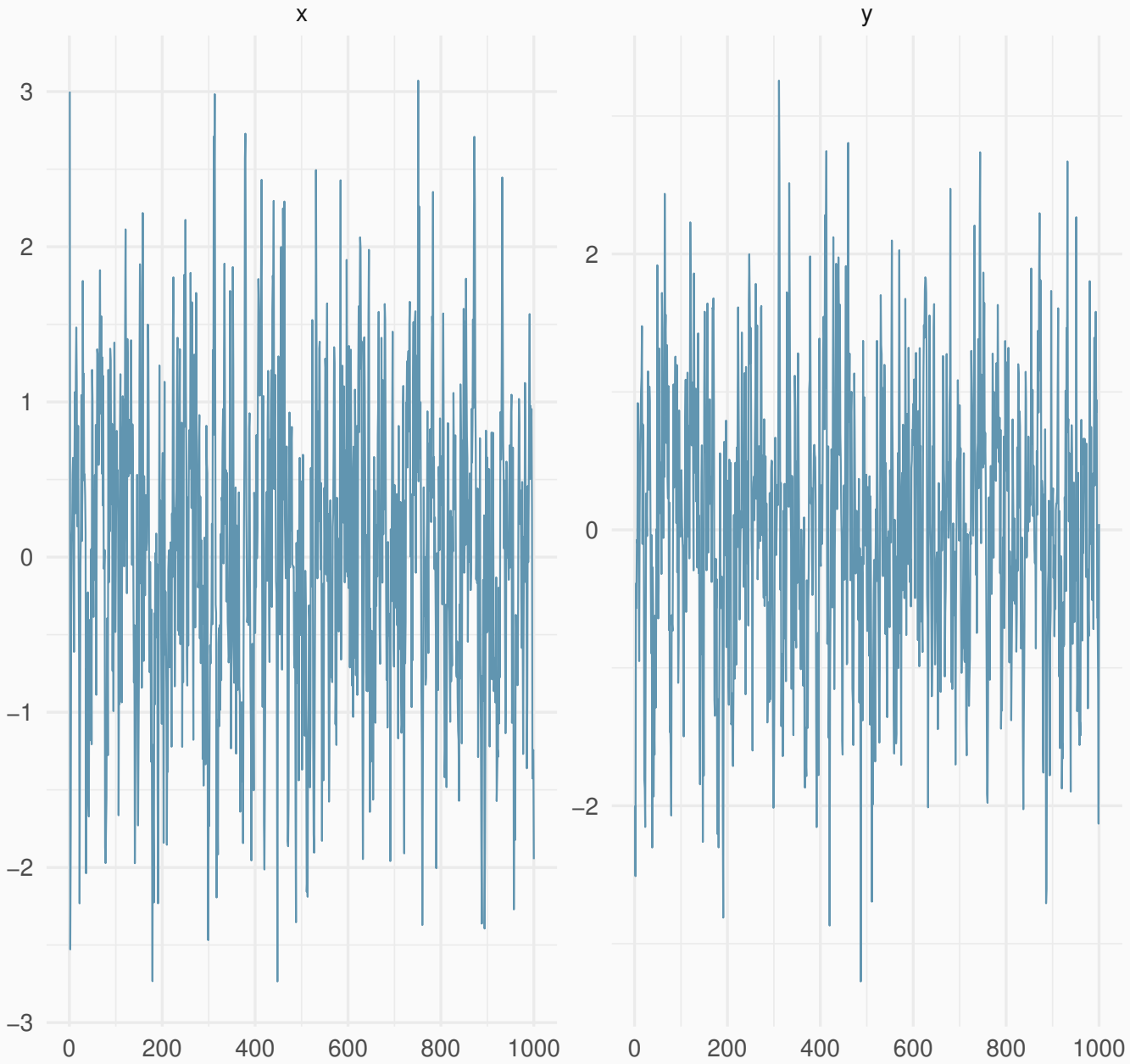
40 simulated values of $(X, Y)$.

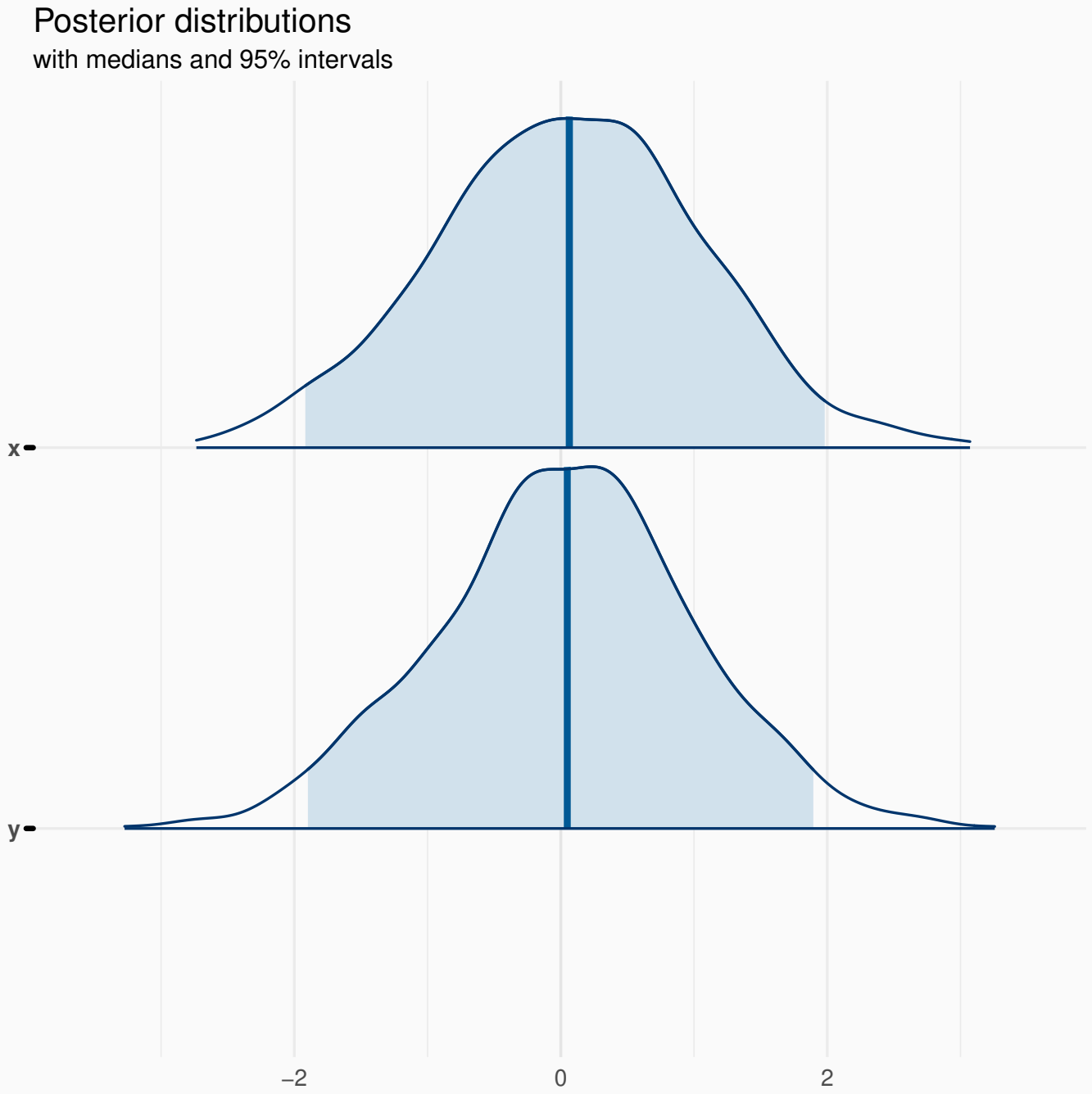


1000 simulated values of $(X, Y)$.



Trace plots of MCMC draws




Posterior distributions
with medians and 95% intervals
x
y
−2
0
2

**Example.** Autoexponential model (Besag, 1974). Let $\mathbf{y} = (y_1, y_2, y_3)$ be a random vector which density function is,

$$f(\mathbf{y}) \propto \exp[-(y_1 + y_2 + y_3 + \theta_{12} y_1 y_2 + \theta_{23} y_2 y_3 + \theta_{31} y_3 y_1)],$$

with $y_1, y_2, y_3 > 0$ and $\theta_{ij}$ known.

Verify that,

$$\begin{aligned} f(y_2|y_1) &\propto \frac{\exp[-(y_1 + y_2 + \theta_{12} y_1 y_2)]}{1 + \theta_{23} y_2 + \theta_{31} y_1}, \\ f(y_1) &\propto \exp(-y_1) \int_0^\infty \frac{\exp[-(y_2 + \theta_{12} y_1 y_2)]}{1 + \theta_{23} y_2 + \theta_{31} y_1} dy_2 \end{aligned}$$

which are dificult to simulate from.

However, the complete conditional distributions are all exponential, for example,

$$Y_3|y_1, y_2 \propto \text{Exp}(1 + \theta_{23}y_2 + \theta_{31}y_1).$$

Check the others!

**Example.** Consider a Bayesian model for the data $\mathbf{y} = (y_1, \ldots, y_n)$ which depends on parameters $\theta$, $\lambda$ and $\delta$. Suppose the joint distribution is given by,

$$p(\mathbf{y}, \theta, \lambda, \delta) \propto p(\mathbf{y}|\theta, \delta)\ p(\theta|\lambda)\ p(\lambda)\ p(\delta).$$

After observing $\mathbf{y}$ the complete conditional distributions are,

$$\begin{aligned}
\pi(\theta|\mathbf{y}, \lambda, \delta) &\propto p(\mathbf{y}|\theta, \delta)\ p(\theta|\lambda) \\
\pi(\lambda|\mathbf{y}, \theta, \delta) &\propto p(\theta|\lambda)\ p(\lambda) \\
\pi(\delta|\mathbf{y}, \theta, \lambda) &\propto p(\mathbf{y}|\theta, \delta)\ p(\delta).
\end{aligned}$$

**Example.** Suppose that $Y_1, \ldots, Y_n$ are independent and such that,

$$\begin{aligned} Y_i &\sim N(\mu, \sigma^2), i = 1, \ldots, n \\ \mu &\sim N(0, s^2) \\ \tau &\sim Gamma(a, b) \end{aligned}$$

where $\tau = \sigma^{-2}$ and $s^2$, $a$ and $b$ are known.

The likelihood function is given by,

$$p(\mathbf{y}|\mu, \tau) \propto \tau^{n/2} \exp\left[-\frac{\tau}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right].$$

So, the joint density of $(\mu, \tau)$ after observing $\mathbf{y}$ is,

$$
\begin{aligned}
p(\mu, \tau|\mathbf{y}) &\propto p(\mathbf{y}|\mu, \tau)\ p(\mu)\ p(\tau) \\
&\propto \tau^{n/2} \exp\left[-\frac{\tau}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right]\ \exp\left[-\frac{\mu^2}{2s^2}\right]\ \tau^{a-1}e^{-b\tau}.
\end{aligned}
$$

This joint density has no standard form but the complete conditional densities are easy to obtain.

$$
\begin{aligned}
p(\mu|\mathbf{y},\tau) &\propto \exp\left[-\frac{\tau}{2}\sum_{i=1}^{n}(y_i-\mu)^2\right]\exp\left[-\frac{\mu^2}{2s^2}\right] \\
&\propto \exp\left[-\frac{1}{2}(n\tau+s^{-2})\mu^2-2\mu\tau\bar{y})\right] \\
&\propto \exp\left[-\frac{1}{2C}(\mu-m)^2\right]
\end{aligned}
$$

where $C^{-1}=n\tau+s^{-2}$ and $m=Cn\tau\bar{y}$.

$$
p(\tau|\mathbf{y},\mu)\propto\tau^{a+n/2-1}\exp\left[-\tau\left(b+\frac{1}{2}\sum_{i=1}^{n}(y_i-\mu)^2\right)\right].
$$

Then, the complete conditional distributions are,

$$\begin{aligned}
\mu|\mathbf{y}, \tau &\sim N(m, C) \\
\tau|\mathbf{y}, \mu &\sim Gamma\left(a + \frac{n}{2},\ b + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right)
\end{aligned}$$

1. initialize the iterations counter $t = 0$;
2. specify initial values $(\mu^{(0)}, \tau^{(0)})$;
3. obtain $(\mu^{(t)}, \tau^{(t)})$ from $(\mu^{(t-1)}, \tau^{(t-1)})$ generating values,

$$\begin{aligned} \mu^{(t)}|\tau^{(t-1)}, \mathbf{y} &\sim N(m^{(t-1)}, C^{(t-1)}), \\ C^{(t-1)} &= (n\tau^{(t-1)} + s^{-2})^{-1}, \\ m^{(t-1)} &= C^{(t-1)} n\tau^{(t-1)} \bar{y} \end{aligned}$$

$$\tau^{(t)}|\mu^{(t)}, \mathbf{y} \sim Gamma\left(a + \frac{n}{2},\ b + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu^{(t)})^2\right)$$

4. Change the counter $t$ to $t + 1$ and return to step 3 until convergence.

```
> gibbs1 <- function(y,s2,a,b,mu0,tau0,niter) {
+   n= length(y)
+   tau= rep(0,niter)
+   mu = rep(0,niter)
+   tau[1]= tau0
+   mu [1]= mu0
+   ybar= mean(y)
+   for (i in 2:niter) {
+     C= 1/(n*tau[i-1] + 1/s2)
+     m= C * ybar
+     mu[i] = rnorm (1, mean=m, sd= sqrt(C) )
+     tau[i]= rgamma(1,a+0.5*n,b+0.5*sum((y-mu[i])^2))
+   }
+   theta = cbind(mu,1/tau)
+   colnames(theta) = c("mu","sigma2")
+   return(theta)
+ }
```

Appying to simulated data $Y_1, \ldots, Y_n \sim N(2, 4)$, $n = 50$ and 1000 iterations.

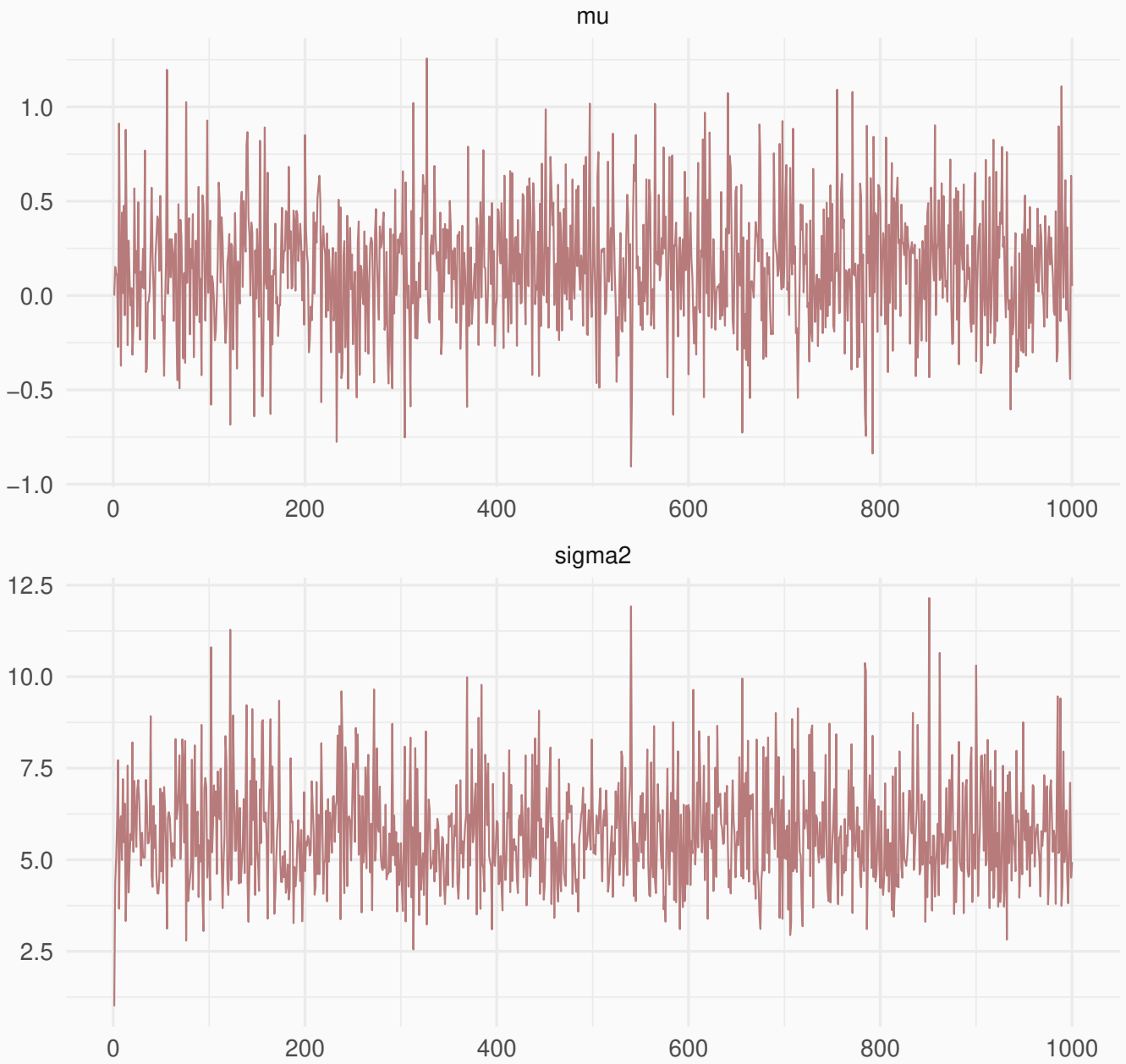
```
> y= rnorm(50,mean=2, sd=2)
```

Hyperparameters, initial values and number of iterations of the Gibbs sampler,

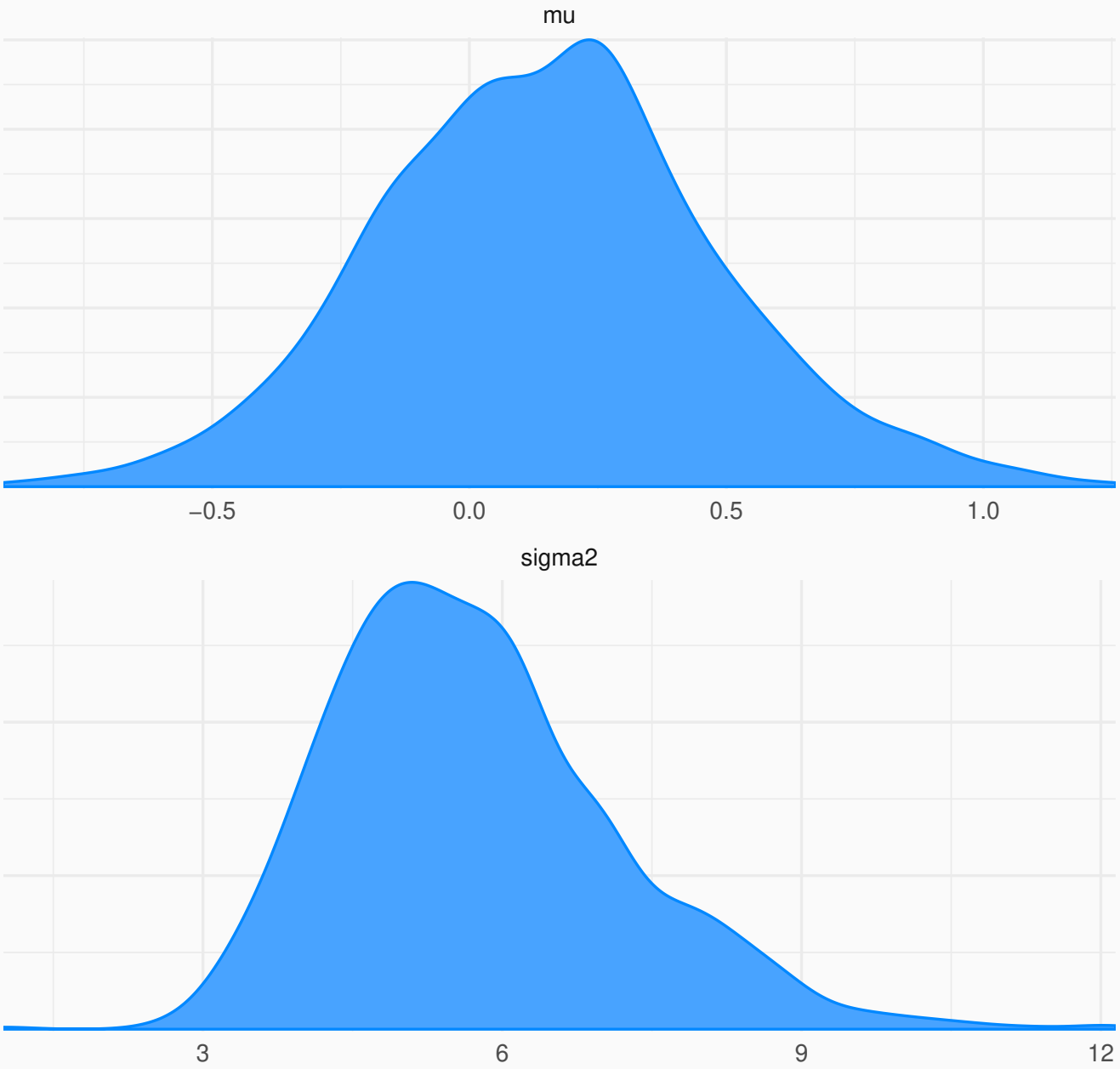
```
> s2 = 4
> a = 0.1
> b = 0.1
> mu0  = 0
> tau0 = 1
> niter= 1000

> m = gibbs1(y,s2,a,b, mu0,tau0,niter)
```

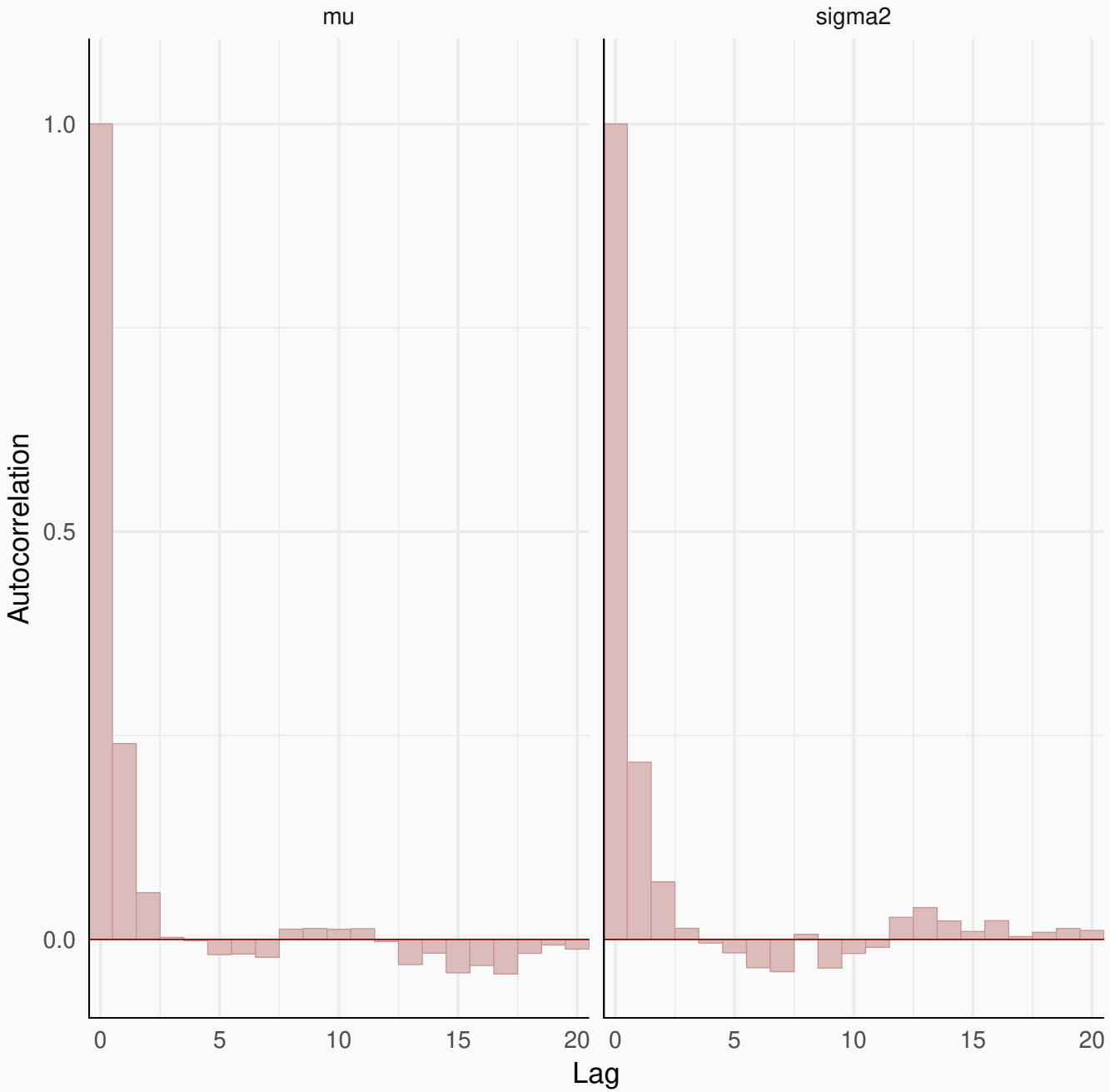
```
Iterations = 1:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean     SD Naive SE Time-series SE
mu      0.1504 0.3318  0.01049        0.01014
sigma2 5.7227 1.4457  0.04572        0.04244

2. Quantiles for each variable:

           2.5%      25%    50%    75%  97.5%
mu      -0.4915 -0.07201 0.1518 0.3544 0.8514
sigma2   3.4095  4.69353 5.5404 6.5064 8.8769
```

# Trace plots of MCMC draws

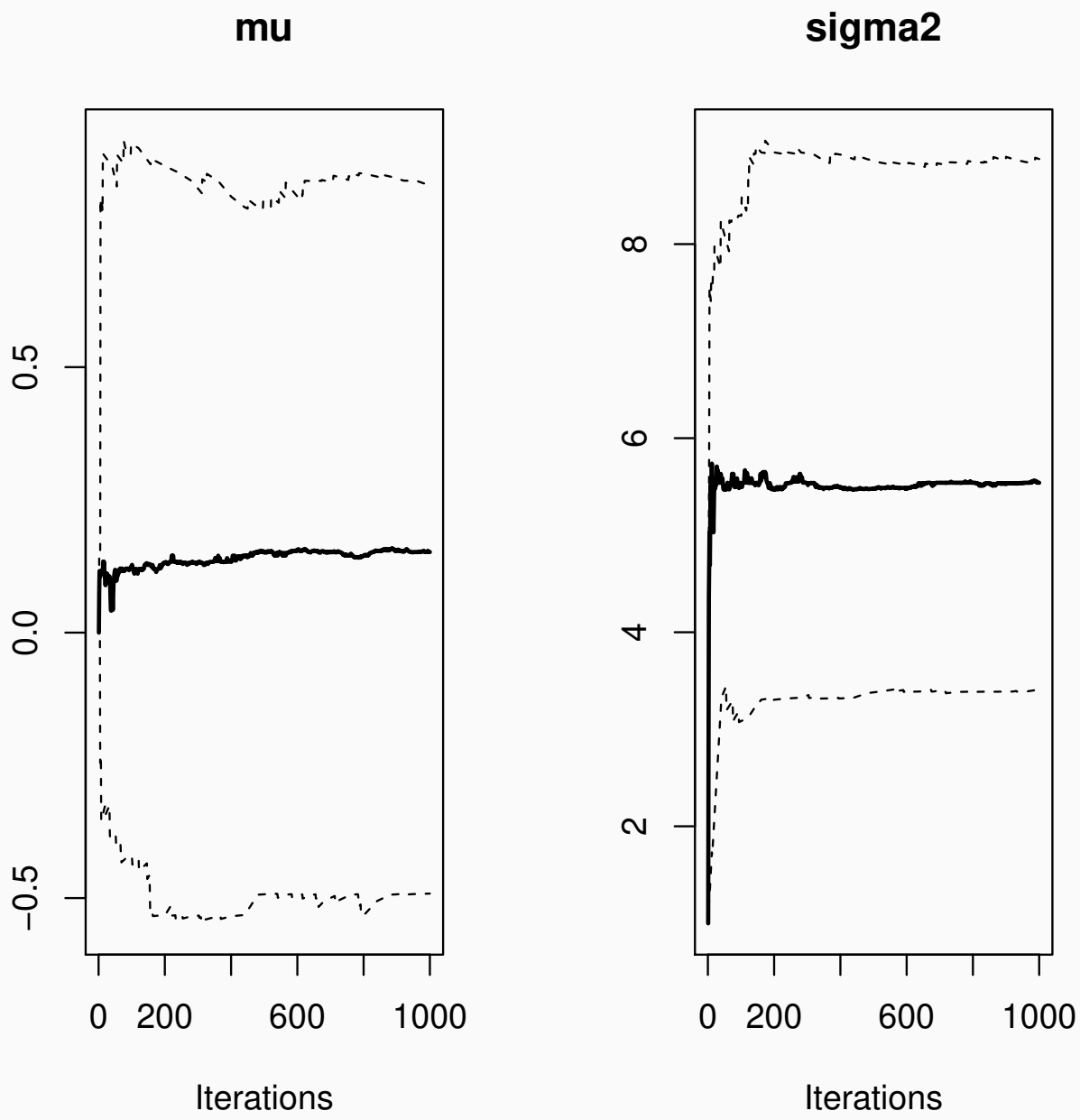


# Density Plots




mu
sigma2
1.0
0.5
0.0
Autocorrelation
0
5
10
15
20
Lag

Evolution of sample quantiles (2.5%, 50%, 97.5%).



**Example.** Let $Y_1, \ldots, Y_n \sim Poisson(\lambda)$ with $\lambda \sim Exp(\beta)$. Instead of specifying a value for $\beta$ this is also assigned a distribution, $\beta \sim Gamma(c, d)$.

Then we have that,

$$\begin{aligned} p(\mathbf{y}|\lambda) &= \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \\ p(\lambda|\beta) &= \beta e^{-\beta\lambda} \\ p(\beta) &= \frac{d^c}{\Gamma(c)} \beta^{c-1} e^{-d\beta} \end{aligned}$$

$$
\begin{aligned}
p(\lambda, \beta|\mathbf{y}) &\propto \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}\ p(\lambda|\beta)\ p(\beta) \\
&\propto \lambda^{t} e^{-n\lambda}\ \beta\ e^{-\beta\lambda}\ \beta^{c-1} e^{-d\beta}, \quad t = \sum_{i=1}^{n} y_i.
\end{aligned}
$$

$$
\begin{aligned}
p(\lambda|\beta, \mathbf{y}) &\propto \lambda^{t} e^{-(\beta+n)\lambda} \\
p(\beta|\lambda, \mathbf{y}) &\propto \beta^{c} e^{-(\lambda+d)\beta}
\end{aligned}
$$

The complete conditional distributions are,

$$
\lambda|\beta, \mathbf{y} \sim \mathit{Gamma}(t+1, \beta+n)
$$

$$
\beta|\lambda, \mathbf{y}) \sim \mathit{Gamma}(c+1, \lambda+d)
$$

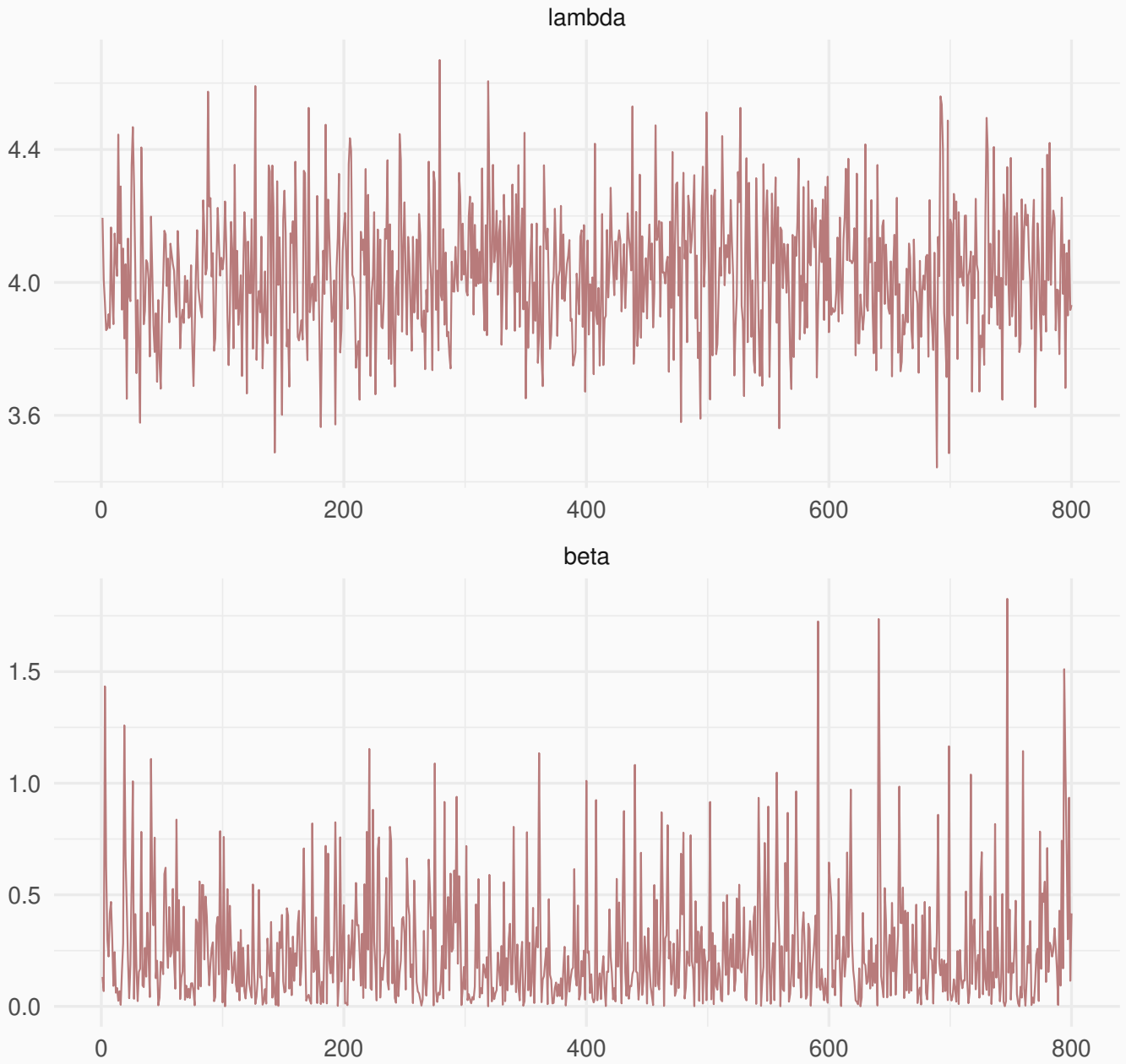
```
> Gibbs2 <- function(c,d,y,niter){
+  N = length(y)
+  lambda = matrix(0, nrow=niter)
+  beta   = matrix(0, nrow=niter)
+  lambda[1]= 1
+  beta  [1]= 1
+  t1 = sum(y)
+  for (i in 2:niter) {
+    lambda[i]= rgamma(1, 1+t1, beta[i-1]+N)
+    beta  [i]= rgamma(1, 1+c, lambda[i]+d)
+  }
+  return(theta = list(lambda=lambda,beta=beta))
+ }
```

**Example.** Applying the previous algorithm with 100 simulated data $Y_1, \ldots, Y_n \sim Poisson(4)$, $n = 100$, $c = d = 0.01$ and 1000 iterations.

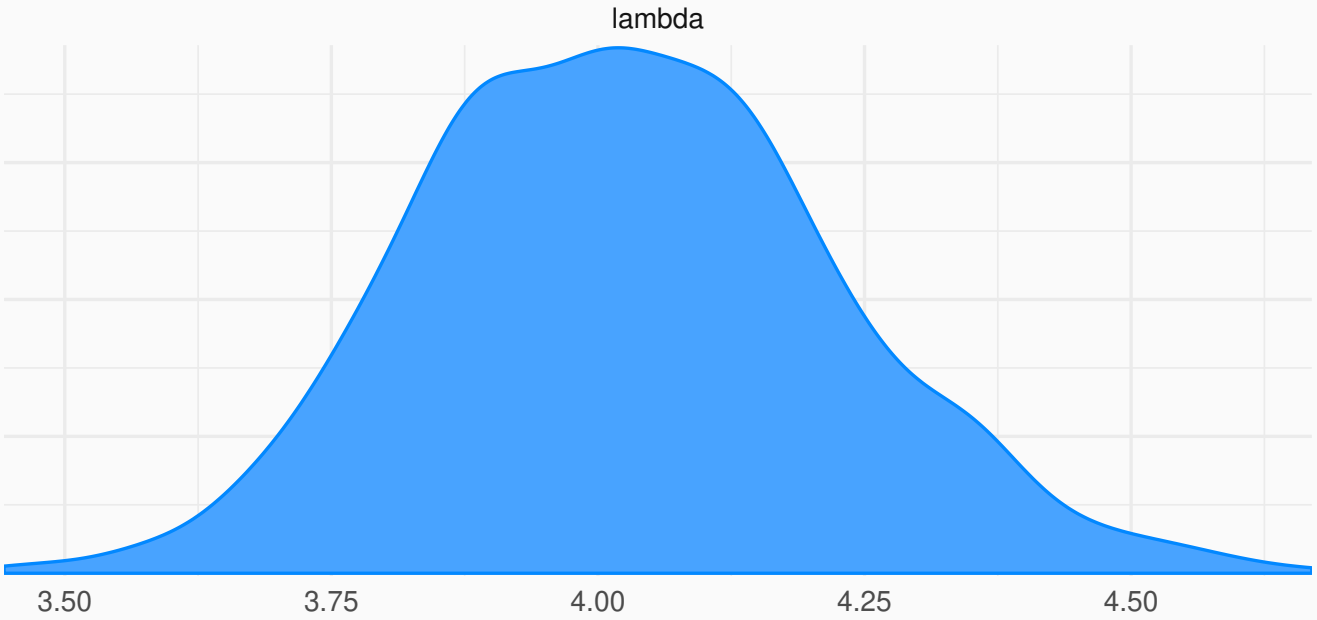
```
> y= rpois(100,lambda=4)
> niter = 1000
> q = Gibbs2(c = 0.01, d = 0.01, y, niter= niter)
> q1 = mcmc(cbind(q$lambda, q$beta))
> q2= window(q1, start=201)
> varnames(q2)=c("lambda","beta")


> y

  [1]  3  5  3  6  3  3  4  4  3  4  4  3  6  2  2  2  2  1  6
 [26]  4  2  6  1  4  4 13  5  3  3  3  2  5  5  3  3  4  4  4
 [51]  3  4  3  8 11  3  3  7  6  3  6  9  2  6  4  2  3  6  3
 [76]  4  1  2  3  3  2  5  3  1  3  4  4  3  3  7  6  2  3  3
```

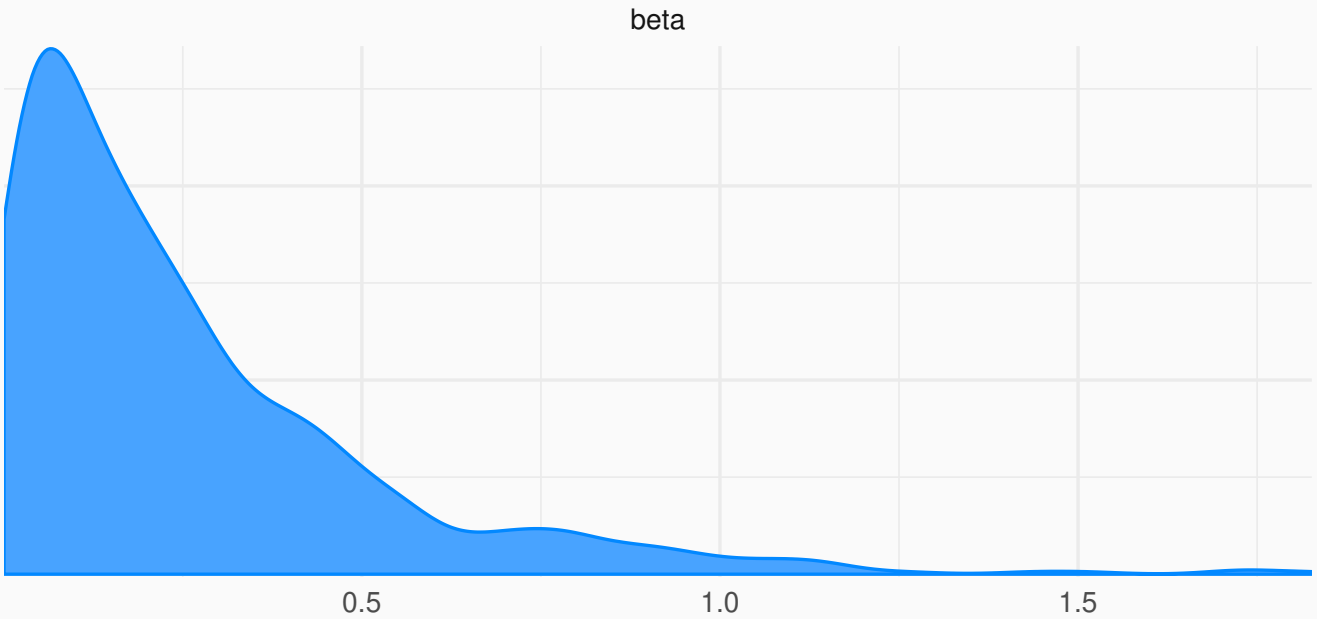
```
Iterations = 201:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 800

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

         Mean     SD Naive SE Time-series SE
lambda 4.0268 0.1948 0.006889       0.006889
beta   0.2541 0.2640 0.009334       0.009334

2. Quantiles for each variable:

           2.5%     25%    50%    75%  97.5%
lambda 3.671315 3.89019 4.0192 4.1509 4.4203
beta   0.004875 0.06878 0.1703 0.3531 0.9634
```

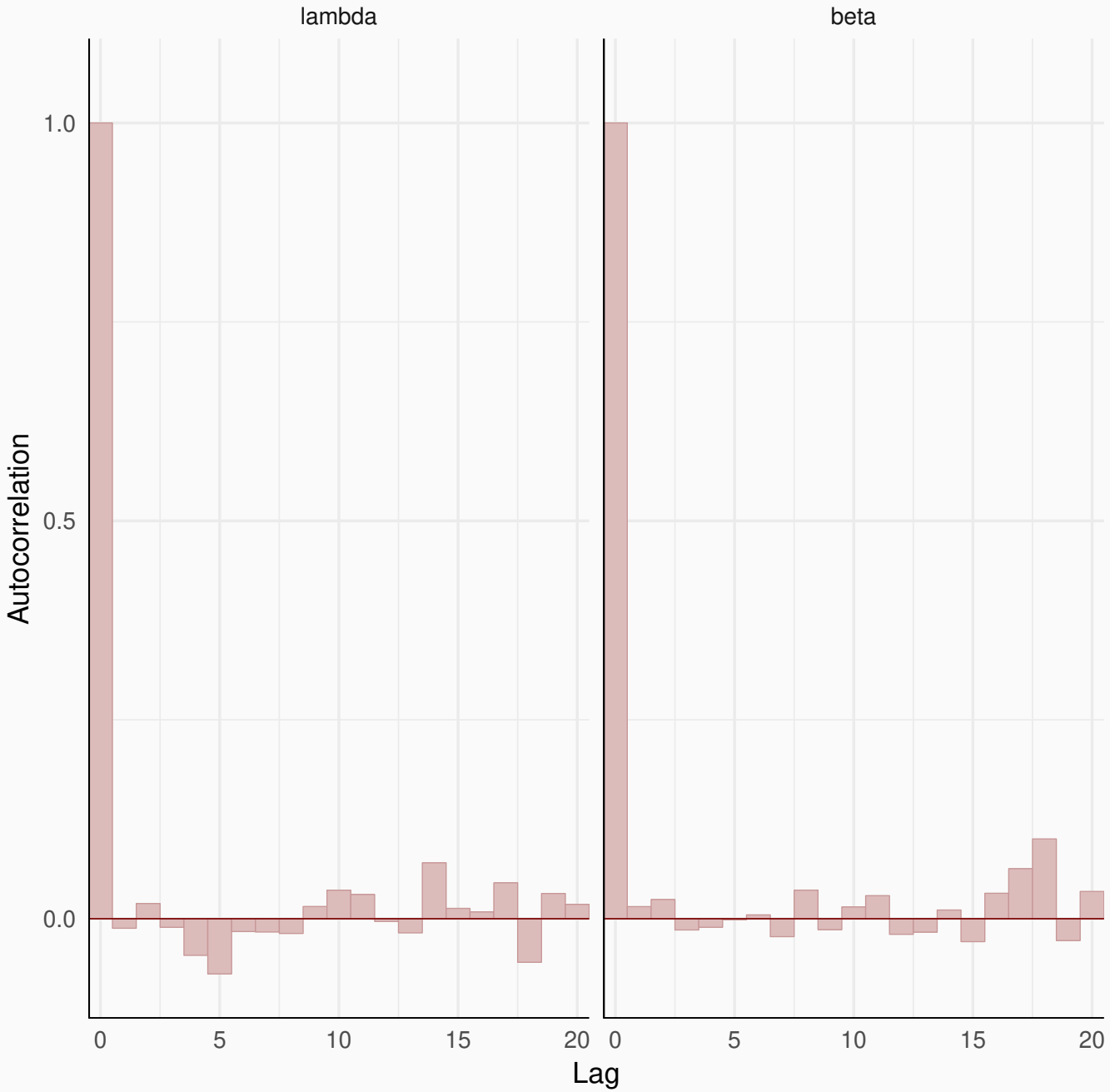
# Trace plots of MCMC draws



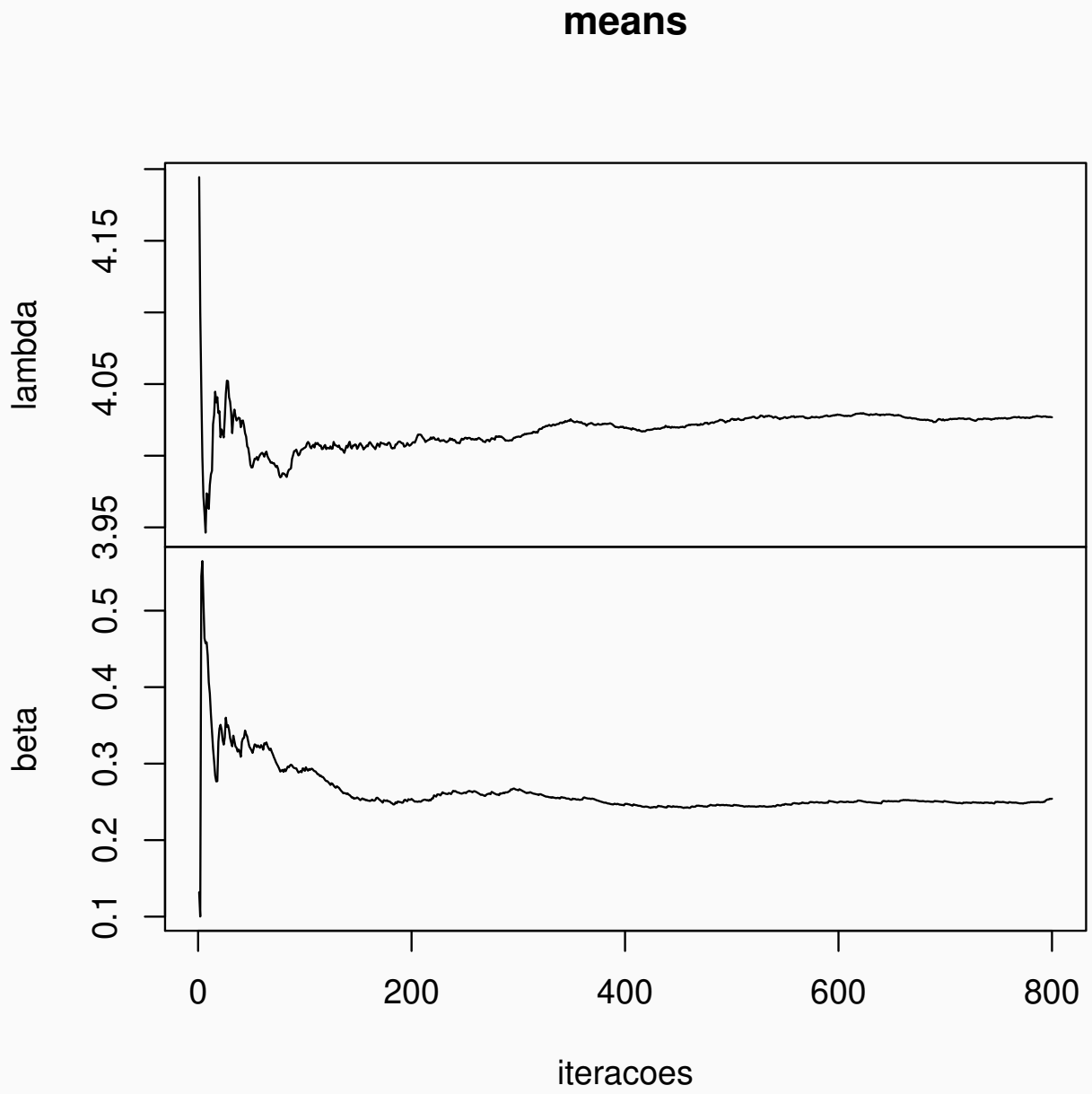
# Density Plots



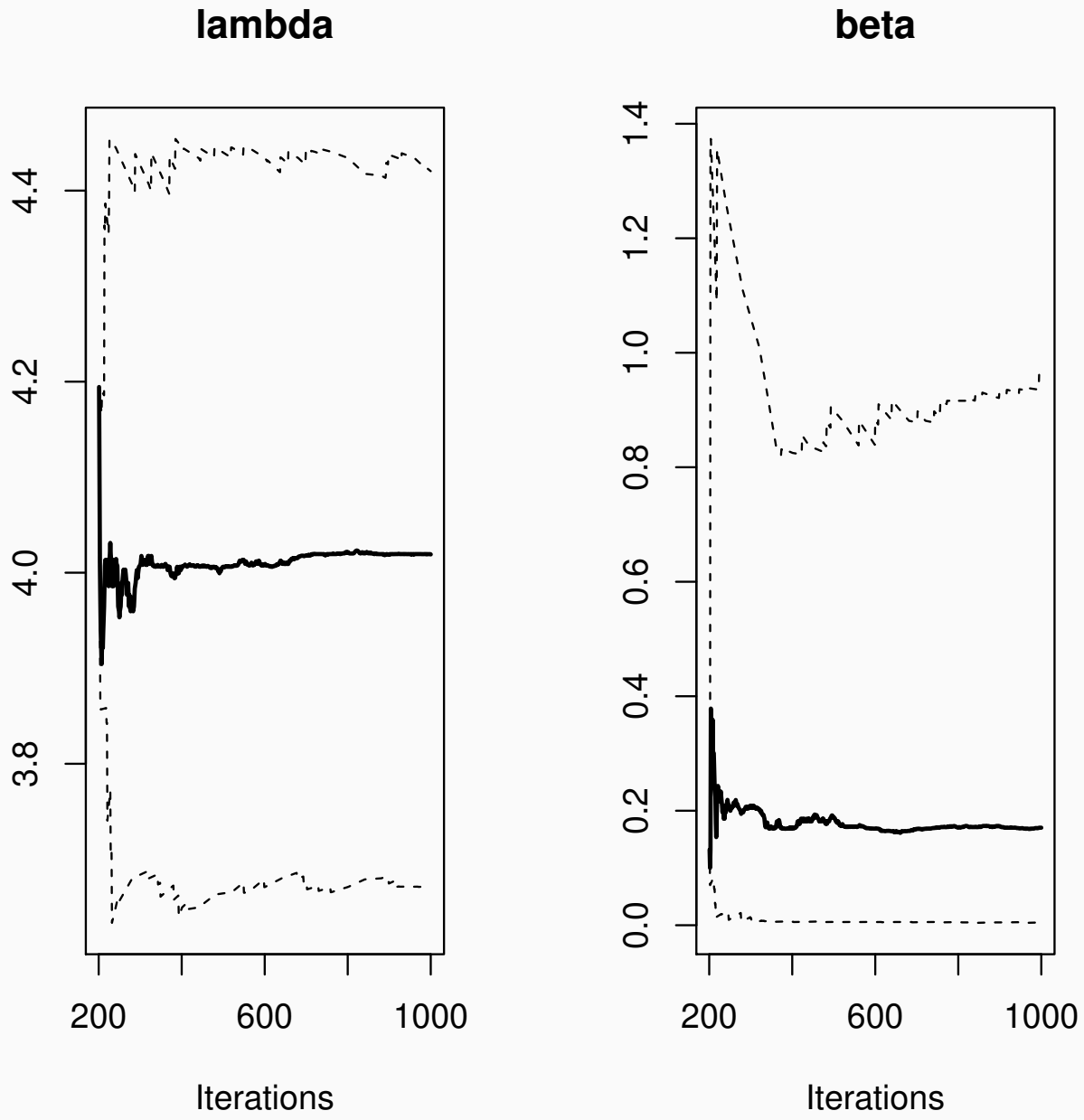




lambda
beta
1.0
0.5
0.0
Autocorrelation
0
5
10
15
20
Lag

Ergodic means along iterations.



Evolution of sample quantiles (2.5%, 50%, 97.5%).



**Example.** The data below are yearly counts on the number of marriages per 1000 people in Italy between 1936 and 1951.

```
> y = c(7,8,9,7,7,6,6,5,5,7,9,10,8,8,8,7)
```

The mean counts should not be constant along the years because of WWII (1939-1945)

Lets assume the following model,

$$\begin{aligned} Y_i|\lambda_i &\sim Poisson(\lambda_i) \\ \lambda_i &\sim Exp(\beta), i = 1, \ldots, n \\ \beta &\sim Gamma(c, d) \end{aligned}$$

Then,

$$
\begin{aligned}
p(\beta, \lambda_1, \ldots, \lambda_n \mid \mathbf{y}) &\propto \prod_{i=1}^{n} \lambda_i^{y_i} e^{-\lambda_i} \prod_{i=1}^{n} \beta e^{-\beta\lambda_i} \; \beta^{c-1} e^{\beta d} \\
p(\beta|\mathbf{y}, \lambda_1, \ldots, \lambda_n) &\propto \beta^{c+n-1} e^{-\beta(d+\sum_{i=1}^{n} \lambda_i)} \\
p(\lambda_i|\beta, \mathbf{y}, \lambda_{-i}) &\propto \lambda_i^{y_i} e^{-\lambda_i(\beta+1)}
\end{aligned}
$$

and the complete conditional distributions are,

$$
\begin{aligned}
\beta|\mathbf{y}, \lambda_1, \ldots, \lambda_n &\sim Gamma(c+n, d+\sum_{i=1}^{n} \lambda_i) \\
\lambda_i|\beta, \mathbf{y}, \lambda_{-i} &\sim Gamma(y_i+1, \beta+1)
\end{aligned}
$$

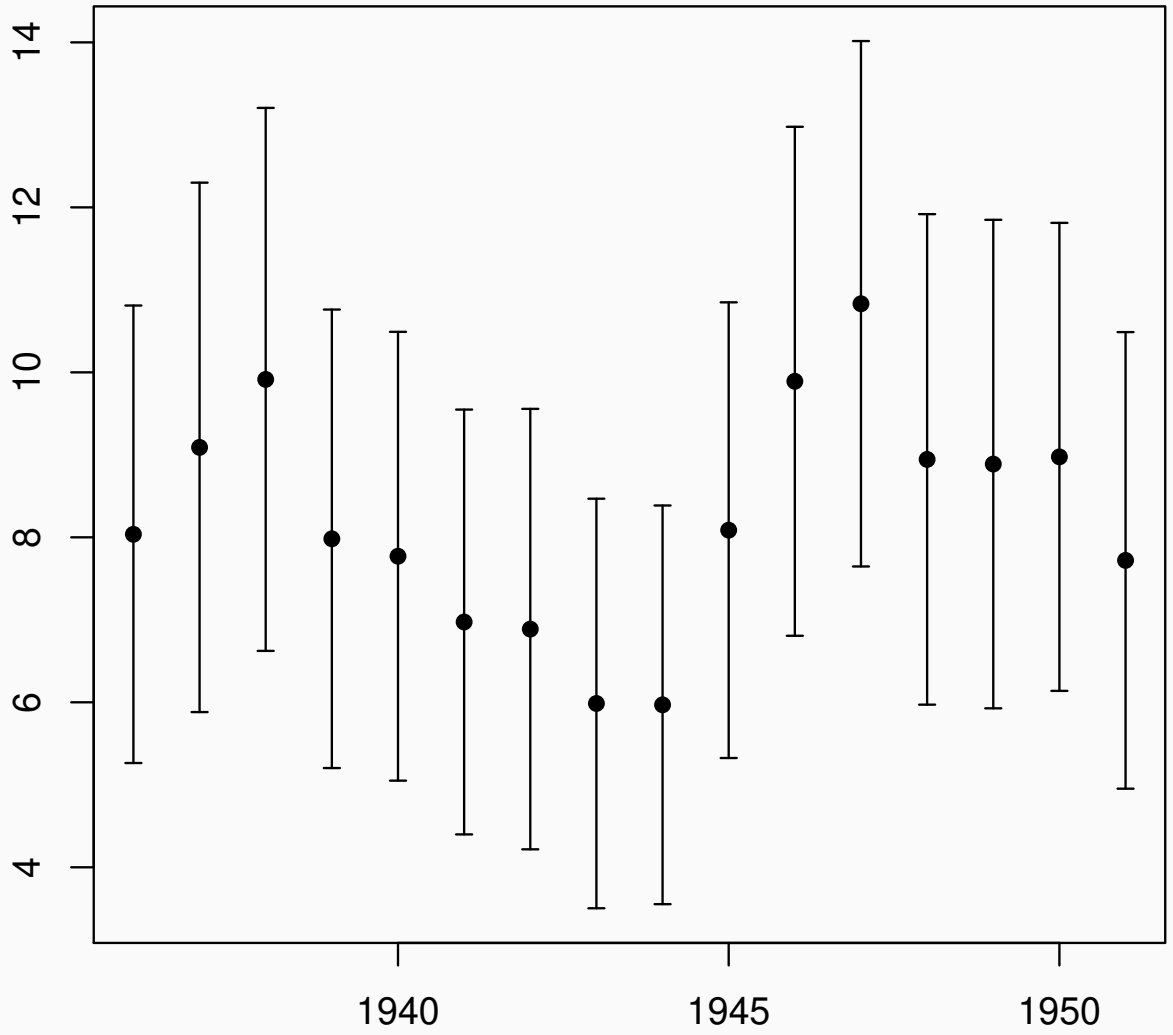
```
> Gibbs3 <- function (c, d, y, niter){
+  N = length(y)
+  lambda = matrix(0, nrow=niter, ncol=N)
+  beta   = matrix(0, nrow=niter)
+  lambda[1,] = rep(1,N)
+  beta[1] = 1
+  for (i in 2:niter){
+    for (j in 1:N) lambda[i,j]=rgamma(1,y[j]+1,beta[i-1]+1)
+    beta[i] = rgamma(1,c+1, d+sum(lambda[i,]))
+  }
+  return(list(lambda = lambda, beta = beta))
+ }
```

```
Iterations = 201:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 800

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean       SD  Naive SE Time-series SE
lambda 1   8.036964 2.772828 0.0980343      0.0980343
lambda 2   9.090367 3.208959 0.1134538      0.1134538
lambda 3   9.914465 3.291191 0.1163612      0.1163612
lambda 4   7.981923 2.778987 0.0982520      0.0982520
lambda 5   7.770618 2.720335 0.0961784      0.1034952
lambda 6   6.973593 2.574933 0.0910376      0.0910376
lambda 7   6.887447 2.669580 0.0943839      0.0943839
lambda 8   5.985916 2.482456 0.0877681      0.0877681
lambda 9   5.969422 2.416393 0.0854324      0.0854324
```

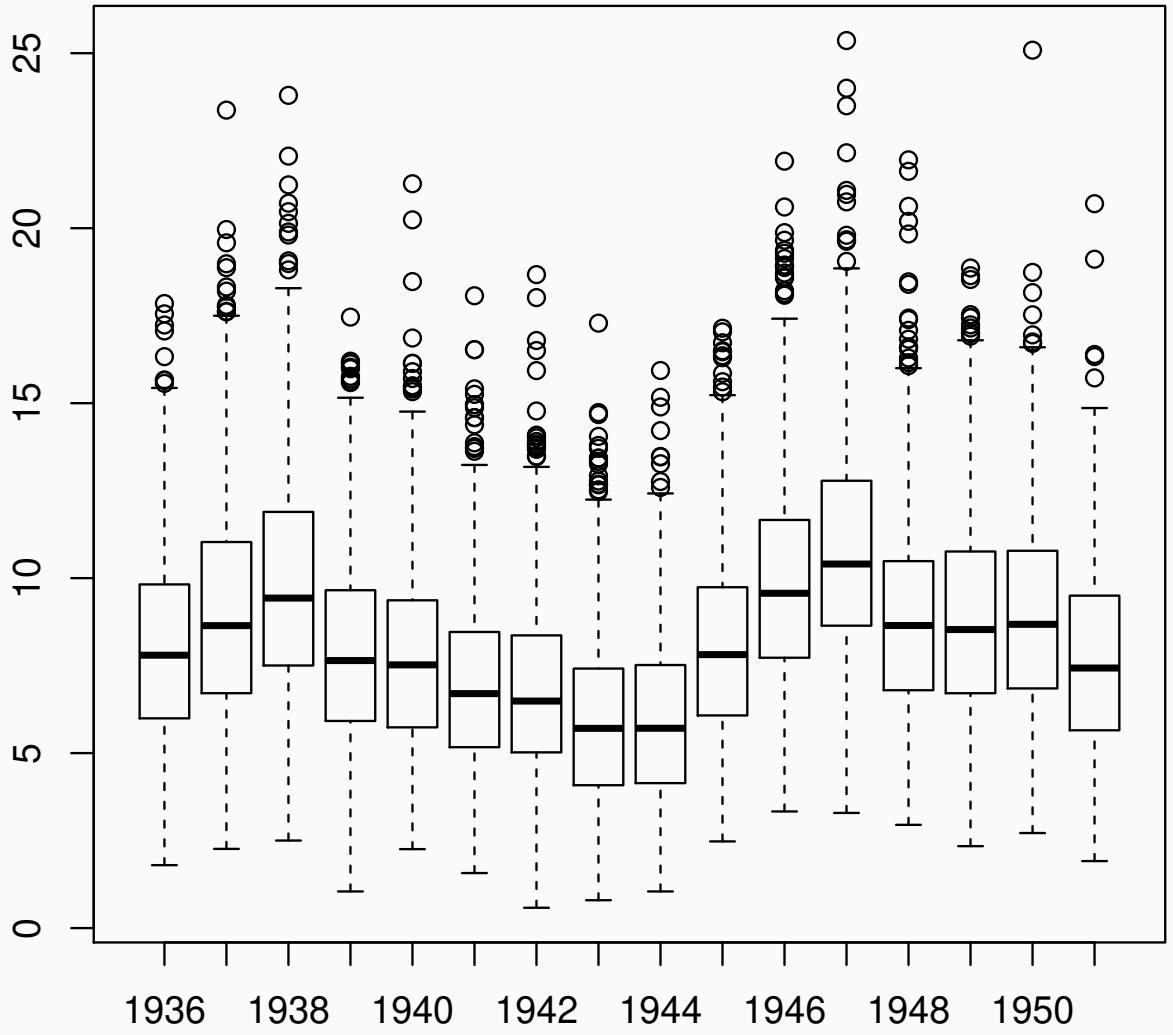
```
lambda 10  8.086695 2.761989 0.0976510      0.0976510
lambda 11  9.891530 3.085049 0.1090730      0.1090730
lambda 12 10.831315 3.184578 0.1125918      0.1063655
lambda 13  8.944409 2.973192 0.1051182      0.1051182
lambda 14  8.888019 2.961097 0.1046906      0.0976864
lambda 15  8.975193 2.836684 0.1002919      0.1002919
lambda 16  7.720314 2.767398 0.0978423      0.0978423
beta       0.008582 0.008066 0.0002852      0.0002852

2. Quantiles for each variable:

                2.5%      25%       50%      75%    97.5%
lambda 1  3.2989197 5.999970  7.798447  9.81764 14.02706
lambda 2  4.0272630 6.713192  8.642866 11.03150 16.30225
lambda 3  4.6681968 7.509754  9.430119 11.89183 17.51223
lambda 4  3.5345713 5.918542  7.645193  9.65230 14.52621
lambda 5  3.3858330 5.737206  7.521261  9.36127 13.70272
lambda 6  2.6603290 5.168934  6.700526  8.45446 12.74351
lambda 7  2.4883211 5.023559  6.487670  8.35184 12.78319
```

```
lambda 8  2.1515758 4.084871  5.707433  7.41437 12.00800
lambda 9  2.2371868 4.147076  5.710186  7.51369 11.18822
lambda 10 3.5293448 6.077594  7.814698  9.73587 14.49105
lambda 11 4.8901952 7.726047  9.567724 11.65814 17.10528
lambda 12 5.6296630 8.641317 10.403110 12.78249 17.45366
lambda 13 4.3366380 6.798819  8.646320 10.47850 15.55489
lambda 14 4.0582237 6.711474  8.530859 10.75974 15.53948
lambda 15 4.5066376 6.851088  8.680670 10.77858 15.09539
lambda 16 3.0208822 5.656204  7.430940  9.49538 13.85627
beta      0.0002779 0.002915  0.006202  0.01195  0.02975
```

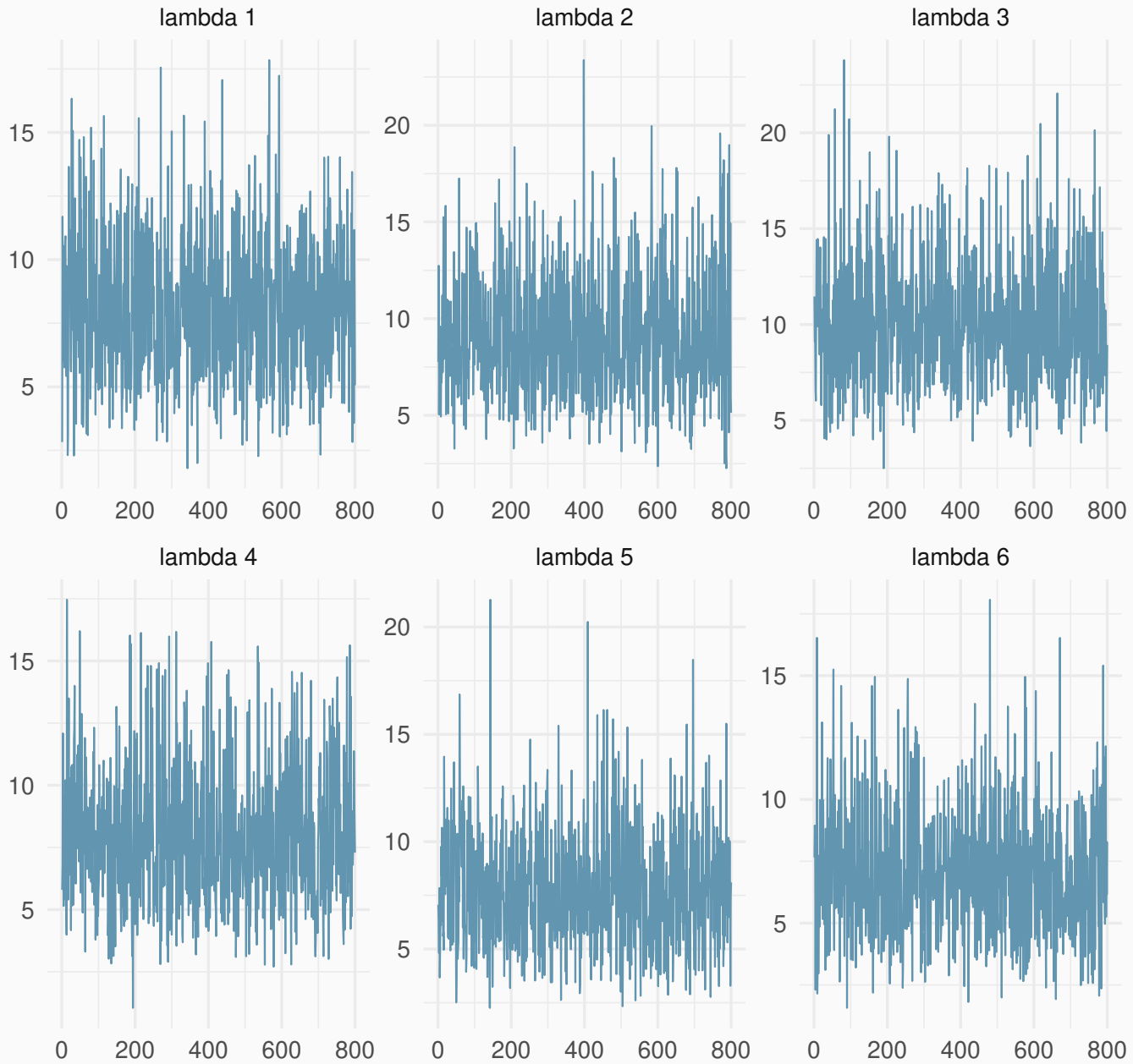
| | mean | sd |
|---|---|---|
| 1936 | 8.036964 | 2.772828 |
| 1937 | 9.090367 | 3.208959 |
| 1938 | 9.914465 | 3.291191 |
| 1939 | 7.981923 | 2.778987 |
| 1940 | 7.770618 | 2.720335 |
| 1941 | 6.973593 | 2.574933 |
| 1942 | 6.887447 | 2.669580 |
| 1943 | 5.985916 | 2.482456 |
| 1944 | 5.969422 | 2.416393 |
| 1945 | 8.086695 | 2.761989 |
| 1946 | 9.891530 | 3.085049 |
| 1947 | 10.831315 | 3.184578 |
| 1948 | 8.944409 | 2.973192 |
| 1949 | 8.888019 | 2.961097 |
| 1950 | 8.975193 | 2.836684 |
| 1951 | 7.720314 | 2.767398 |


14
12
10
8
6
4
1940
1945
1950

Boxplots of simulated values of latent variables.



# Trace plots of MCMC draws



**Example.** Let $Y_1, \ldots, Y_n$ be independent counts where we suspect that there was a change point $m$ such that,

$$\begin{aligned} Y_i|\lambda &\sim Poisson(\lambda), \quad i = 1, \ldots, m \\ Y_i|\phi &\sim Poisson(\phi), \quad i = m+1, \ldots, n. \end{aligned}$$

We want to estimate $m$, $\lambda$ and $\phi$.

Assign the following independent priors,

$$\begin{aligned} \lambda &\sim Gamma(a, b) \\ \phi &\sim Gamma(c, d) \\ p(m) &= 1/n. \end{aligned}$$

The joint posterior density is,

$$\begin{aligned} p(\lambda, \phi, m|\mathbf{y}) &\propto \prod_{i=1}^{m} e^{-\lambda}\lambda^{y_i} \prod_{i=m+1}^{n} e^{-\phi}\phi^{y_i}\ \lambda^{a-1}e^{-b\lambda}\ \phi^{c-1}e^{-d\phi}\ \ \frac{1}{n} \\ &\propto \lambda^{a+t_1-1}e^{-(b+m)\lambda}\phi^{c+t_2-1}e^{-(d+n-m)\phi}\ \ \frac{1}{n} \end{aligned}$$

where,

$$t_1 = \sum_{i=1}^{m} y_i \quad \text{and} \quad t_2 = \sum_{i=m+1}^{n} y_i.$$

.

The complete conditional densities are given by,

$$\begin{aligned}
p(\lambda|\phi, m, \mathbf{y}) &\propto \lambda^{a+t_1-1} e^{-(b+m)\lambda} \\
p(\phi|\lambda, m, \mathbf{y}) &\propto \phi^{c+t_2-1} e^{-(d+n-m)\phi} \\
p(m|\lambda, \phi, \mathbf{y}) &\propto \lambda^{t_1} e^{-m\lambda} \phi^{t_2} e^{-(n-m)\phi}, \quad m = 1, \ldots, n.
\end{aligned}$$

so that,

$$\begin{aligned}
\lambda|\phi, m, \mathbf{y} &\sim Gamma(a + t_1, b + m) \\
\phi|\lambda, m, \mathbf{y} &\sim Gamma(c + t_2, d + n - m)
\end{aligned}$$

```
> Gibbs <- function(a,b,c,d,y,niter){
+ N = length(y)
+ lambda = phi = m = matrix(0, nrow=niter)
+ lambda[1] = phi[1] = 1; m[1] = 10
+ for (i in 2:niter) {
+   t1 = sum(y[1:m[i-1]]); t2 = 0; prob = NULL
+   if (m[i-1] < N) t2 = sum(y[(m[i-1]+1):N])
+   lambda[i] = rgamma(1,(a + t1), (b +   m[i-1]))
+   phi[i]    = rgamma(1,(c + t2), (d + N-m[i-1]))
+   for (j in 1:N){
+     t1 = sum(y[1:j])
+     t2 = 0
+     if (j < N) t2 = sum(y[(j+1):N])
+     aux=(lambda[i-1]^t1)*exp(-j*lambda[i-1])*(phi[i-1]^t2)*exp
+     prob = c(prob,aux)
+     }
+   soma = sum(prob)
+   probm = prob/soma
+   m[i] = sample(x=N, size=1, prob=probm)
+   }
+   return(list(lambda=lambda, phi=phi, m=m))}
```

**Example.** Testing the function `Gibbs` with 40 simulated data from processes with means 2 and 5 with change point at 23.

```
> set.seed(124)
> y= c(rpois(n=22, lambda=2),rpois(n=18, lambda=5))
> y

 [1]  0  2  2  1  1  1  2  2  4  1  3  3  3  3  2  0  2  3  4  0
[26]  3  7  4  7  4  7  4  4  7  2 10  3  9  3  4

> x = Gibbs(a=0.1, b=0.1, c=0.1, d=0.1, y=y, niter=5000)
```

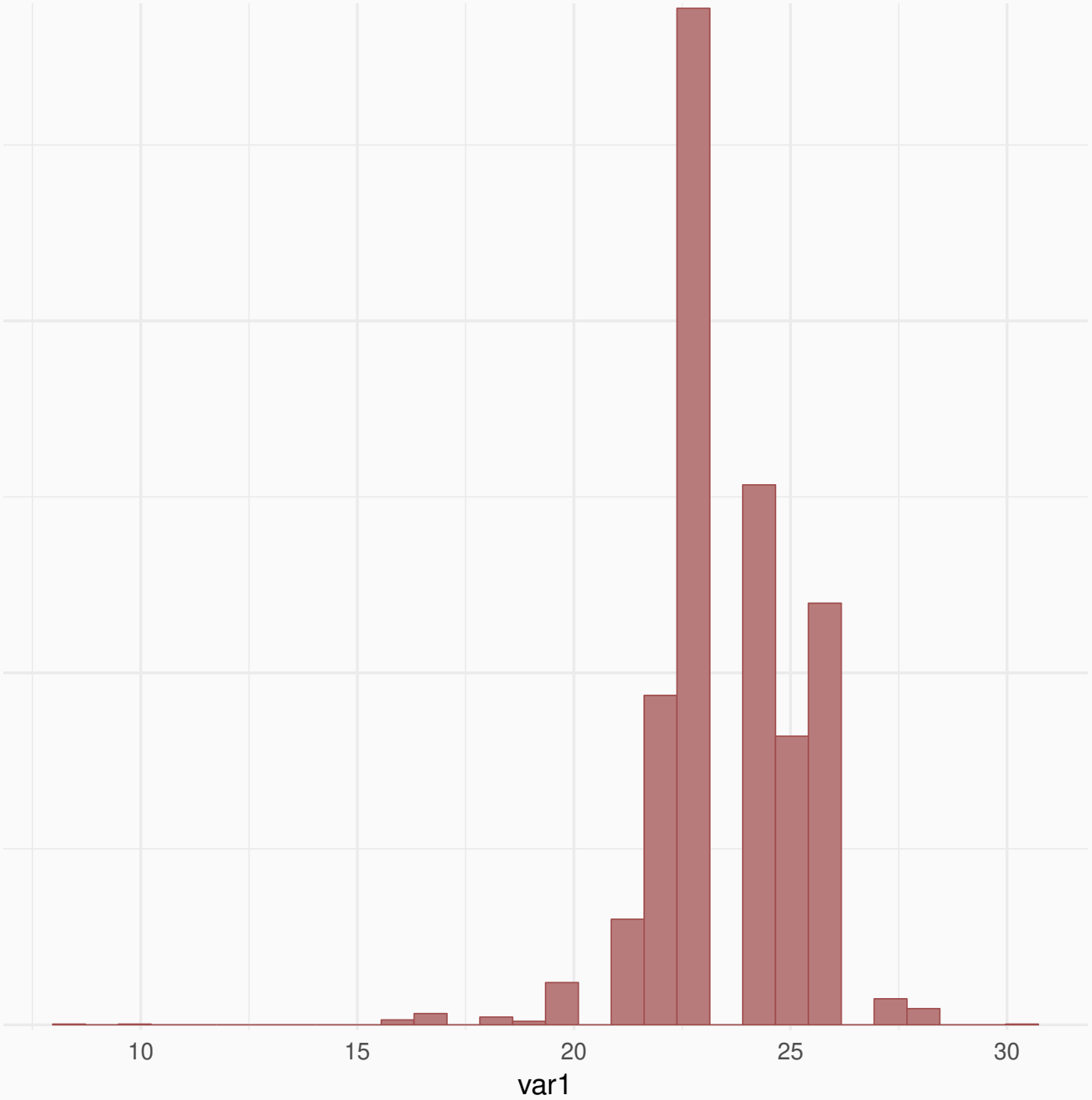
```
> theta = mcmc(cbind(x$lambda,x$phi,x$m))
> theta = window(theta, start=1001)
> colnames(theta) = names(x)
> summary(theta)

Iterations = 1001:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 4000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean     SD Naive SE Time-series SE
lambda  2.025 0.3036 0.004801       0.005302
phi     5.068 0.5827 0.009214       0.009872
m      23.614 1.6418 0.025959       0.029750
```

2. Quantiles for each variable:

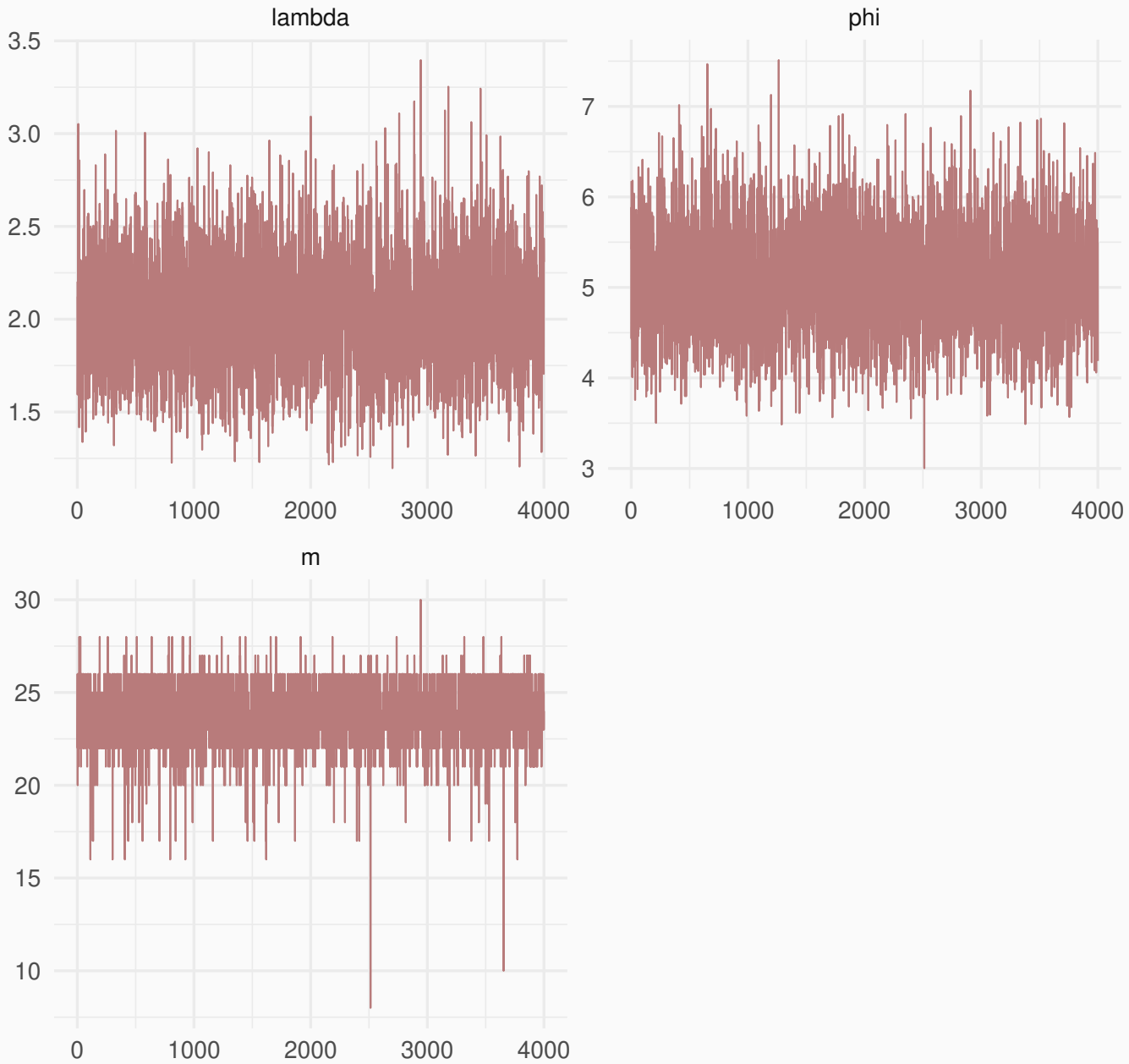
| | 2.5% | 25% | 50% | 75% | 97.5% |
|---|---|---|---|---|---|
| lambda | 1.470 | 1.817 | 2.011 | 2.218 | 2.653 |
| phi | 3.975 | 4.667 | 5.044 | 5.455 | 6.239 |
| m | 20.000 | 23.000 | 23.000 | 25.000 | 26.000 |

Posterior probabilities of change point.

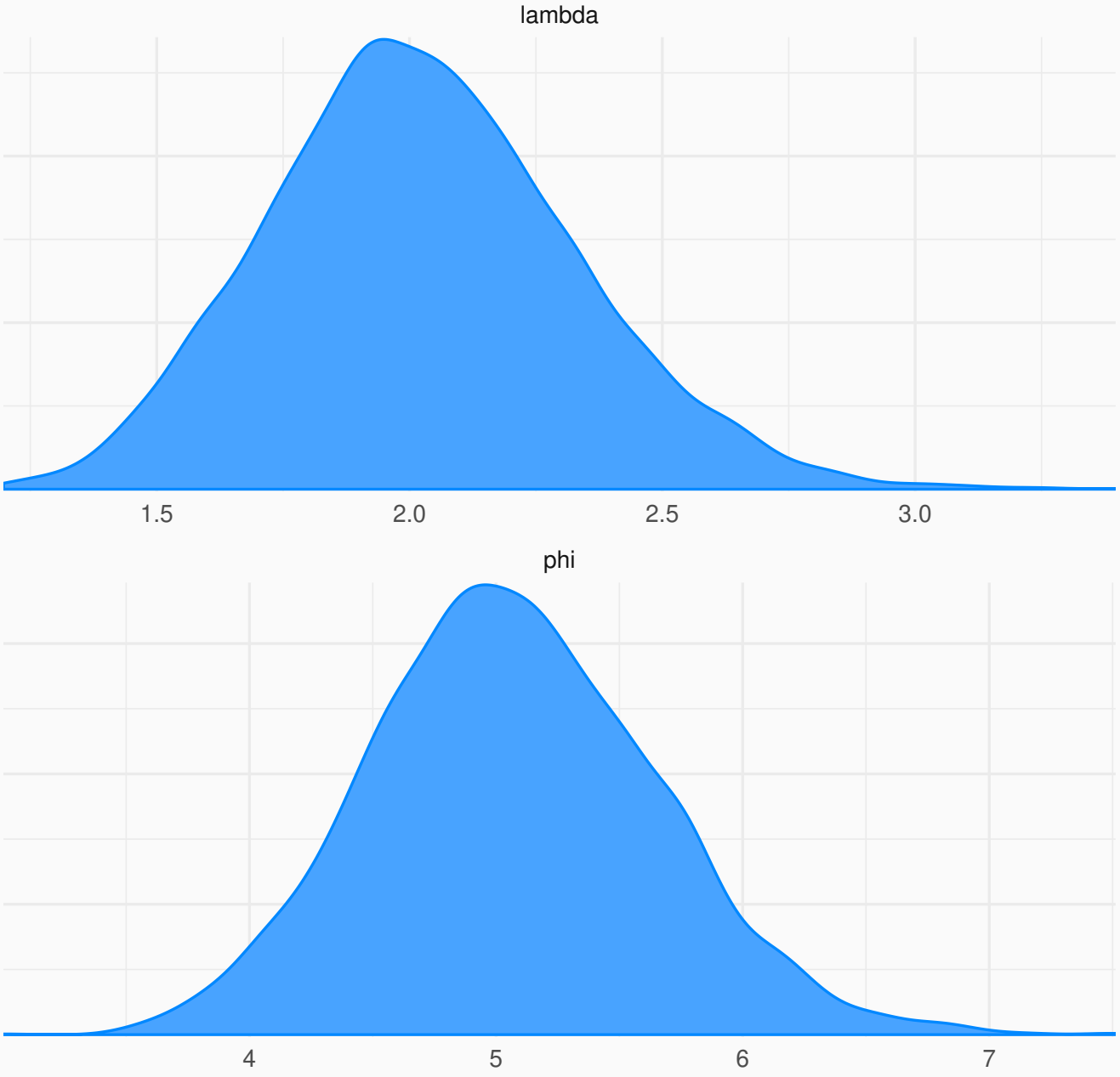
```
> tm= table(theta[,"m"])
> print(tm[1:10])

   8   10   16   17   18   19   20   21   22   23
   1    1    7   16   11    5   60  150  468 1444

> print(tm[11:length(tm)])

 24  25  26  27  28  30
767 410 599  37  23   1
```


10
15
20
25
30
var1

# Trace plots of MCMC draws



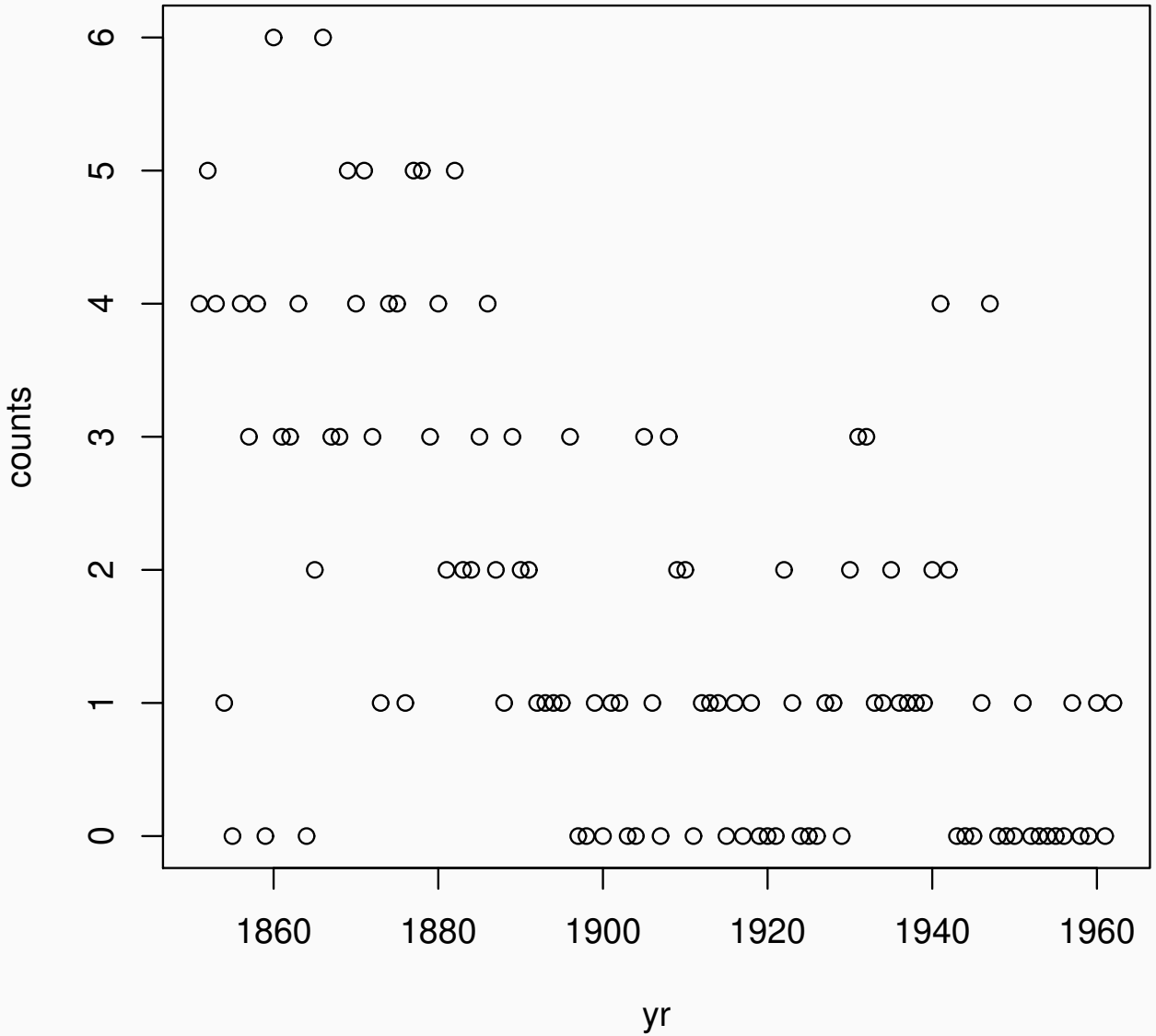
# Posterior distributions



**Example.** The data below refer to the number of accidents in coal mining in Great Britain (accidents involving at least 10 deaths) between 1851 and 1962. Is there a change point in this data set?

```
> yr= 1851:1962
> counts = c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,
+ 5,2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,
+ 1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,
+ 0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1)
```

```
> plot(yr, counts)
```

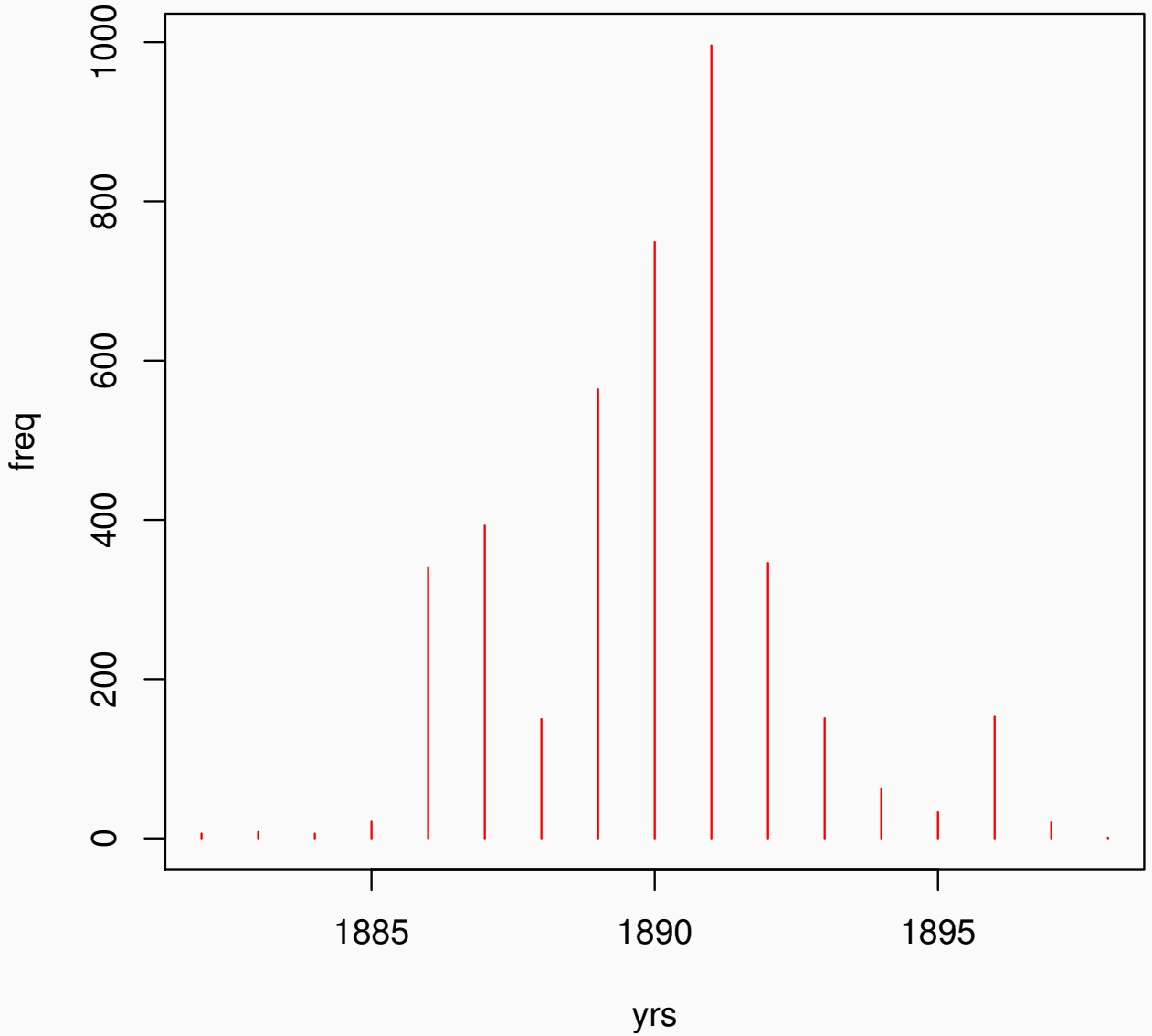


```
> x = Gibbs(a=0.1,b=0.1,c=0.1,d=0.1,y=counts,niter=5000)
> theta = mcmc(cbind(x$lambda,x$phi,x$m))
> theta = window(theta, start=1001)
> colnames(theta) = names(x)
> summary(theta)

Iterations = 1001:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 4000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean     SD Naive SE Time-series SE
lambda  3.1087 0.2910 0.004601       0.004896
phi     0.9223 0.1155 0.001826       0.001927
m      39.9940 2.4386 0.038558       0.043829
```

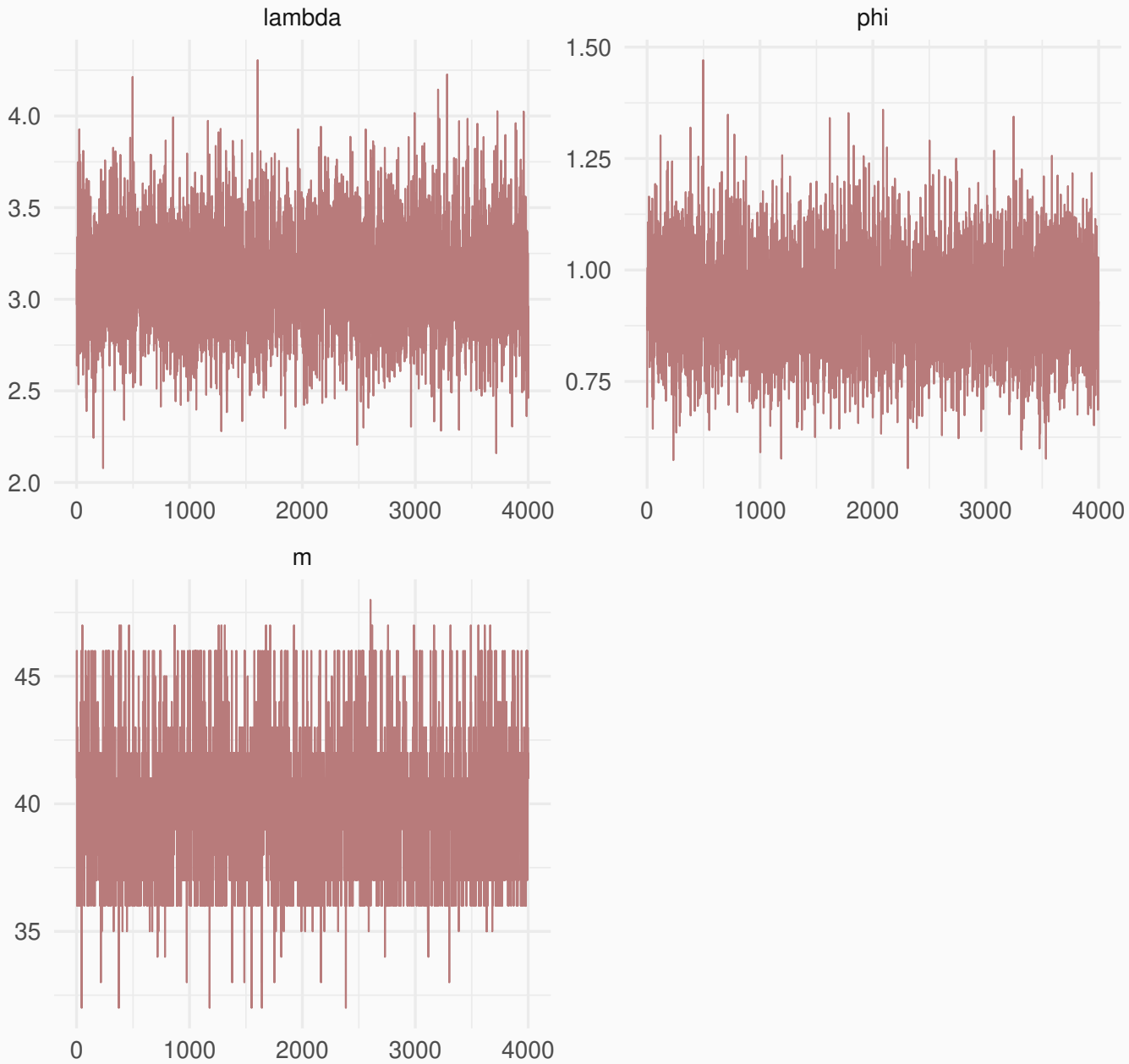
2. Quantiles for each variable:

| | 2.5% | 25% | 50% | 75% | 97.5% |
|---|---|---|---|---|---|
| lambda | 2.5656 | 2.9009 | 3.1040 | 3.301 | 3.708 |
| phi | 0.7086 | 0.8408 | 0.9188 | 0.996 | 1.160 |
| m | 36.0000 | 39.0000 | 40.0000 | 41.000 | 46.000 |

Posterior probabilities of change point.



# Trace plots of MCMC draws



**Example.** Let $X_1, \ldots, X_n$ be a random sample from the Student-$t$ distribution with location $\theta$, scale $\sigma$ (known) and degrees of freedom $\nu > 0$.

The likelihood function is,

$$p(\mathbf{x}|\nu, \theta, \sigma^2) = \prod_{i=1}^{n} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu}\,\sigma}\left[1 + \frac{(x-\theta)^2}{\nu\sigma^2}\right]^{-(\nu+1)/2}.$$

The model can be rewritten by creating latent variables, $\lambda_1, \ldots, \lambda_n$,

$$\begin{aligned} X_i|\theta, \sigma, \lambda_i &\sim N\left(\theta, \frac{\sigma^2}{\lambda_i}\right) \\ \lambda_i|\nu &\sim Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \end{aligned}$$

for $i = 1, \ldots, n$.

The complete likelihood is given by,

$$f(\mathbf{x}, \boldsymbol{\lambda}|\theta, \nu) = f(\mathbf{x}|\theta, \boldsymbol{\lambda}) f(\boldsymbol{\lambda}|\nu).$$

Specifying prior distributions, $\theta \sim N(\mu_0, \tau_0^2)$ and $\nu \sim Exp(\beta)$ we have the following posterior density,

$$
\begin{aligned}
p(\theta, \nu, \boldsymbol{\lambda}|\mathbf{x}) &\propto f(\mathbf{x}|\theta, \boldsymbol{\lambda}) f(\boldsymbol{\lambda}|\nu) f(\theta) f(\nu) \\
&\propto \prod_{i=1}^{n} (2\pi\sigma^2/\lambda_i)^{-1/2} \exp\left\{ -\frac{\lambda_i}{2\sigma^2}(x_i - \theta)^2 \right\} \\
&\quad \prod_{i=1}^{n} \frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \lambda_i^{\nu/2-1} \exp(-\nu\lambda_i/2) \\
&\quad (2\pi\tau_0^2)^{-1/2} \exp\left\{ -\frac{1}{2\tau_0^2}(\theta - \mu_0)^2 \right\} \exp(-\beta\nu).
\end{aligned}
$$

So, the complete conditional densities are,

$$p(\boldsymbol{\lambda}|\theta,\nu,\mathbf{x}) \quad \propto \quad \prod_{i=1}^{n} \lambda_i^{1/2} \exp\left\{-\frac{\lambda_i}{2\sigma^2}(x_i-\theta)^2\right\} \lambda_i^{\nu/2-1} \exp(-\nu\lambda_i/2)$$

so that,

$$\lambda_i|\lambda_{-i},\theta,\nu,\mathbf{x} \sim \mathit{Gamma}\left(\frac{\nu+1}{2}, \frac{\nu}{2}+\frac{(x_i-\theta)^2}{2\sigma^2}\right)$$

$$p(\theta|\boldsymbol{\lambda},\nu,\mathbf{x}) \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\lambda_i(x_i-\theta)^2\right\}\exp\left\{-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2\right\}$$

$$\propto \quad \exp\left\{-\frac{1}{2\sigma^2}\left(\theta^2\sum_{i=1}^{n}\lambda_i - 2\theta\sum_{i=1}^{n}\lambda_i x_i\right) - \frac{1}{2\tau_0^2}\left(\theta^2-2\theta\mu_0\right)\right\}$$

$$\propto \quad \exp\left\{-\frac{1}{2}\left[\theta^2(\sigma^{-2}\sum_{i=1}^{n}\lambda_i+\tau_0^{-2}) - 2\theta(\sigma^{-2}\sum_{i=1}^{n}\lambda_i x_i+\tau_0^{-2}\mu_0)\right]\right\}$$

so that,

$$\theta|\boldsymbol{\lambda},\nu,\mathbf{x} \sim N(\mu_1,\tau_1^2)$$

$$\tau_1^{-2} = \sigma^{-2}\sum_{i=1}^{n}\lambda_i + \tau_0^{-2} \qquad \mu_1 = \frac{\sigma^{-2}\sum_{i=1}^{n}\lambda_i x_i + \tau_0^{-2}\mu_0}{\sigma^{-2}\sum_{i=1}^{n}\lambda_i + \tau_0^{-2}}$$

Finally,

$$p(\nu|\theta, \boldsymbol{\lambda}, \mathbf{x}) \quad \propto \quad \prod_{i=1}^{n} \frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \lambda_i^{\nu/2-1} \exp(-\nu\lambda_i/2) \ \exp(-\beta\nu).$$

which is of unknown functional form.

**Example.** Testing with 20 simulated data from a $t$-Student with $\nu = 6$ and $\sigma^2 = 1$.

```
> set.seed(1234)
> x = round(rt(n=n, df = nu),4)
```

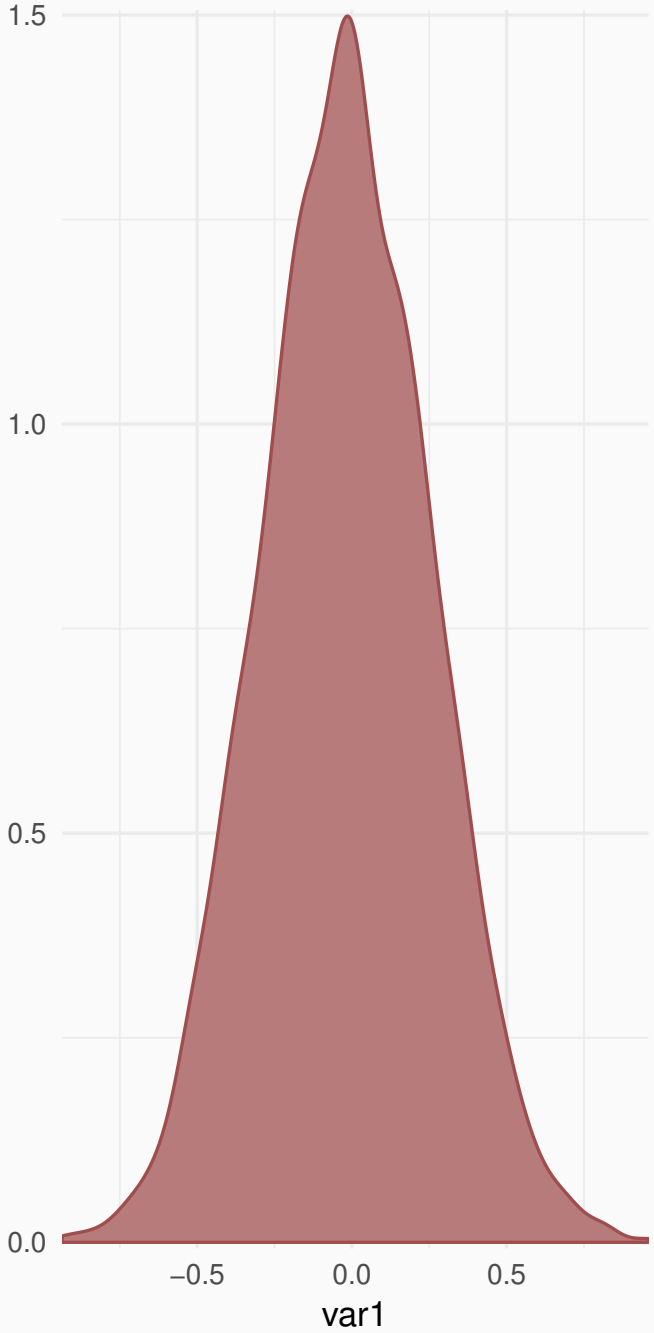
An outlier,

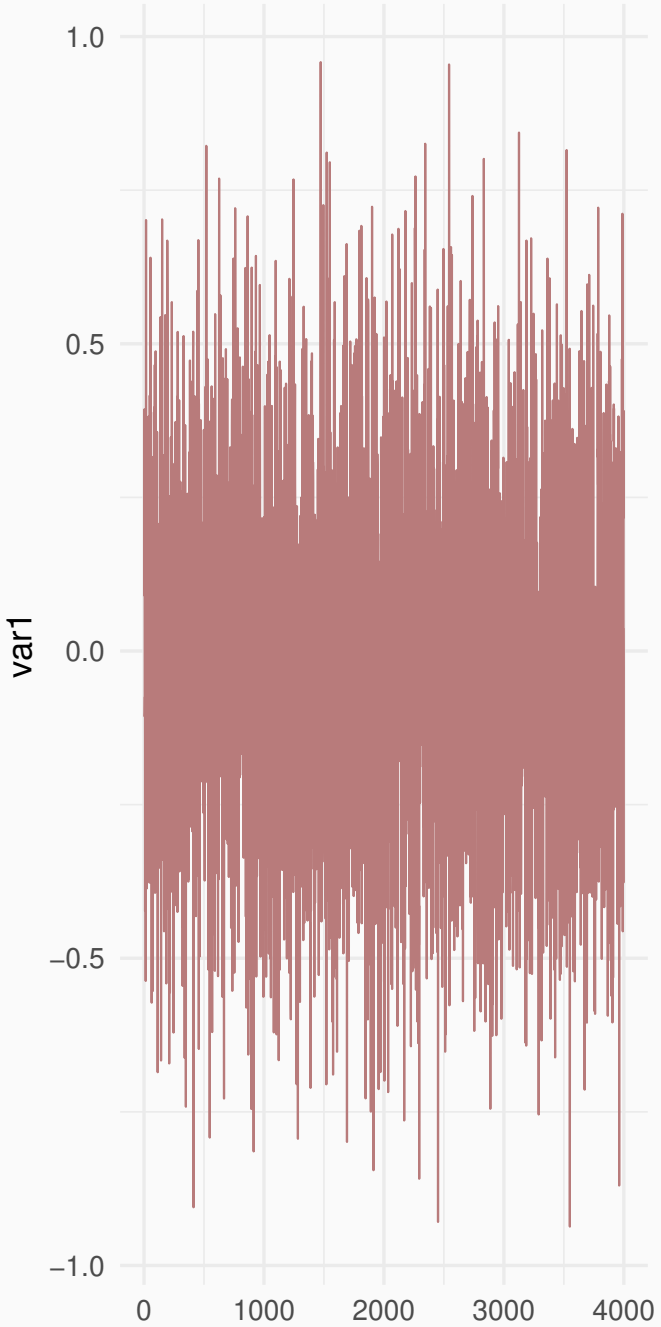
```
> x[2]= 5 * sqrt(nu/(nu-2)) * x[2]
```

```
> options(width=50)
> round(x,3)
```

```
 [1] -1.216  3.584  0.700 -1.358  0.850  0.339
 [7] -0.034 -0.542  0.009  1.216  0.488 -1.028
[13]  0.982 -1.214 -1.755  0.243 -1.172 -2.216
[19]  2.775  1.008
```

```
> mu0 = 0
> tau0= 10
> sigma2= 1
> niter = 5000
> nburn = 1000
> theta = array(0,niter)
> lambda= matrix(0,nrow=niter, ncol= 20)
> theta[1] = mean(x)
> lambda[1,] = 1
> for (i in 2:niter) {
+   s1 = sum(lambda[i-1,] )
+   s2 = sum(lambda[i-1,] * x)
+   tau1= 1 / ((1/tau0) + s1/sigma2)
+   mu1 = (s2/sigma2 + mu0/tau0) * tau1
+   theta[i] = rnorm(1,mean=mu1, sd=sqrt(tau1))
+   for (j in 1:20) {
+     lambda[i,j]=rgamma(1,(nu+1)/2,nu/2+(x[j]-theta[i])^2/(2*sigma2))
+   }
+ }
```


1.5
1.0
0.5
0.0
−0.5
0.0
0.5
var1


1.0
0.5
0.0
−0.5
−1.0
var1
0
1000
2000
3000
4000

0
1
2
3
4
5
1
3
5
7
9
11
13
15
17
19

# Metropolis-Hastings Algorithms

Metropolis-Hastings algorithms use the same idea of rejection methods. A value is generated from an auxiliary distribution and is accepted with a certain probability.

This correction mechanism ensures convergence of the chain to the equilibrium distribution.

## Definition

Suppose that the chain is at state $x$ and a value $x'$ is generated from a proposal distribution $q(\cdot|x)$. Note that the proposal distribution may depend on the current state of the chain, for example $q(\cdot|x)$ could be a normal distribution centered on $x$. The new value $x'$ is accepted with probability,

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')\ q(x|x')}{\pi(x)\ q(x'|x)}\right\}. \tag{1}$$

where $\pi$ is the target distribution.

- We only need to know $\pi$ up to a constant as the acceptance probability does not change.
- This is important in Bayesian applications where we do not know the posterior density completely.
- The chain may remain the same state for many iterations and in practice we need to monitor this by computing the proportion of iterations that new values are accepted.

In practical terms, the Metropolis-Hastings algorithm can be specified by the following steps,

1. Initialize the iterations counter $t = 0$ and specify an initial value $x^{(0)}$.
2. Generate a new value $x'$ from the distribution $q(\cdot|x)$.
3. Compute the acceptance probability $\alpha(x, x')$ and generate $u \sim U(0, 1)$.
4. If $u \leq \alpha$ then accept the new value and set $x^{(t+1)} = x'$, otherwise reject and set $x^{(t+1)} = x^{(t)}$.
5. Change the counter $t$ to $t + 1$ and return to step 2.

- The proposal distribution can be chosen arbitrarily but in practice we need to be careful to improve efficiency.

- In Bayesian applications the target is the posterior density, $\pi = p(\theta|x)$ and the acceptance probability assumes a particular form,

$$\alpha(\theta, \theta') = \min \left\{ 1, \frac{p(x|\theta')}{p(x|\theta)} \frac{p(\theta')}{p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \right\}. \tag{2}$$

**Example.** In a certain population it is known that each animal may belong to one of 4 genetic lineage with probabilities,

$$p_1 = \frac{1}{2} + \frac{\theta}{2},\ p_2 = \frac{1-\theta}{4},\ p_3 = \frac{1-\theta}{4},\ p_4 = \frac{\theta}{4}.$$

where $0 < \theta < 1$ is unknown. For any $\theta \in (0,1)$ it is easy to verify that $p_i > 0,\ i = 1,2,3,4$ and $p_1 + p_2 + p_3 + p_4 = 1$.

When observing $n$ animals from which $y_i$ belong to lineage $i$ the random vector $\mathbf{Y} = (y_1, y_2, y_3, y_4)$ has a multinomial distribution with parameters $n, p_1, p_2, p_3, p_4$ and then,

$$\begin{aligned} p(\mathbf{y}|\theta) &= \frac{n!}{y_1!y_2!y_3!y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4} \\ &\propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}. \end{aligned}$$

Assigning a prior distribution $\theta \sim U(0,1)$ it follows that the posterior density is proportional to the above expression,

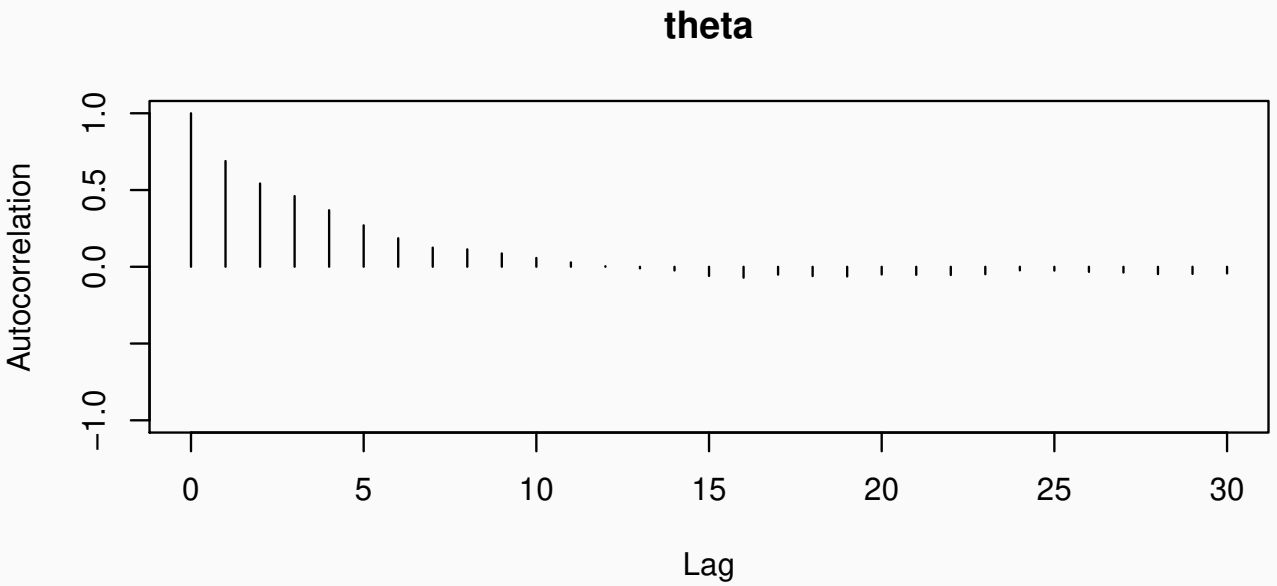
$$p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta) \propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}.$$

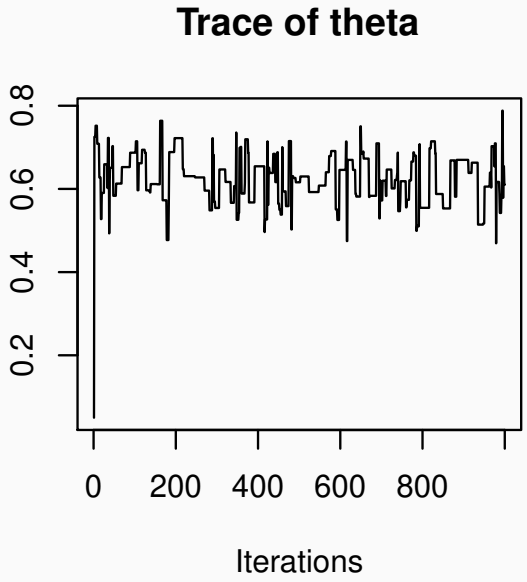
Taking the $U(0,1)$ as the proposal distribution then $q(\theta) = 1, \forall\ \theta$ and the acceptance probability simplifies to,

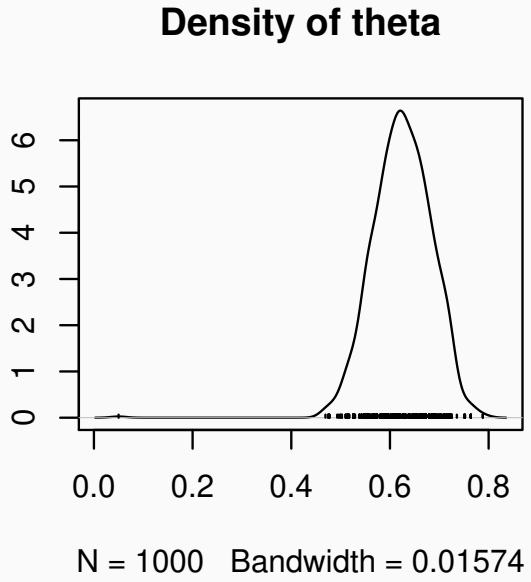
$$\begin{aligned}
\alpha(\theta, \theta') &= \min\left\{1, \frac{p(x|\theta')}{p(x|\theta)}\right\} \\
&= \min\left\{1, \left(\frac{2+\theta'}{2+\theta}\right)^{y_1}\left(\frac{1-\theta'}{1-\theta}\right)^{y_2+y_3}\left(\frac{\theta'}{\theta}\right)^{y_4}\right\}.
\end{aligned}$$

```
> p <- function(x,y)  (2+x)^y[1]*(1-x)^(y[2]+y[3])*x^y[4]

> metr <- function(n,y,p,start){
+  theta = matrix(NA, nrow=n)
+  theta[1] = start
+  taxa = 0
+  for (i in 2:n){
+  x    = runif(1)
+  A    = p(x,y)/p(theta[i-1],y)
+  prob = min(1,A)
+  u = runif(1)
+  if (u < prob) {
+    theta[i] = x
+    taxa = taxa + 1
+    }
+  else theta[i] = theta[i-1]
+  }
+  taxa = taxa/n
+  return(list(theta=theta,taxa=round(taxa,2)))
+ }
```

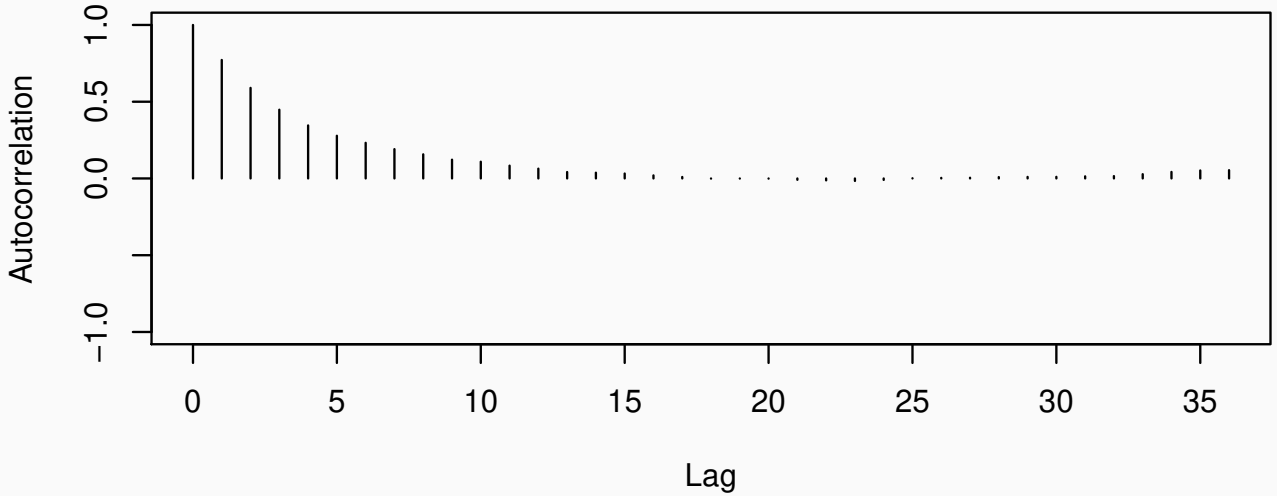
```
> m = metr(n=1000,y=c(125,18,20,34),p,start=0.05)
> theta = as.mcmc(m$theta)
> colnames(theta)="theta"
> summary(theta)

Iterations = 1:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean             SD       Naive SE
      0.624844       0.059130       0.001870
Time-series SE
      0.005437

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.5142 0.5878 0.6276 0.6686 0.7221
```

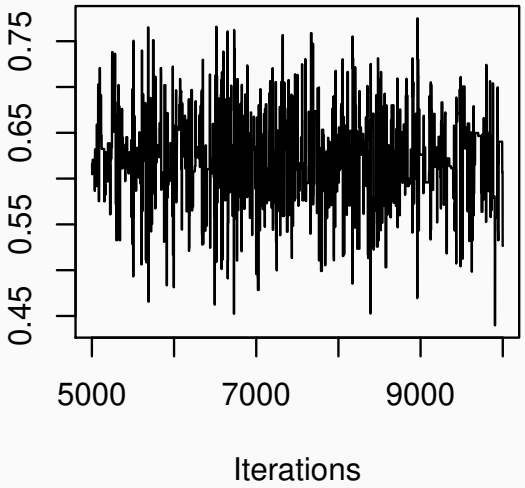

theta
Autocorrelation
1.0
0.5
0.0
−1.0
0
5
10
15
20
25
30
Lag

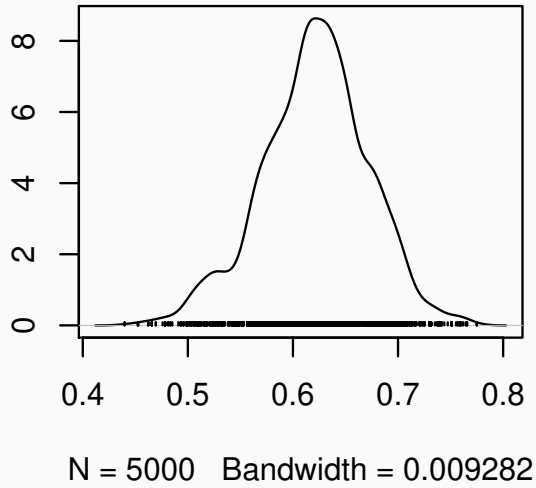

Trace of theta
0.8
0.6
0.4
0.2
0
200
400
600
800
Iterations


Density of theta
6
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
N = 1000 Bandwidth = 0.01574

10000 simulations discarding the first 5000.



Obtaining a sample of probabilities $(p_1, p_2, p_3, p_4)$.

```
> p1 = 1/2+theta/4
> p2 = (1-theta)/4
> p3 = p2
> p4 = theta/4
> prob  = as.mcmc(cbind(p1,p2,p3,p4))
> colnames(prob)=c("p1","p2","p3","p4")
```

```
> round(summary(prob)$statistics,6)

       Mean       SD Naive SE Time-series SE
p1 0.655257 0.012655 0.000179       0.000514
p2 0.094743 0.012655 0.000179       0.000514
p3 0.094743 0.012655 0.000179       0.000514
p4 0.155257 0.012655 0.000179       0.000514

> round(summary(prob)$quantile,6)

       2.5%      25%      50%      75%    97.5%
p1 0.627635 0.646995 0.655794 0.663108 0.678478
p2 0.071522 0.086892 0.094206 0.103005 0.122365
p3 0.071522 0.086892 0.094206 0.103005 0.122365
p4 0.127635 0.146995 0.155794 0.163108 0.178478
```

**Example.** In the previous example the algorithm can become more efficient using a reparameterization to the real line. Using the logit transformation,

$$\phi = \log\left(\frac{\theta}{1-\theta}\right) \in \mathbb{R}$$

with inverse transformation,

$$\theta = \frac{\exp(\phi)}{1+\exp(\phi)}.$$

The prior must be in the same transformed scale. If $\theta \sim U(0,1)$ the density function of $\phi$ is,

$$p(\phi) = \left|\frac{d\theta}{d\phi}\right| = \frac{\exp(\phi)}{(1+\exp(\phi))^2}.$$

Values of $\phi$ can be now proposed as $\phi'|\phi \sim N(\phi, 1)$ so that,

$$\frac{q(\phi|\phi')}{q(\phi'|\phi)} = 1$$

The acceptande probability is,

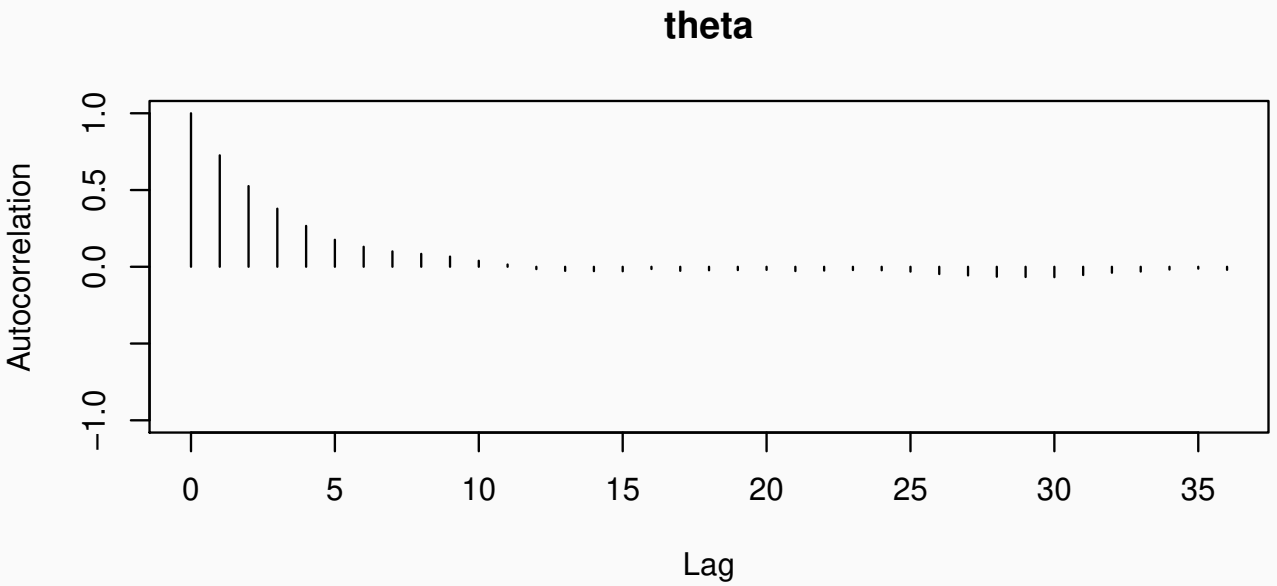
$$\alpha(\theta, \theta') = \min\left\{1, \frac{p(x|\theta')}{p(x|\theta)}\frac{p(\phi')}{p(\phi)}\right\}$$

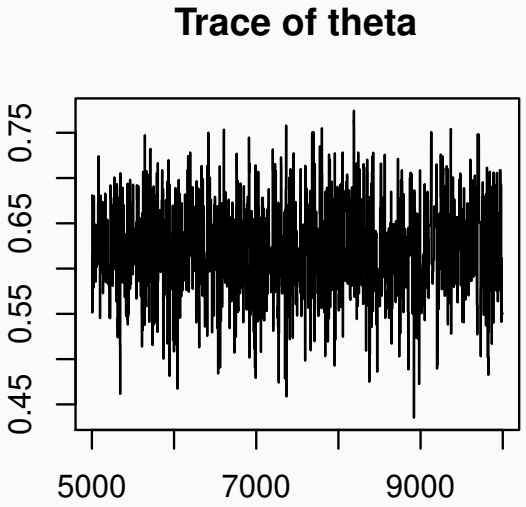
where,

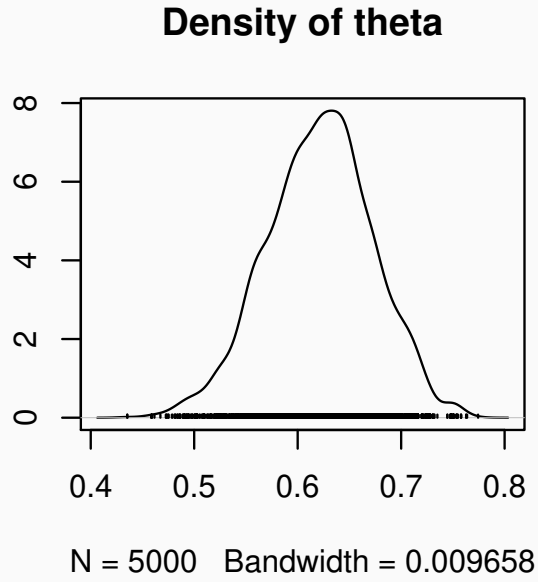
$$\begin{aligned}
\frac{p(x|\theta')}{p(x|\theta)} &= \left[\frac{2+\theta'}{2+\theta}\right]^{y_1}\left[\frac{1-\theta'}{1-\theta}\right]^{y_2+y_3}\left[\frac{\theta'}{\theta}\right]^{y_4} \\
\frac{p(\phi')}{p(\phi)} &= \frac{\exp(\phi')/(1+\exp(\phi'))^2}{\exp(\phi)/(1+\exp(\phi))^2}
\end{aligned}$$

```
> prior <- function(phi) exp(phi)/(1+exp(phi))^2

> metr1 <- function(niter,y,p,theta0) {
+  phi = matrix(NA, nrow=niter)
+  phi[1] = log(theta0/(1-theta0)); taxa=0
+  for (i in 2:niter) {
+    z    = exp(phi[i-1])/(1+exp(phi[i-1]))
+    old  = p(z,y)*prior(phi[i-1])
+    x    = rnorm(1,mean = phi[i-1], sd=1)
+    z    = exp(x)/(1+exp(x))
+    prob = min(1, p(z,y)*prior(x)/old)
+    u = runif(1)
+    if (u < prob) {
+      phi[i] = x
+      taxa = taxa + 1
+    } else {
+      phi[i] = phi[i-1]
+    }
+  }
+  theta = exp(phi)/(1+exp(phi))
+  taxa = taxa/niter
+  return(list(theta=theta,taxa=taxa))
+ }
```

```
Iterations = 5001:10000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean             SD       Naive SE
     0.6213011      0.0510327      0.0007217
Time-series SE
     0.0016679

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.5195 0.5869 0.6223 0.6539 0.7158
```


theta
Autocorrelation
1.0
0.5
0.0
−1.0
0
5
10
15
20
25
30
35
Lag


Trace of theta
0.75
0.65
0.55
0.45
5000
7000
9000
Iterations


Density of theta
8
6
4
2
0
0.4
0.5
0.6
0.7
0.8
N = 5000 Bandwidth = 0.009658

```
Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

      Mean      SD  Naive SE Time-series SE
p1 0.65533 0.01276 0.0001804       0.000417
p2 0.09467 0.01276 0.0001804       0.000417
p3 0.09467 0.01276 0.0001804       0.000417
p4 0.15533 0.01276 0.0001804       0.000417

2. Quantiles for each variable:
```

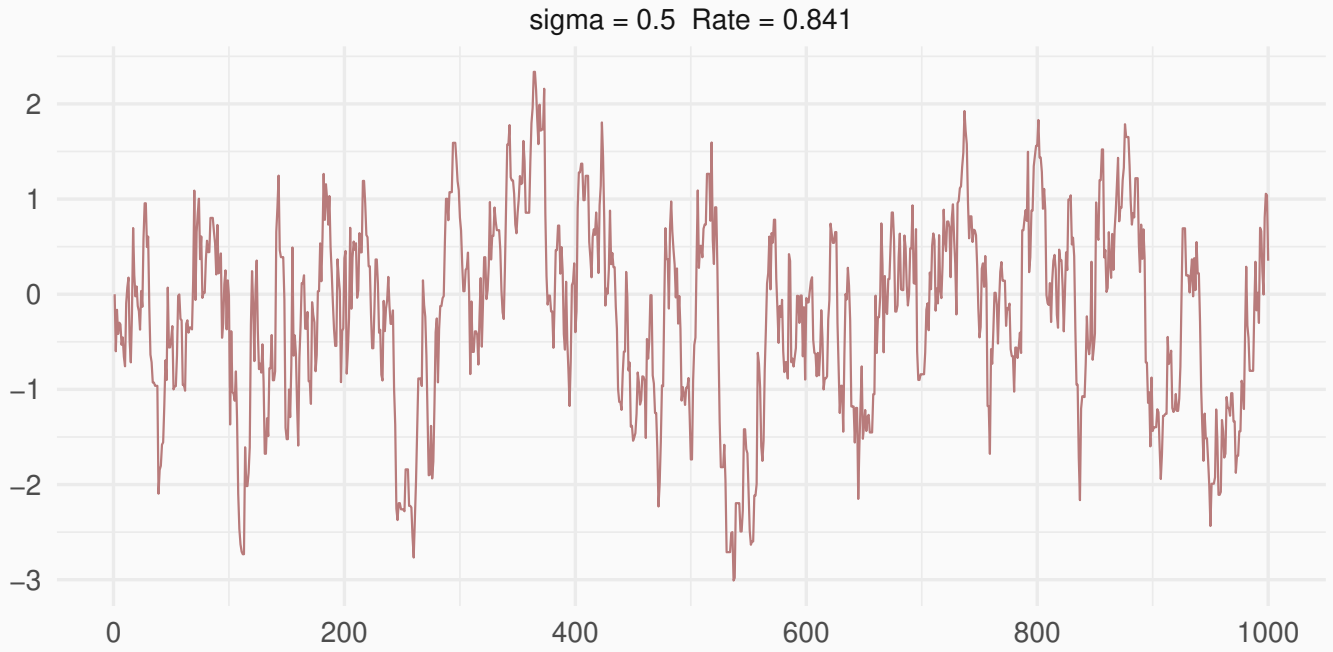
```
      2.5%     25%     50%    75%  97.5%
p1 0.62989 0.64672 0.65557 0.6635 0.6790
p2 0.07104 0.08651 0.09443 0.1033 0.1201
p3 0.07104 0.08651 0.09443 0.1033 0.1201
p4 0.12989 0.14672 0.15557 0.1635 0.1790
```

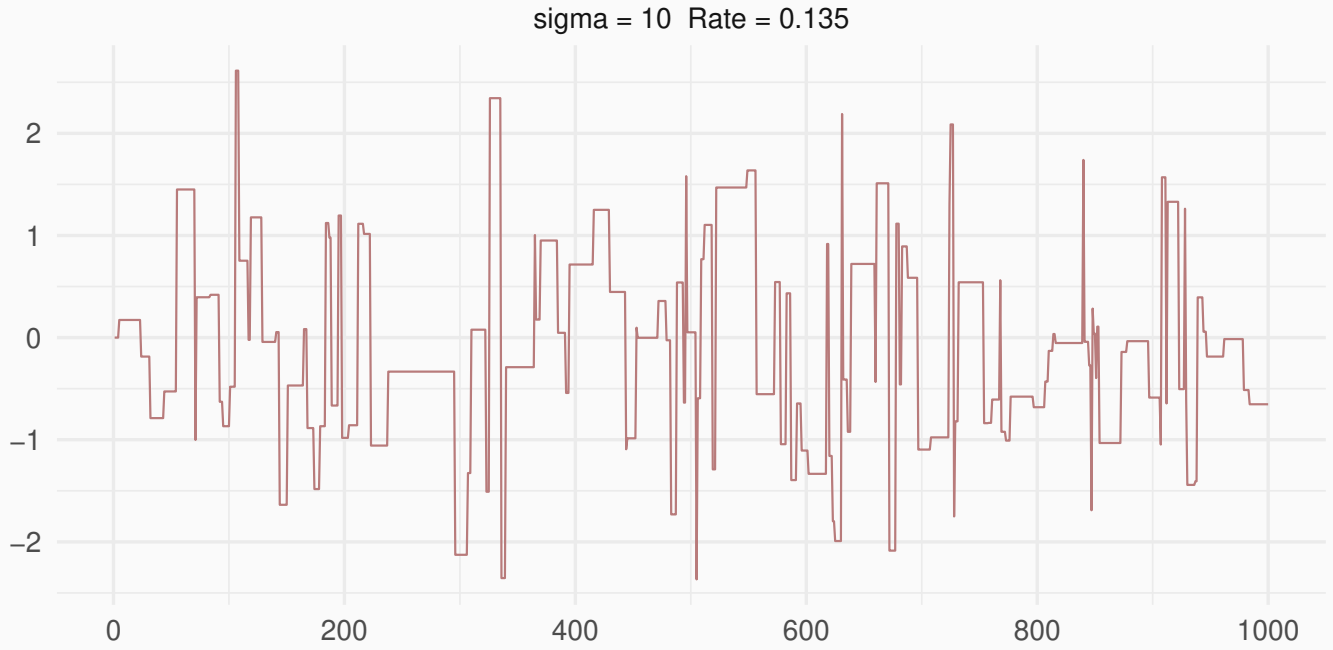
**Example.** Suppose that we want to simulate values $X \sim N(0,1)$ proposing valores $Y \sim N(x, \sigma^2)$.

The acceptance probability is,

$$\begin{aligned}
\alpha(x,y) &= \min\left\{1, \frac{f_N(y|0,1)}{f_N(x|0,1)} \frac{f_N(x|y,\sigma^2)}{f_N(y|x,\sigma^2)}\right\} \\
&= \min\left\{1, \frac{f_N(y|0,1)}{f_N(x|0,1)}\right\} \\
&= \min\left\{1, \exp\left(-\frac{1}{2}(y^2 - x^2)\right)\right\}.
\end{aligned}$$

```
> metrop <- function(n,sigma){
+  x = matrix(NA,nrow=n)
+  x[1] = rate = 0
+  for (i in 2:n){
+    y = rnorm(1,x[i-1],sigma)
+    prob = min(1,exp(-0.5*(y^2-x[i-1]^2)))
+    u = runif(1)
+    if (u < prob) {x[i]=y; rate=rate+1} else x[i]=x[i-1]
+  }
+  return(list(x=x,rate=rate/n))
+ }
```


sigma = 0.5  Rate = 0.841
2
1
0
−1
−2
−3
0
200
400
600
800
1000


sigma = 10  Rate = 0.135
2
1
0
−1
−2
0
200
400
600
800
1000

- Note that the value of $\sigma$ had a large impact on the acceptance rate of the algorithm.
- This is because with $\sigma = 0.5$ the proposal distribution is much closer to the target distribution than with $\sigma = 10$.
- In the two previous examples special cases were illustrated in which the proposal distribution does not depend on the current state of the chain or the dependence follows a random walk.

# Special Cases

# Independent Sampler

A particular case is when the proposal distribution does not depend on the current state of the chain, i.e. $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}')$.

The acceptance probability becomes,

$$\alpha(\mathbf{x}, \mathbf{x}') = \min\left\{1, \frac{\pi(\mathbf{x}')\ q(\mathbf{x})}{\pi(\mathbf{x})\ q(\mathbf{x}')}\right\}. \tag{3}$$

- In general, $q(\cdot)$ should be a good approximation of $\pi(\cdot)$, but it is safer if $q(\cdot)$ has heavier tails than $\pi(\cdot)$.
- The chain will not be independent since the acceptance probability still depends on $\mathbf{x}$.

If $\mathbf{x} \in \mathbb{R}^d$ a mostly used proposal distribution is $\mathbf{x}' \sim N(\mathbf{x}^*, \Sigma)$ where $\mathbf{x}^*$ is the mode of $\pi(\cdot)$ and

$$\Sigma = \tau \left[ -\frac{\partial^2 \log \pi(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right]^{-1}$$

evaluated at $\mathbf{x}^*$.

Note that the proposal distribution is fixed.

# The Metropolis Algorithm

Another particular case is called Metropolis algorithm and considers only symmetric proposal distributions, i.e. $q(x'|x) = q(x|x')$ for all values of $x$ and $x'$.

In this case the acceptance probability reduces to,

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}.$$

The Metropolis algorithm can also be based on a random walk so that the probability that the chain moves from $x$ to $x'$ depends only on the distance between them, i.e. $q(x'|x) = q(|x - x'|)$.

In this case, if the proposal distribution has variance $\sigma^2$ two extreme situations might occur,

1. if $\sigma^2$ is too small the values generated will be close to the current value and almost always will be accepted. However it will take many iterations for the algorithm to traverse all of $x$ space;
2. large values of $\sigma^2$ lead to an excessively high rejection rate and the chain moves very little.

In both situations the algorithm becomes inefficient and in practice we need to try different values of $\sigma^2$, monitoring the acceptance rate.

If $\mathbf{x} \in \mathbb{R}^d$, a possible proposal is $\mathbf{x}' \sim N(\mathbf{x}, \Sigma)$ where

$$\Sigma = \tau \left[ -\frac{\partial^2 \log \pi(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right]^{-1},$$

evaluated at $\mathbf{x}^*$.

Note that the proposal distribution now can change at each iteration.

# Block Updating

In general, $\mathbf{x} = (x_1, \ldots, x_d)'$ is a $d$-dimensional vector of parameters.

In this case, it can be computationally more efficient to divide $\mathbf{x}$ in $k$ blocks $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and within each iteration we have the algorithm applied $k$ times.

Define the vector $\mathbf{x}_{-i} = (\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_k)$ which contains all elements of $\mathbf{x}$ except $\mathbf{x}_i$.

Suppose that at iteration $t+1$ blocks $1, 2, \ldots, i-1$ have already been updated, i.e.

$$\mathbf{x}_{-i} = (\mathbf{x}_1^{(t+1)}, \ldots, \mathbf{x}_{i-1}^{(t+1)}, \mathbf{x}_{i+1}^{(t)}, \ldots, \mathbf{x}_k^{(t)}).$$

To update the $i$th component, a value of $\mathbf{x}_i$ is generated from the proposal distribution $q(\cdot|\mathbf{x}_i, \mathbf{x}_{-i})$ and this value is accepted with probability,

$$\alpha(\mathbf{x}_i, \mathbf{x}_i') = \min\left\{1, \frac{\pi(\mathbf{x}_i'|\mathbf{x}_{-i})\ q(\mathbf{x}_i|\mathbf{x}_i', \mathbf{x}_{-i}))}{\pi(\mathbf{x}_i|\mathbf{x}_{-i})\ q(\mathbf{x}_i'|\mathbf{x}_i, \mathbf{x}_{-i})}\right\}. \tag{4}$$

Note that $\pi(\mathbf{x}_i|\mathbf{x}_{-i})$ is the complete conditional distribution of $\mathbf{x}_i$.

Therefore, the Gibbs sampler is a special case of the Metropolis-Hastings algorithm, in which the elements of $\mathbf{x}$ are updated one at a time (or in blocks), taking the complete conditional as the proposal distribution and acceptance probability equal to 1.

**Example.** Let $X_1, \ldots, X_n \sim N(\mu, 1/\tau)$ where $\mu \sim N(0,1)$ and $\tau \sim Exp(1)$.

$$
\begin{aligned}
p(\mathbf{x}|\mu,\tau) &\propto \tau^{n/2} \exp\left\{-\frac{\tau}{2}\sum_{i=1}^{n}(x_i-\mu)^2\right\} \\
p(\mu) &\propto \exp(-\mu^2/2) \\
p(\tau) &= \exp(-\tau) \\
p(\mu,\tau|\mathbf{x}) &\propto \tau^{n/2} \exp\left\{-\frac{\tau}{2}\sum_{i=1}^{n}(x_i-\mu)^2\right\} \ \exp(-\mu^2/2)\ \exp(-\tau).
\end{aligned}
$$

One possible Metropolis-Hastings algorithm is,

1. generate $(\mu', \tau')$,
2. accept with probability,

$$\min\left\{1, \frac{p(\mathbf{x}|\mu', \tau')}{p(\mathbf{x}|\mu, \tau)} \; \frac{p(\mu')\; p(\tau')}{p(\mu)\; p(\tau)} \; \frac{q(\mu, \tau|\mu', \tau')}{q(\mu', \tau'|\mu, \tau)}\right\}$$

Assuming independent proposals we have $q(\mu, \tau) = q(\mu)q(\tau)$.

For example, using the prior distributions as proposals,

1. generate $\mu' \sim N(0, 1)$ and $\tau' \sim Exp(1)$,
2. accept with probabilities,

$$\min\left\{1, \frac{p(\mathbf{x}|\mu', \tau')}{p(\mathbf{x}|\mu, \tau)}\right\}.$$

```
> ll <- function(mu,tau,x)
+     sum(dnorm(x,mean=mu,sd=1/sqrt(tau),log=T))

> metrop <- function(x,mu0,tau0,s.mu,s.tau,niter) {
+   mu.vec  = tau.vec = numeric(niter)
+   mu  = mu0; tau = tau0; rate= 0
+   for (i in 1:niter) {
+     prop.mu = rnorm(1, mean=0, sd=1)
+     prop.tau= rexp (1, rate=1)
+     ratio = ll(prop.mu,prop.tau,x) - ll(mu,tau,x)
+     if (runif(1) < exp(ratio)) {
+       mu  = prop.mu
+       tau = prop.tau
+       rate= rate + 1
+     }
+     mu.vec [i] = mu
+     tau.vec[i] = tau
+   }
+   cat("Acceptance rate:", rate/niter,"\n")
+   return(cbind(mu=mu.vec,tau=1/tau.vec))}
```

**Example.** Testing with simulated data $X_1, \ldots, X_n \sim N(2, 4)$, with $n = 50$, 10000 iterations and 5000 burn-in.

```
> x = rnorm(50, mean=2, sd=2)
> m = metrop(x,mu0=0,tau0=1,s.mu=0.1,s.tau=0.1,niter=10000)

Acceptance rate: 0.0104
```

This acceptance rate is too low. One possible solution comes in the next example.

**Example.** In the previous example, reparameterize $\boldsymbol{\theta} = (\mu, \log(\tau))$ where,

$$p(\log(\tau)) = \exp(-\tau)\tau$$

and the posterior density is,

$$p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\mathbf{x}|\mu, \tau)\ p(\mu)\ p(\log(\tau)).$$

Propose new values as $\boldsymbol{\theta}' \sim N(\boldsymbol{\theta}^*, \Sigma)$ where $\boldsymbol{\theta}^*$ is the mode of $p(\boldsymbol{\theta}|\mathbf{x})$ and

$$\Sigma = \tau \left[ -\frac{\partial^2 \log \pi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \right]^{-1}$$

evaluated at $\boldsymbol{\theta}^*$.

Computing the mode and Hessian matrix using the R function optim()

```
> log.posterior <- function(theta,x) {
+   mu  = theta[1]
+   tau = exp(theta[2])
+   log.prior = -tau + log(tau) - mu^2/2
+   log.likelihood=(length(x)/2)*log(tau)+sum(-tau*(x-mu)^2/2)
+   log.prior + log.likelihood
+ }

> out=optim(par=c(0,0),fn=log.posterior,control=list(fnscale=-1)
+           hessian=T,x=x)
```

Mode,

```
[1]  1.770954 -1.542934
```

Hessian matrix,

```
           [,1]       [,2]
[1,] -11.687656   1.769176
[2,]   1.769176 -26.003801
```

variance matrix,

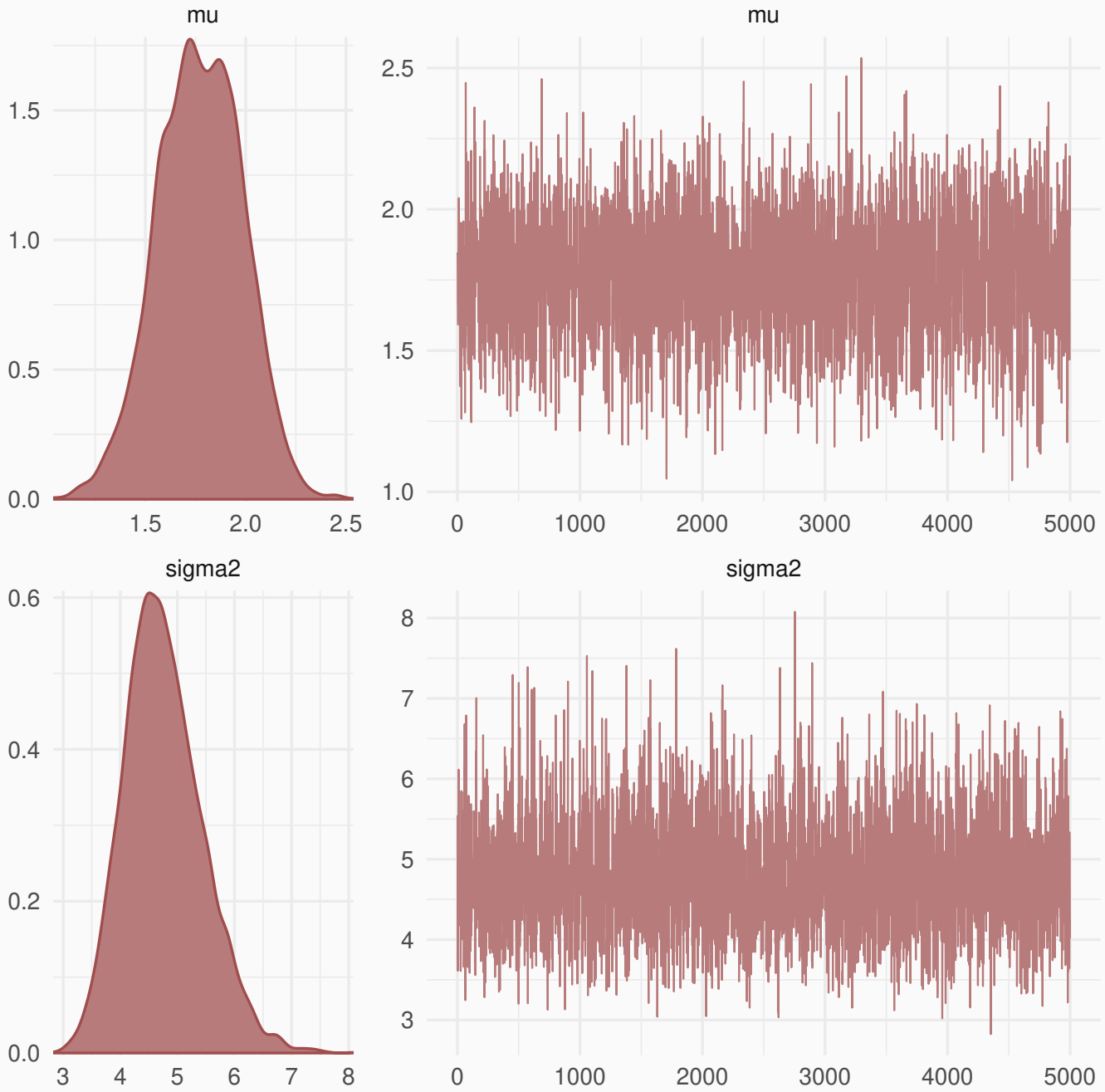
```
            [,1]        [,2]
[1,] 0.086450675 0.005881697
[2,] 0.005881697 0.038856079
```

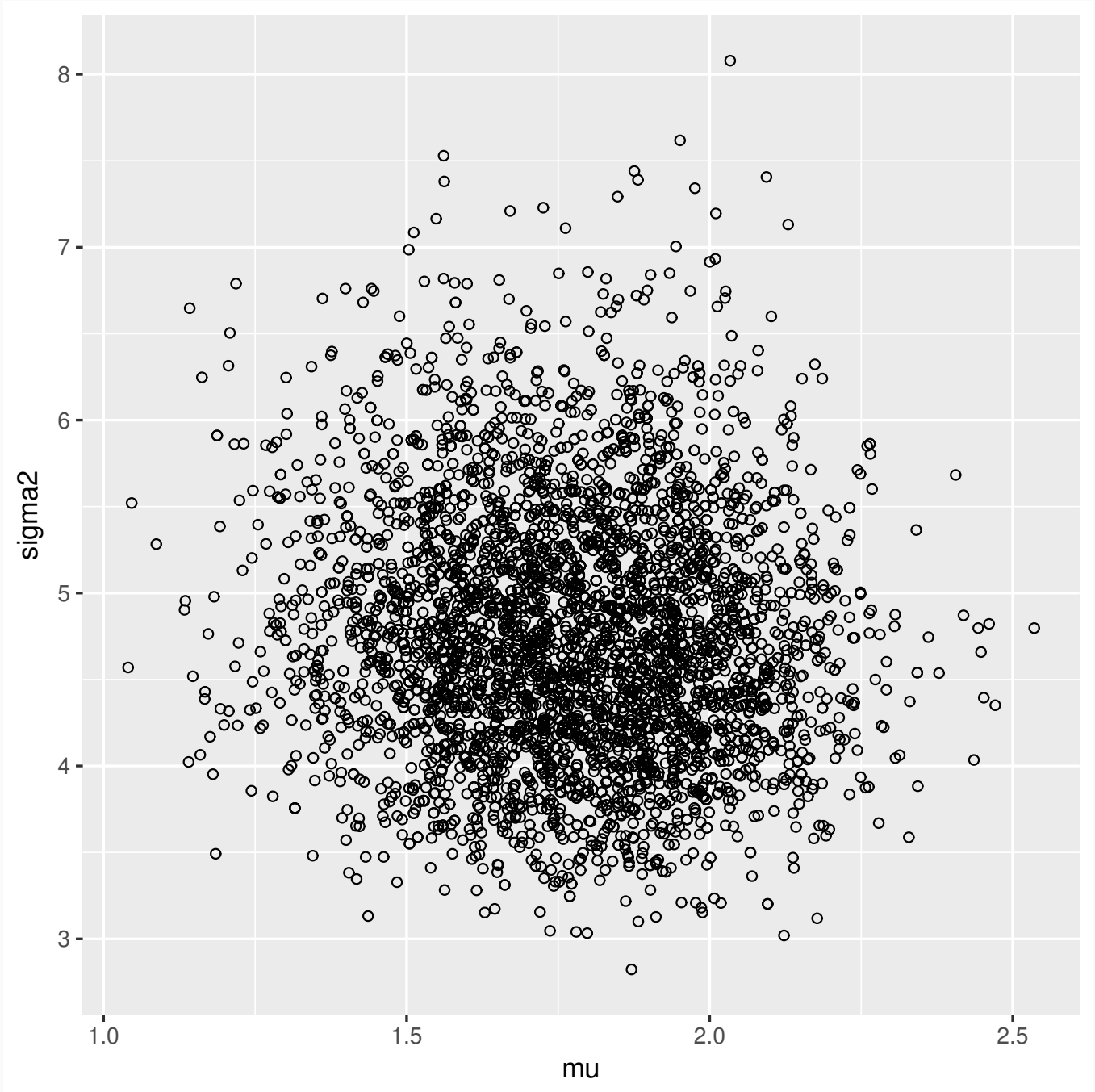
R functions to simulate values from multivariate normal and $t$-Student.

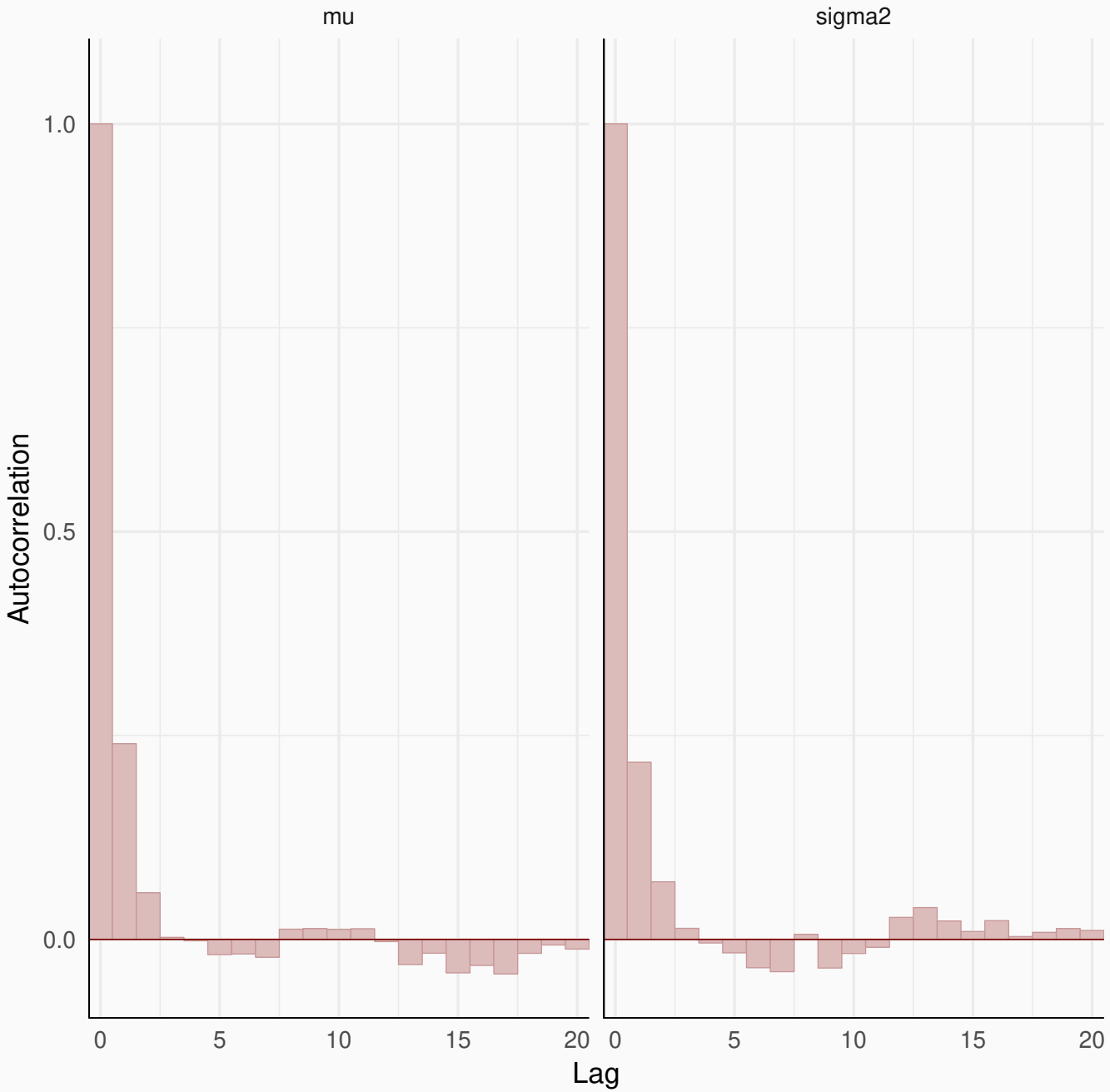
```
> library(mnormt)
> args(rmnorm)

function (n = 1, mean = rep(0, d), varcov, sqrt = NULL)
NULL

> library(MASS)
> args(mvrnorm)

function (n = 1, mu, Sigma, tol = 1e-06, empirical = FALSE, EISPACK = F
NULL
```

```
> metrop1 <- function(x,mu0,tau0,theta,Sigma,niter) {
+   mu.vec  = tau.vec = numeric(niter)
+   mu  = mu0; tau = log(tau0)
+   for (i in 1:niter) {
+     prop = rmnorm(n = 1, mean=theta,varcov=Sigma)
+     prop.mu = prop[1]
+     prop.tau= prop[2]
+     ratio = log.posterior(c(prop.mu,prop.tau),x)-
+             log.posterior(c(mu,tau),x)
+     if (runif(1) < exp(ratio)) {
+       mu = prop.mu
+       tau= prop.tau
+     }
+     mu.vec[i]  = mu
+     tau.vec[i] = exp(tau)
+   }
+   return(cbind(mu=mu.vec,sigma2=1/tau.vec))
+ }
```

```
> m=metrop1(x,mu0=0,tau0=1,theta=theta,Sigma=Sigma,niter=10000)
> m=as.mcmc(m[5001:10000,])
> summary(m)

Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean     SD Naive SE Time-series SE
mu     1.772 0.2105 0.002977       0.003804
sigma2 4.761 0.6850 0.009687       0.011792

2. Quantiles for each variable:

        2.5%   25%   50%   75% 97.5%
mu     1.352 1.623 1.773 1.926 2.167
sigma2 3.587 4.271 4.703 5.177 6.246
```


mu
1.5
1.0
0.5
0.0
1.5
2.0
2.5
mu
2.5
2.0
1.5
1.0
0
1000
2000
3000
4000
5000
sigma2
0.6
0.4
0.2
0.0
3
4
5
6
7
8
sigma2
8
7
6
5
4
3
0
1000
2000
3000
4000
5000


8
7
6
5
4
3
sigma2
1.0
1.5
2.0
2.5
mu


mu
sigma2
Autocorrelation
1.0
0.5
0.0
0
5
10
15
20
Lag

**Example.** Suppose now that the distribution of $X|\theta$ is such that,

$$p(x|\theta) = \frac{f(x,\theta)}{Z(\theta)},$$

where $Z(\theta)$ is a normalising constant assumed intractable. It is either not available analytically or uncomputable with finite computational resources.

The likelihood function is then given by,

$$p(\mathbf{x}|\theta) = \prod_{i=1}^{n} \frac{f(x_i,\theta)}{Z(\theta)},$$

Suppose we assign a prior distribution $p(\theta)$ and proposing a new value $\theta'$ with density $q(\cdot|\theta)$. The acceptance probability assumes a particular form,

$$\alpha(\theta, \theta') = \min \left\{ 1, \frac{\prod_{i=1}^{n} f(x_i, \theta')}{\prod_{i=1}^{n} f(x_i, \theta)} \frac{p(\theta')}{p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \frac{Z^n(\theta)}{Z^n(\theta')} \right\}, \qquad (5)$$

which is impossible to compute.