MCMC

Yonglin Zhuo

Result -1

Result -2

Result -3

Result -4

Result -5

Metropolis algorithm

Metropolis-Hasting

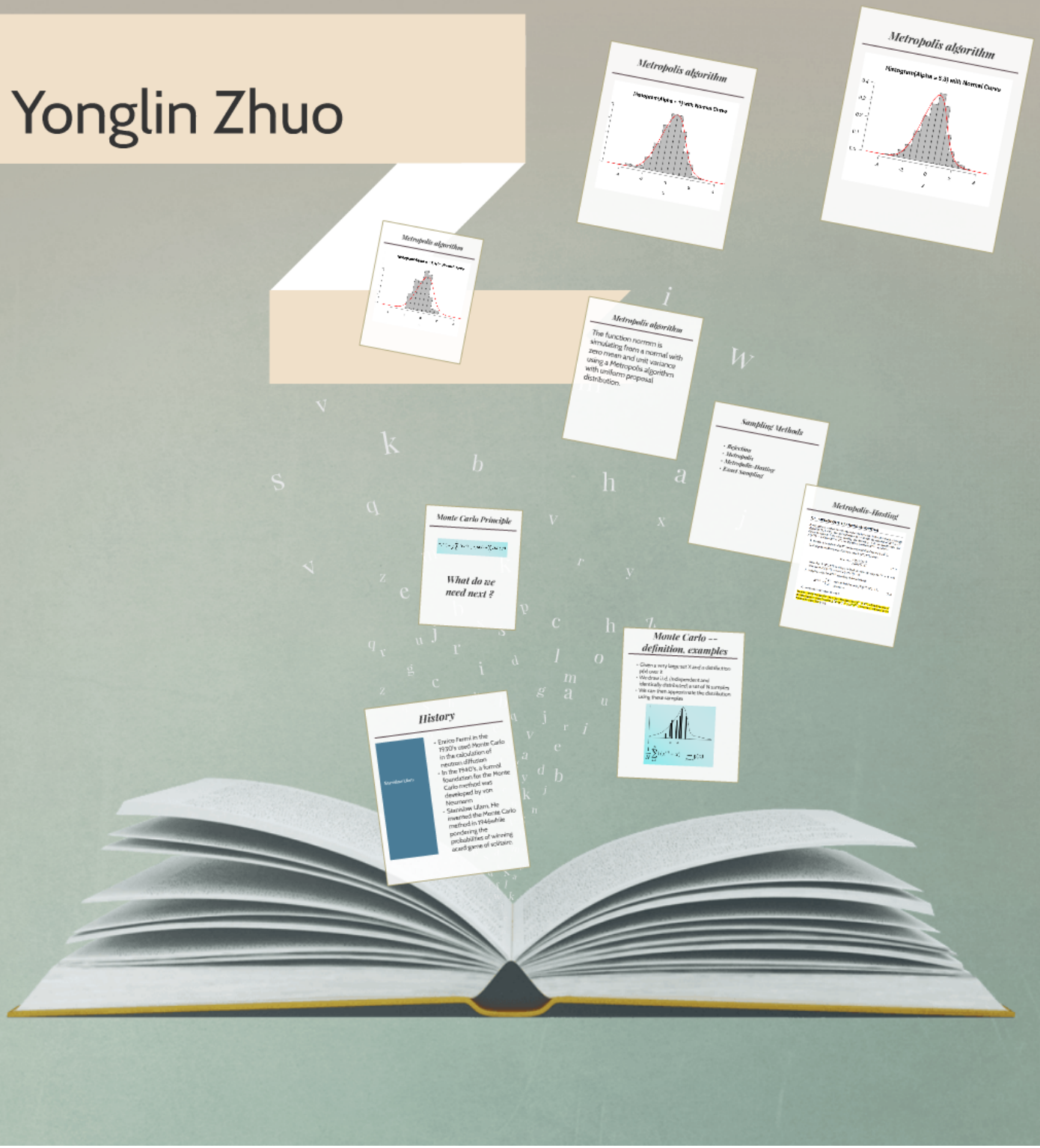
Metropolis-Hasting

Q and A

Tack

MCMC

Yonglin Zhuo



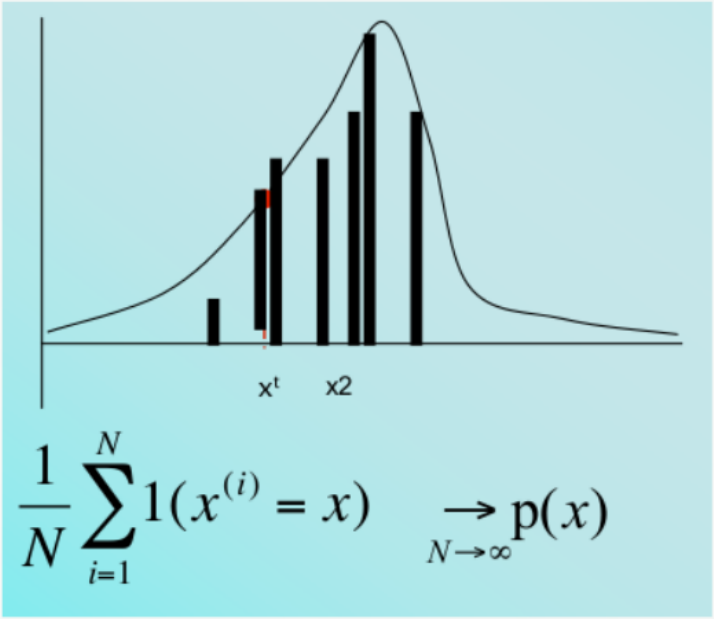
History

Stanislaw Ulam

- Enrico Fermi in the 1930's used Monte Carlo in the calculation of neutron diffusion
- In the 1940's, a formal foundation for the Monte Carlo method was developed by von Neumann
- Stanislaw Ulam, He invented the Monte Carlo method in 1946while pondering the probabilities of winning acard game of solitaire.

Monte Carlo -- definition, examples

- Given a very large set X and a distribution p(x) over it
- We draw i.i.d. (independent and identically distributed) a set of N samples
- We can then approximate the distribution using these samples



Monte Carlo Principle

$$E_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \underset{N\to\infty}{\rightarrow} E(f) = \sum_x f(x)\,\mathrm{p}(x)$$

What do we need next ?

Sampling Methods

- *Rejection*
- *Metropolis*
- *Metropolis-Hasting*
- *Exact Sampling*

a

j

X

Metrop

7.1 METROPOLIS–HASTIN

A very general method for constructi
algorithm [324, 460]. The method be
drawn at random from some starti
$f\left(\mathbf{x}^{(0)}\right) > 0$. Given $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$, the al

1. Sample a candidate value $\mathbf{X}^*$ fr
2. Compute the *Metropolis–Hasti*

$R(\mathbf{u},$

Note that $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right)$ is alway
only occur if $f\left(\mathbf{x}^{(t)}\right) > 0$ and

Metropolis-Hasting

7.1 METROPOLIS–HASTINGS ALGORITHM

A very general method for constructing a Markov chain is the Metropolis–Hastings algorithm [324, 460]. The method begins at $t = 0$ with the selection of $\mathbf{X}^{(0)} = \mathbf{x}^{(0)}$ drawn at random from some starting distribution g, with the requirement that $f\left(\mathbf{x}^{(0)}\right) > 0$. Given $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$, the algorithm generates $\mathbf{X}^{(t+1)}$ as follows:

1. Sample a candidate value $\mathbf{X}^*$ from a *proposal distribution* $g\left(\cdot \mid \mathbf{x}^{(t)}\right)$.
2. Compute the *Metropolis–Hastings ratio* $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right)$, where

Metropolis-Hasting

7.1 METROPOLIS–HASTINGS ALGORITHM

A very general method for constructing a Markov chain is the Metropolis–Hastings algorithm [324, 460]. The method begins at $t = 0$ with the selection of $\mathbf{X}^{(0)} = \mathbf{x}^{(0)}$ drawn at random from some starting distribution g, with the requirement that $f\left(\mathbf{x}^{(0)}\right) > 0$. Given $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$, the algorithm generates $\mathbf{X}^{(t+1)}$ as follows:

1. Sample a candidate value $\mathbf{X}^*$ from a *proposal distribution* $g\left(\cdot|\,\mathbf{x}^{(t)}\right)$.
2. Compute the *Metropolis–Hastings ratio* $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right)$, where

$$R(\mathbf{u}, \mathbf{v}) = \frac{f(\mathbf{v})\, g(\mathbf{u}|\,\mathbf{v})}{f(\mathbf{u})\, g(\mathbf{v}|\,\mathbf{u})}. \tag{7.1}$$

   Note that $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right)$ is always defined, because the proposal $\mathbf{X}^* = \mathbf{x}^*$ can only occur if $f\left(\mathbf{x}^{(t)}\right) > 0$ and $g\left(\mathbf{x}^*|\,\mathbf{x}^{(t)}\right) > 0$.

3. Sample a value for $\mathbf{X}^{(t+1)}$ according to the following:

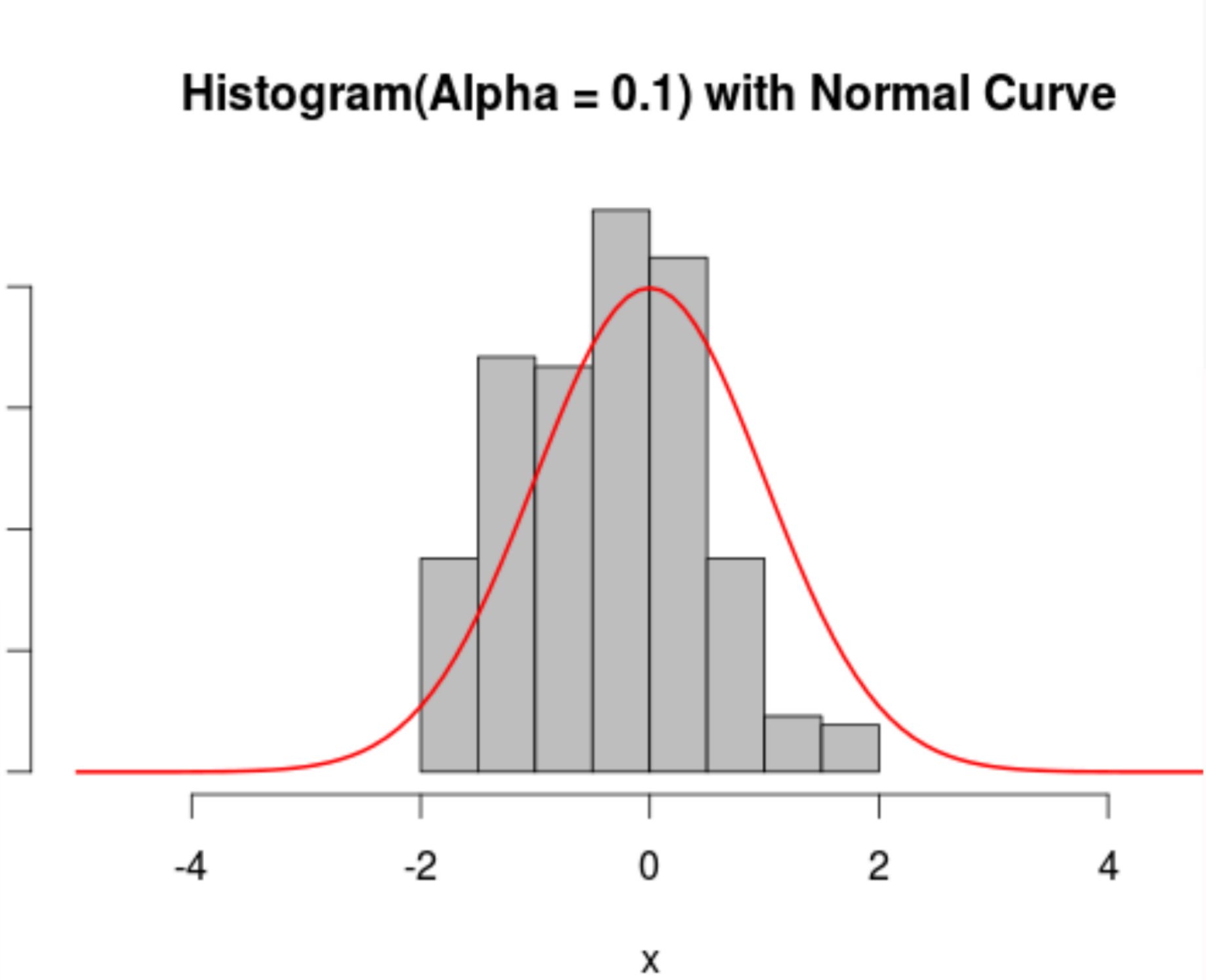
$$\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{X}^* & \text{with probability } \min\left\{R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right), 1\right\}, \\ \mathbf{x}^{(t)} & \text{otherwise.} \end{cases} \tag{7.2}$$

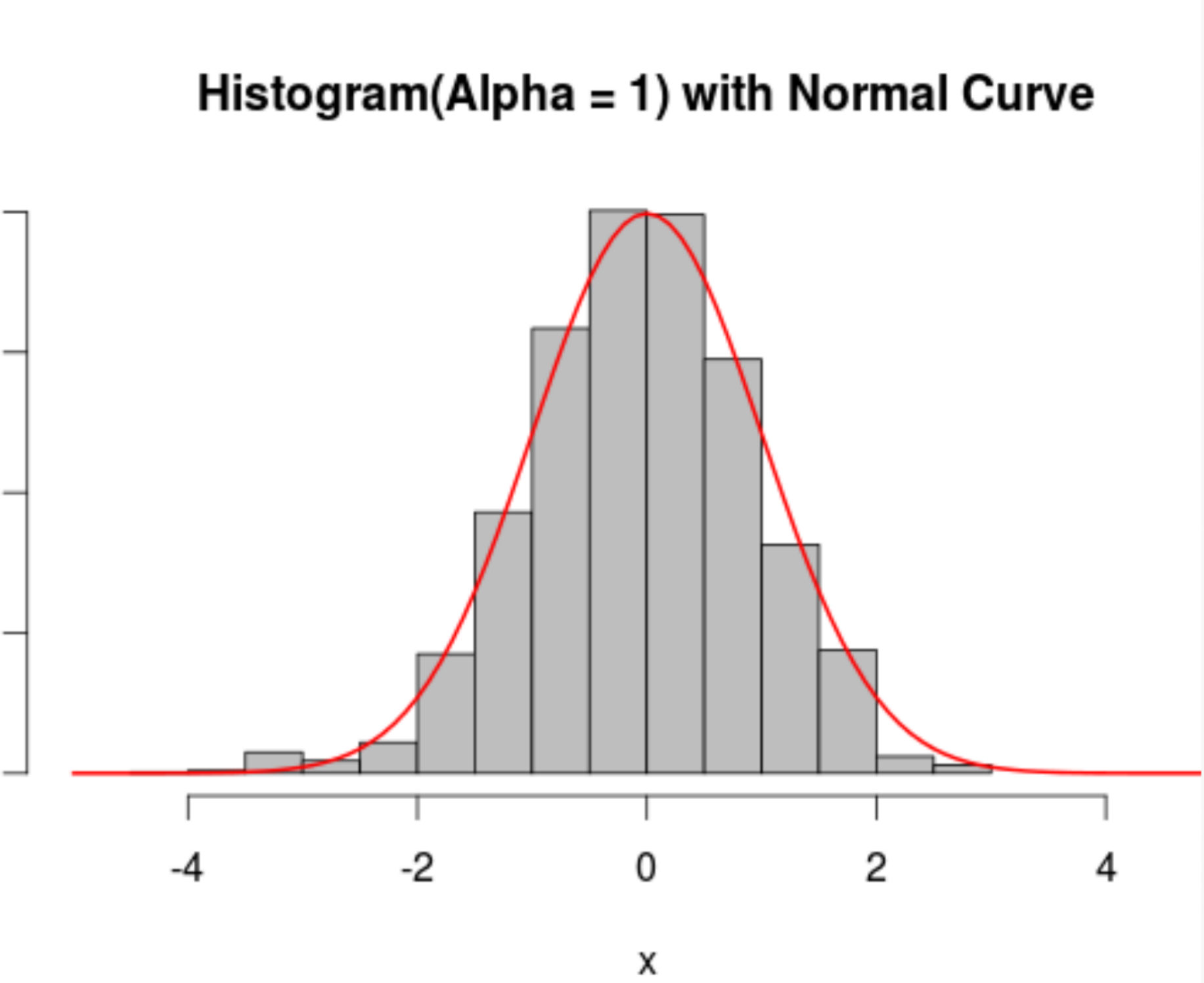
4. Increment t and return to step 1.

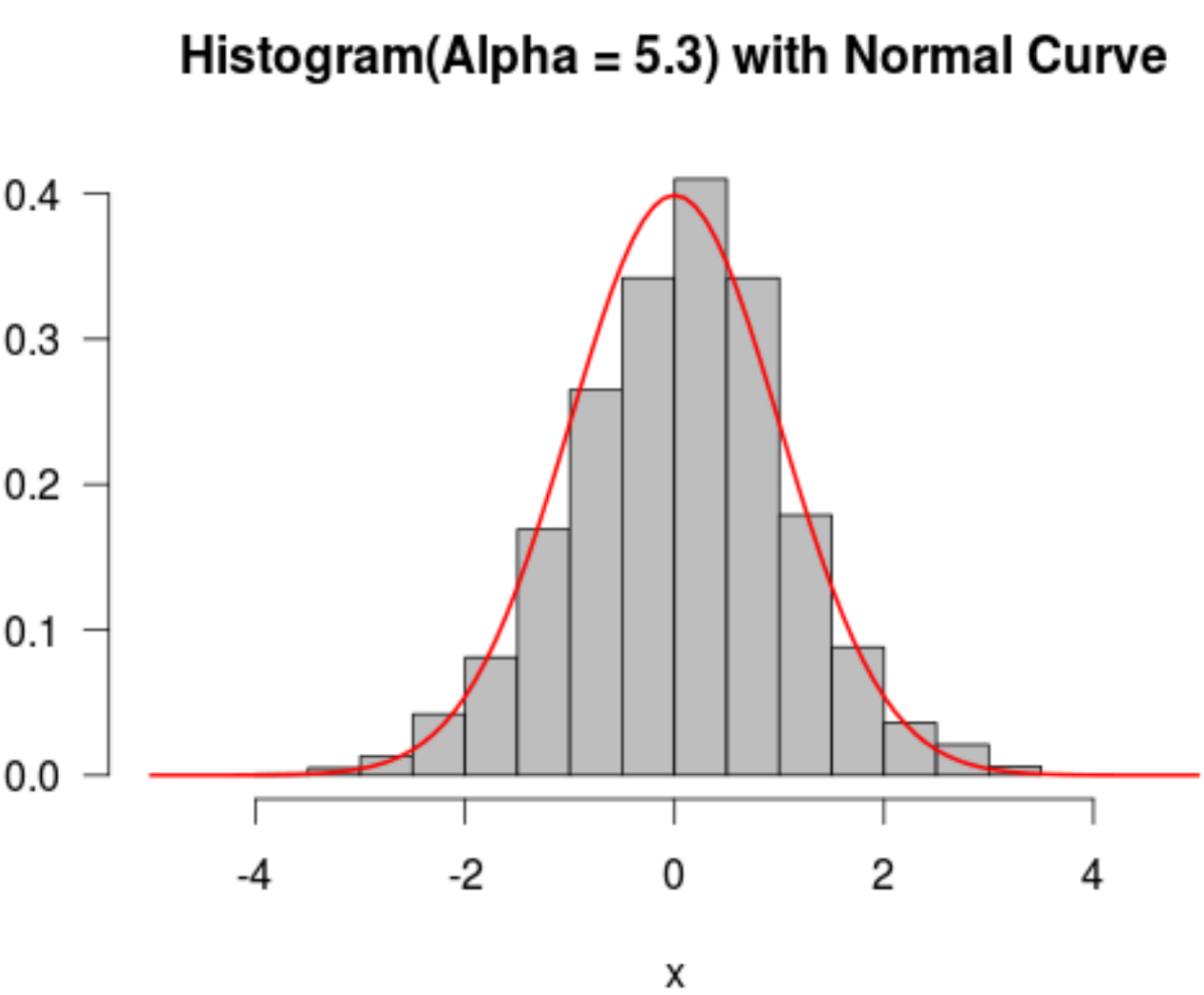
We will call the tth iteration the process that generates $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$. When the proposal distribution is symmetric so that $g\left(\mathbf{x}^{(t)}|\mathbf{x}^*\right) = g\left(\mathbf{x}^*|\mathbf{x}^{(t)}\right)$, the method is known as the Metropolis algorithm [460].

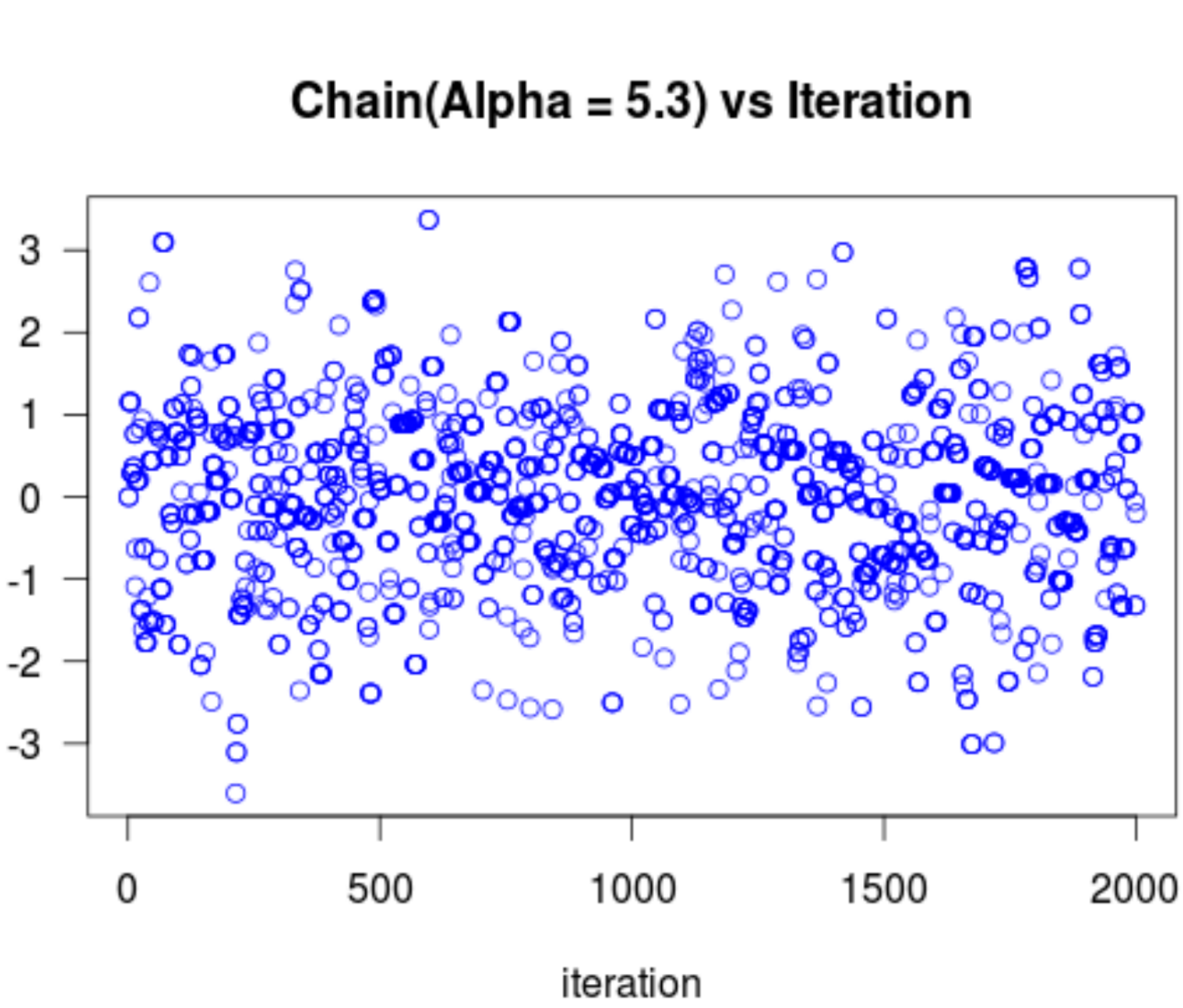
Metropolis algorithm

The function normm is simulating from a normal with zero mean and unit variance using a Metropolis algorithm with uniform proposal distribution.








Chain(Alpha = 5.3) vs Iteration
3
2
1
0
-1
-2
-3
0
500
1000
1500
2000
iteration

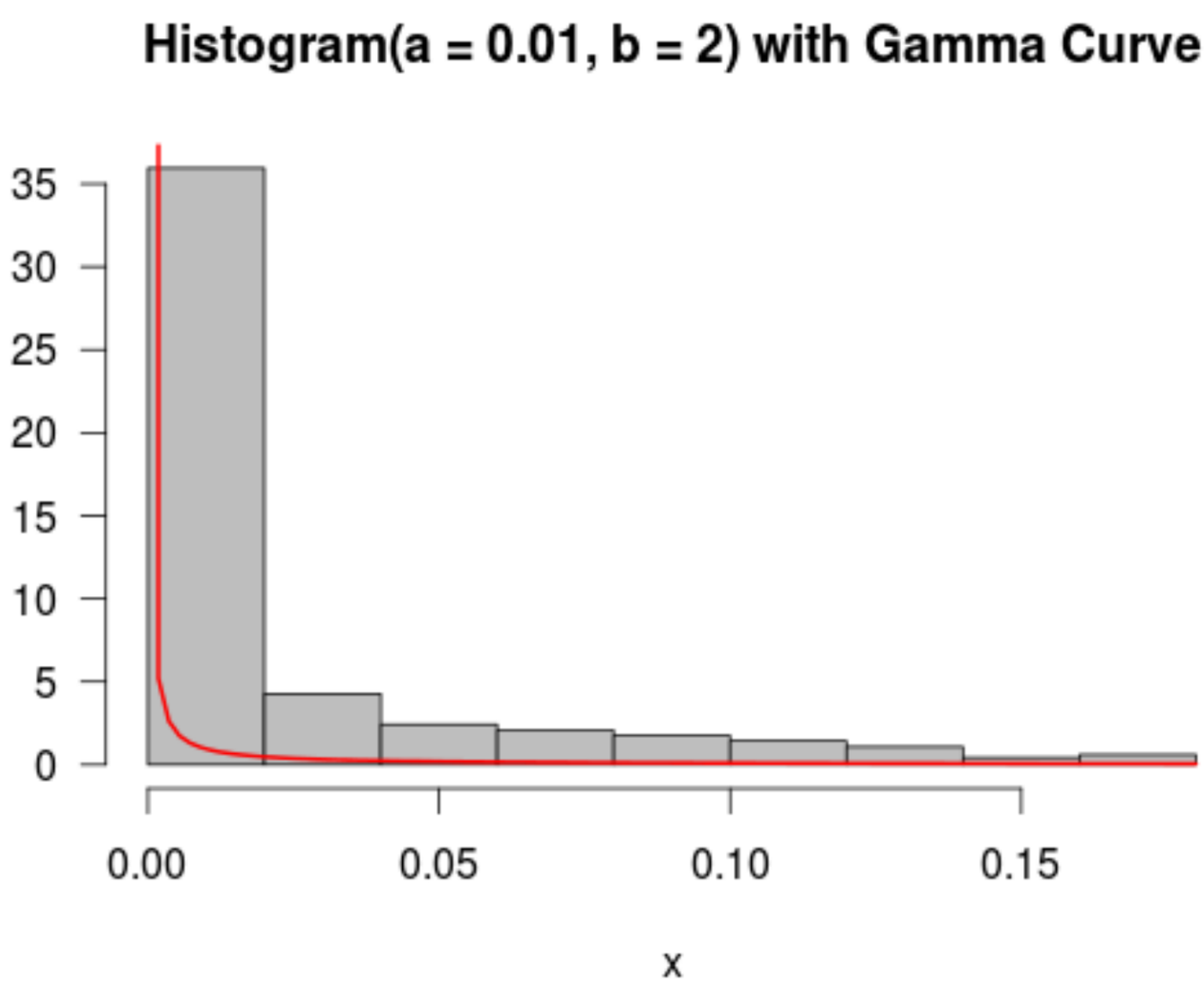
Suppose that the proposal distribution for the Metropolis–Hastings algorithm is chosen such that $g\left(\mathbf{x}^* | \mathbf{x}^{(t)}\right) = g(\mathbf{x}^*)$ for some fixed density g. This yields an independence chain, where each candidate value is drawn independently of the past. In this case, the Metropolis–Hastings ratio is

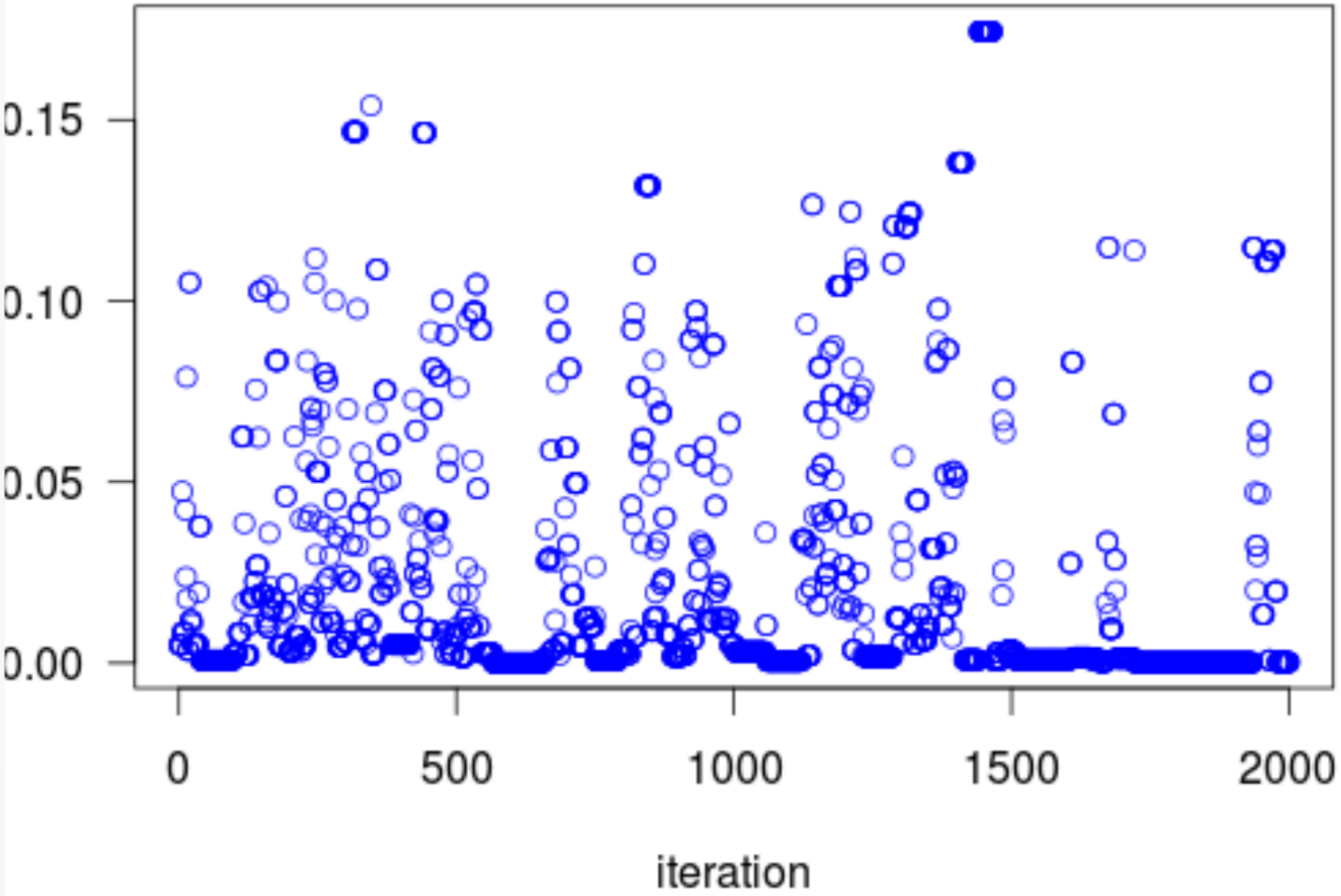
$$R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right) = \frac{f\left(\mathbf{X}^*\right) g\left(\mathbf{x}^{(t)}\right)}{f\left(\mathbf{x}^{(t)}\right) g\left(\mathbf{X}^*\right)}. \tag{7.4}$$

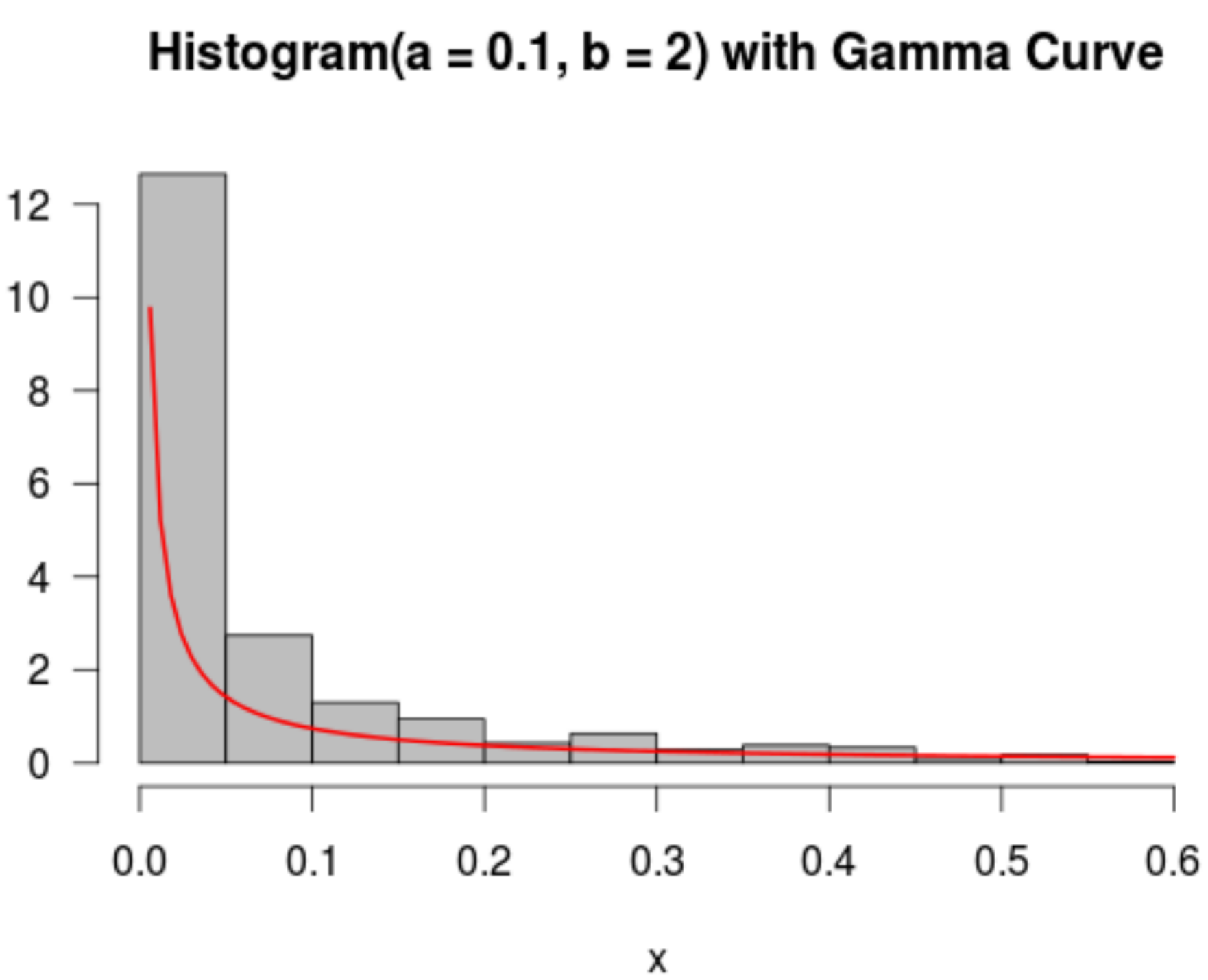
The resulting Markov chain is irreducible and aperiodic if $g(\mathbf{x}) > 0$ whenever $f(\mathbf{x}) > 0$.

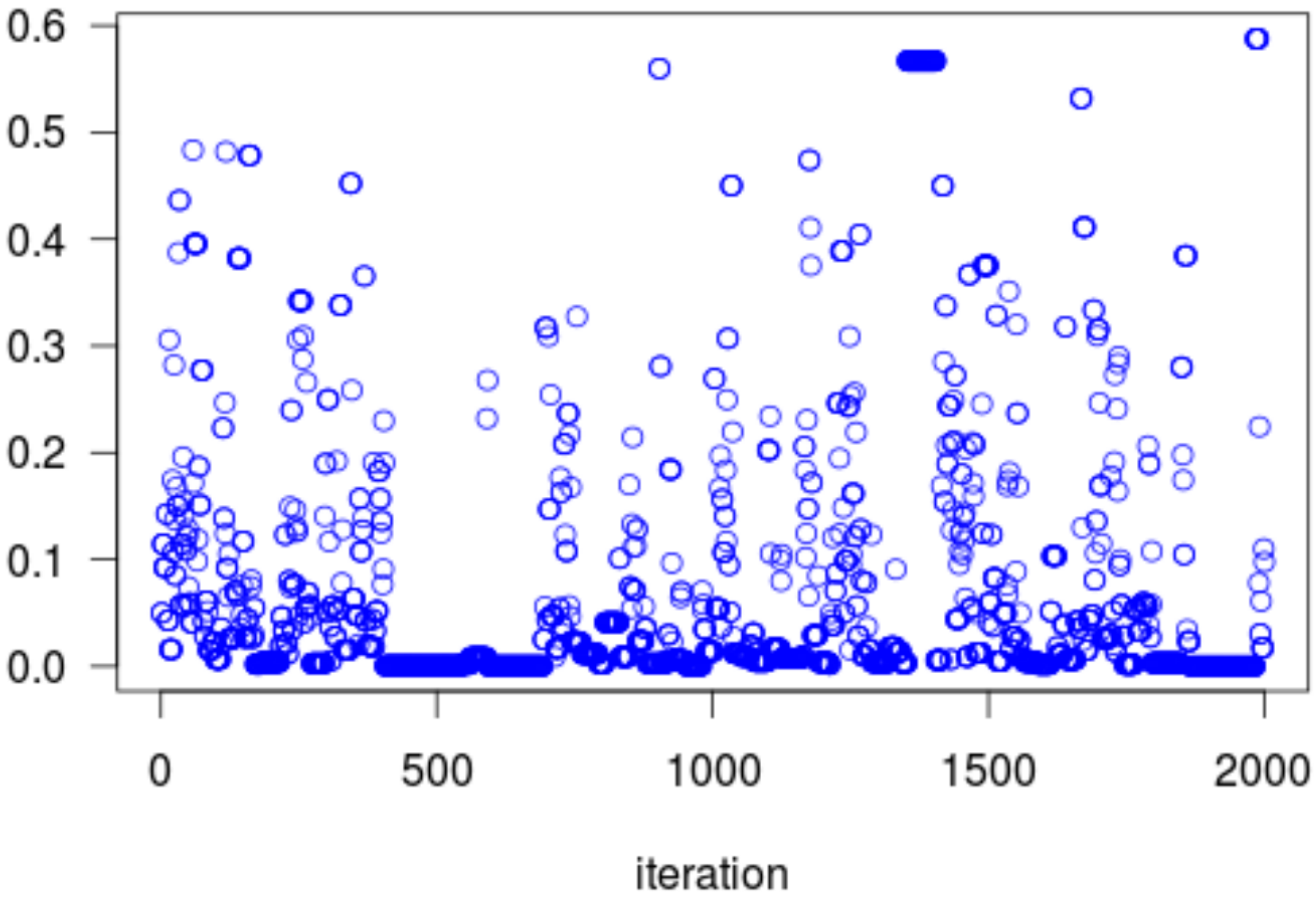
Notice that the Metropolis–Hastings ratio in (7.4) can be reexpressed as the ratio of importance ratios (see Section 6.4.1) where f is the target and g is the envelope: If $w^* = f\left(\mathbf{X}^*\right) / g\left(\mathbf{X}^*\right)$ and $w^{(t)} = f\left(\mathbf{x}^{(t)}\right) / g\left(\mathbf{x}^{(t)}\right)$, then $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right) = w^* / w^{(t)}$. This reexpression indicates that when $w^{(t)}$ is much larger than typical w^* values, then the chain will tend to get stuck for long periods at the current value. Therefore, the criteria discussed in Section 6.3.1 for choosing importance sampling envelopes are also relevant here for choosing proposal distributions: The proposal distribution g should resemble the target distribution f, but should cover f in the tails.

Example 7.1 (Bayesian Inference) MCMC methods like the Metropolis–Hastings algorithm are particularly popular tools for Bayesian inference, where some data $\mathbf{y}$ are observed with likelihood function $L(\boldsymbol{\theta}|\mathbf{y})$ for parameters $\boldsymbol{\theta}$ which have prior distribution $p(\boldsymbol{\theta})$. Bayesian inference is based on the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}) =$


Histogram(a = 0.01, b = 2) with Gamma Curve
35
30
25
20
15
10
5
0
0.00
0.05
0.10
0.15
x


Chain(a = 0.01, b = 2) vs Iteration
0.15
0.10
0.05
0.00
0
500
1000
1500
2000
iteration


Histogram(a = 0.1, b = 2) with Gamma Curve
12
10
8
6
4
2
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
x


Chain(a = 0.1, b = 2) vs Iteration
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
500
1000
1500
2000
iteration

MCMC

A quick review of discrete-state-space Markov chain theory is provided in Section 1.7. Let the sequence $\{\mathbf{X}^{(t)}\}$ denote a Markov chain for $t = 0, 1, 2, \ldots$, where $\mathbf{X}^{(t)} = \left(X_1^{(t)}, \ldots, X_p^{(t)}\right)$ and the state space is either continuous or discrete. For the types of Markov chains introduced in this chapter, the distribution of $\mathbf{X}^{(t)}$ converges to the limiting stationary distribution of the chain when the chain is irreducible and aperiodic. The MCMC sampling strategy is to construct an irreducible, aperiodic Markov chain for which the stationary distribution equals the target distribution f. For sufficiently large t, a realization $\mathbf{X}^{(t)}$ from this chain will have approximate marginal distribution f. A very popular application of MCMC methods is to facilitate Bayesian inference where f is a Bayesian posterior distribution for parameters $\mathbf{X}$; a short review of Bayesian inference is given in Section 1.5.

The art of MCMC lies in the construction of a suitable chain. A wide variety of algorithms has been proposed. The dilemma lies in how to determine the degree of distributional approximation that is inherent in realizations from the chain as well as estimators derived from these realizations. This question arises because the distribution of $\mathbf{X}^{(t)}$ may differ substantially from f when t is too small (note that t is always limited in computer simulations), and because the $\mathbf{X}^{(t)}$ are serially dependent.

Gibbs sampling

Let $X = (X_1, \ldots, X_d)$ be a random vector in $\mathbb{R}^d$. Define the $d-1$ dimensional random vectors

$$X_{(-j)} = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_d),$$

and denote the corresponding univariate conditional density of X_j given $X_{(-j)}$ by $f(X_j|X_{(-j)})$. The Gibbs sampler generates the chain by sampling from each of the d conditional densities $f(X_j|X_{(-j)})$.

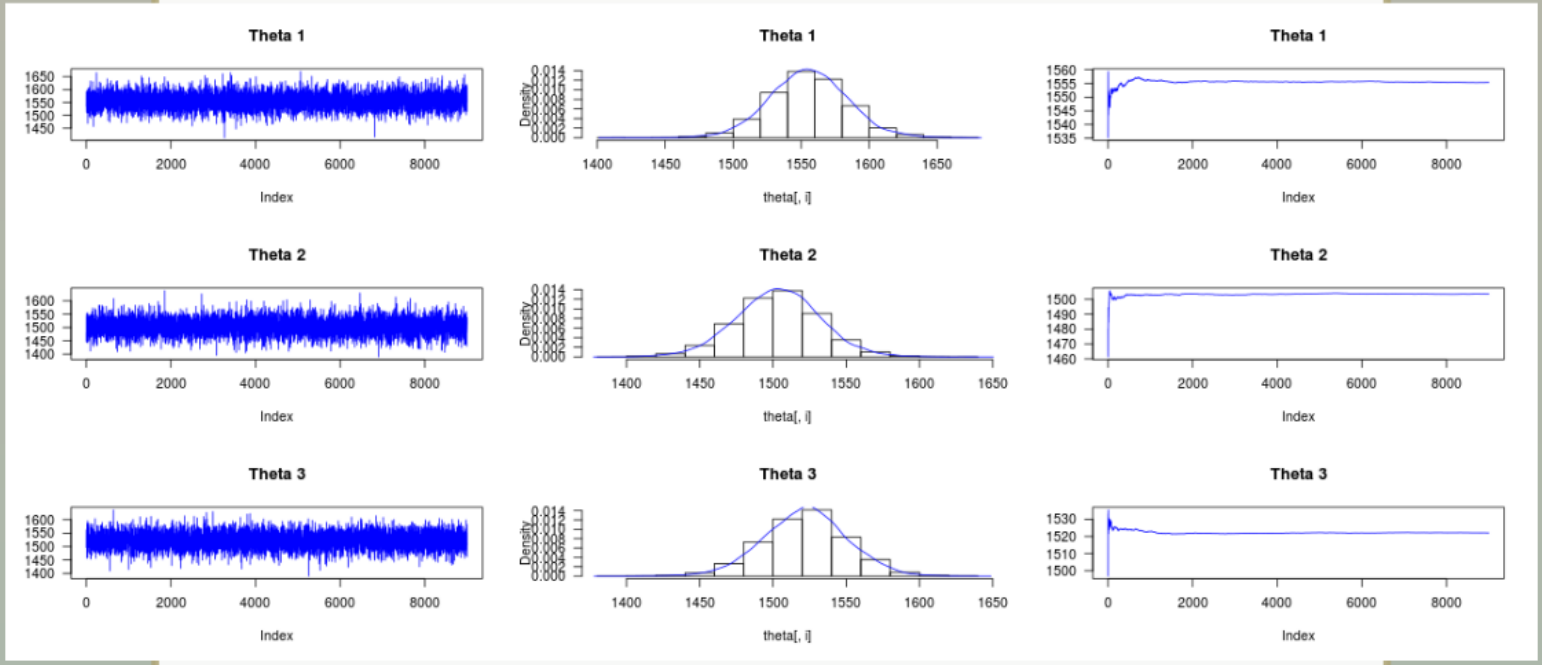
In the following algorithm for the Gibbs sampler, we denote X_t by $X(t)$.

1. Initialize $X(0)$ at time $t = 0$.
2. For each iteration, indexed $t = 1, 2, \ldots$ repeat:
   (a) Set $x_1 = X_1(t-1)$.
   (b) For each coordinate $j = 1, \ldots, d$
      (a) Generate $X_j^*(t)$ from $f(X_j|x_{(-j)})$.
      (b) Update $x_j = X_j^*(t)$.
   (c) Set $X(t) = (X_1^*(t), \ldots, X_d^*(t))$ (every candidate is accepted).
   (d) Increment t.

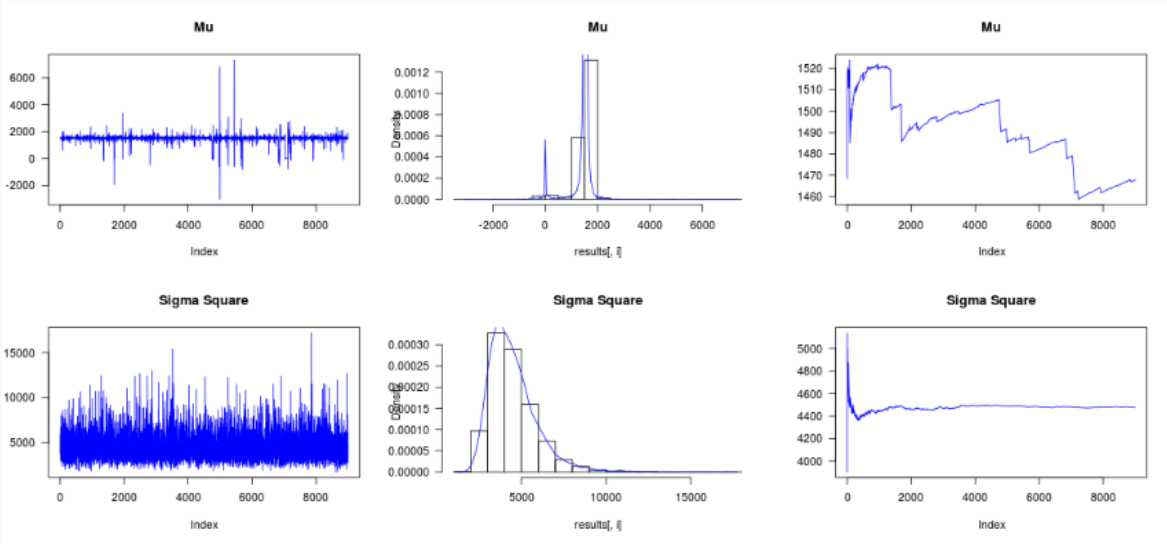
Result -1

	Mean	Sd	lower	upper
θ_1	1555.37	28.48	1499.30	1611.80
θ_2	1503.50	28.79	1447.39	1561.48
θ_3	1522.08	28.31	1465.80	1577.17
θ_4	1509.73	28.85	1453.16	1566.20
θ_5	1522.91	28.40	1469.31	1580.90
θ_6	1550.55	28.33	1496.18	1607.05
μ	1467.87	331.79	941.01	1909.73
σ^2	4478.23	1397.89	2218.08	7178.89
τ_2	840686.15	5275551.11	218.41	1466706.88
σ_μ^2	10466580702.17	435632416730.93	0.00	614466261.07

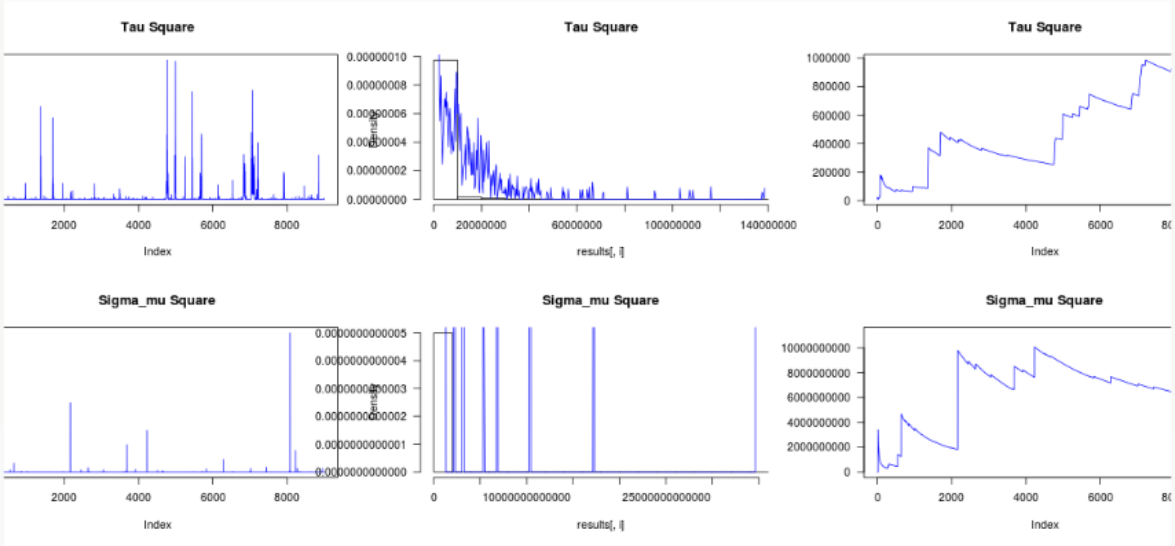
Result -2



Result -3


Mu
Index
Mu
Density
results[, i]
Mu
Index
Sigma Square
Index
Sigma Square
Density
results[, i]
Sigma Square
Index

Result -4


Tau Square
Index
Tau Square
results[, i]
Tau Square
Index
Sigma_mu Square
Index
Sigma_mu Square
results[, i]
Sigma_mu Square
Index

Result -5

	Mean	SD	Naive SE	Time-series SE
beta.(Intercept)	0.007846	1.005	0.01005	0.01005
b.(Intercept).1	1504.737280	23.164	0.23164	0.22812
b.(Intercept).2	1527.600689	23.048	0.23048	0.22637
b.(Intercept).3	1563.436784	23.193	0.23193	0.23193
b.(Intercept).4	1497.542896	23.155	0.23155	0.23155
b.(Intercept).5	1599.416036	23.071	0.23071	0.22578
b.(Intercept).6	1469.639794	23.272	0.23272	0.23272
VCV.(Intercept).(Intercept)	2768403.634708	2181862.591	21818.62591	21818.62591
sigma2	2681.544750	850.396	8.50396	10.49865
Deviance	320.546141	4.301	0.04301	0.05413

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
beta.(Intercept)	-1.95	-0.6656	0.008974	0.6796	2.011
b.(Intercept).1	1458.80	1489.3055	1504.789726	1520.0128	1549.648
b.(Intercept).2	1481.68	1512.7453	1527.789610	1542.6936	1573.812
b.(Intercept).3	1517.57	1547.9565	1563.545310	1578.6787	1609.432
b.(Intercept).4	1451.24	1482.2187	1497.456942	1512.4841	1543.284
b.(Intercept).5	1554.21	1584.2422	1599.307265	1614.7616	1644.876
b.(Intercept).6	1422.77	1454.2027	1469.993666	1485.0459	1515.979

Q and A

Tack

MCMC

Yonglin Zhuo