An Introduction to Bayesian Inference and MCMC

Dr David A. Stephens

Department of Mathematics & Statistics
Room 1225, Burnside Hall

david.stephens@mcgill.ca

www.math.mcgill.ca/dstephens/

Part I

Bayesian Inference

Statistical Inference and Testing

Standard statistical analysis procedures

- summary
- estimation
- hypothesis testing
- uncertainty representation (i.e. confidence intervals)
- prediction

aim to turn *data* into *decisions*.

Data

The observed data we collect provide only a 'noise corrupted' version of the underlying truth:

- in a finite data set, we have *incomplete* information;
- we need to represent this lack of complete knowledge using uncertainty reasoning;
- we use *probability theory* to do this.

Data

Typically we assume that we have access to a (finite) *random sample* of data: for $i = 1, \ldots, n$,

- single variables x_i
- p-dimensional vectors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$
- predictor/response pairs $(\mathbf{x}_i, y_i)$

believed to be drawn from some probability distribution, F (or density f).

We aim to learn about F, in order to make decisions.

Inference

The term *inference* is usually used to mean learning about F:

- typically we assume a *parametric* model using parameter θ and write the pdf and cdf as

$$f(x;\theta) \qquad F(x;\theta)$$

- θ is a d-dimensional vector parameter that determines the 'shape' of the distribution.
  - Binomial(m,p): $\theta = p$;
  - Poisson(λ): $\theta = \lambda$;
  - Normal(μ,σ^2): $\theta = (\mu,\sigma^2)^\top$;

Likelihood

The most common form of inference is based on the *likelihood function* which for a random sample takes the form

$$\mathcal{L}_n(\theta) = \prod_{i=1}^{n} f(x_i; \theta)$$

that is, the product of density values evaluated at the data $x_1, \ldots, x_n$ but viewed as a function of θ. The *log likelihood* is then

$$\ell_n(\theta) = \sum_{i=1}^{n} \log f(x_i; \theta).$$

We then typically estimate θ by maximizing these functions to yield the *maximum likelihood estimate* (mle)

$$\widehat{\theta} = \operatorname*{argmax}_{\theta} \mathcal{L}_n(\theta) = \operatorname*{argmax}_{\theta} \ell_n(\theta)$$

Likelihood (contd.)

Example: Binomial case

If $X_i \sim \text{Binomial}(m, \theta)$, $i = 1, \ldots, n$, then

$$\mathcal{L}_n(\theta) = \prod_{i=1}^{n} \binom{m}{x_i} \theta^{x_i} (1-\theta)^{m-x_i} \propto \theta^{S_n} (1-\theta)^{nm-S_n}$$

where $S_n = \sum_{i=1}^{n} x_i$. Then

$$\widehat{\theta} = \frac{S_n}{nm}.$$

Likelihood (contd.)

Example: Binomial case

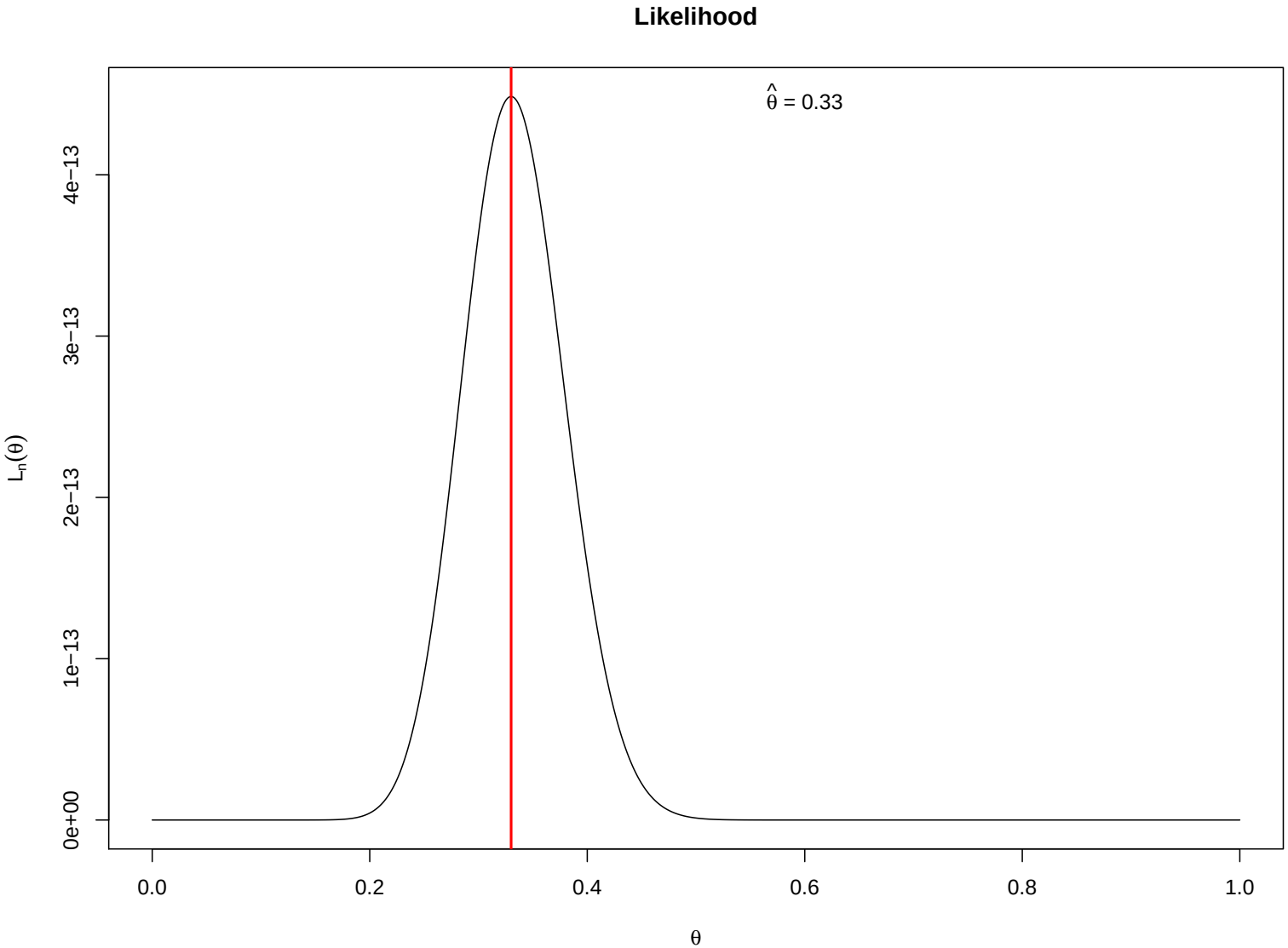
In the following numerical example, $\theta = 0.3$, with

$$m = 5 \qquad n = 20$$

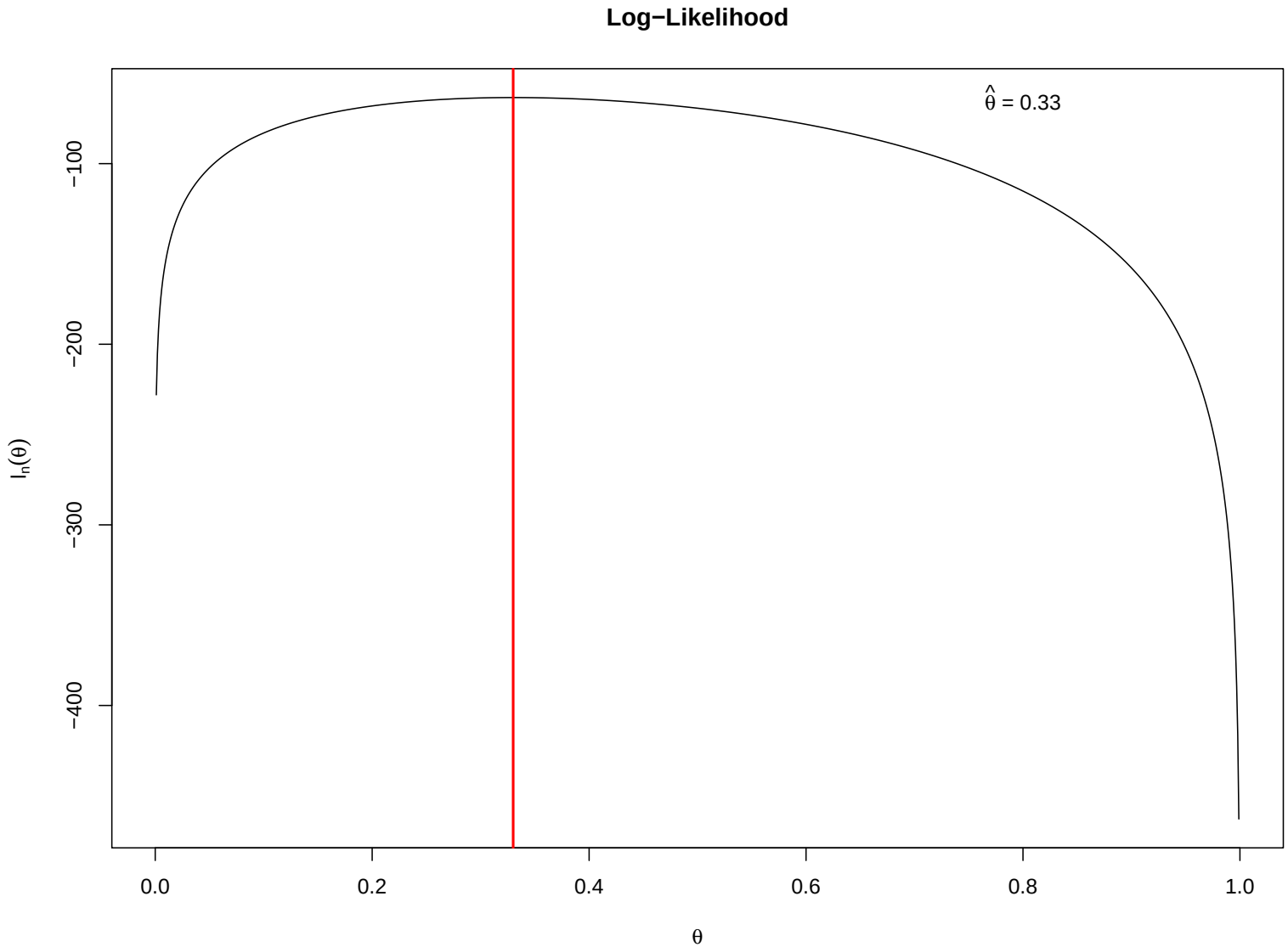
For the simulated data

$$S_n = 33 \qquad \therefore \qquad \widehat{\theta} = 0.33.$$

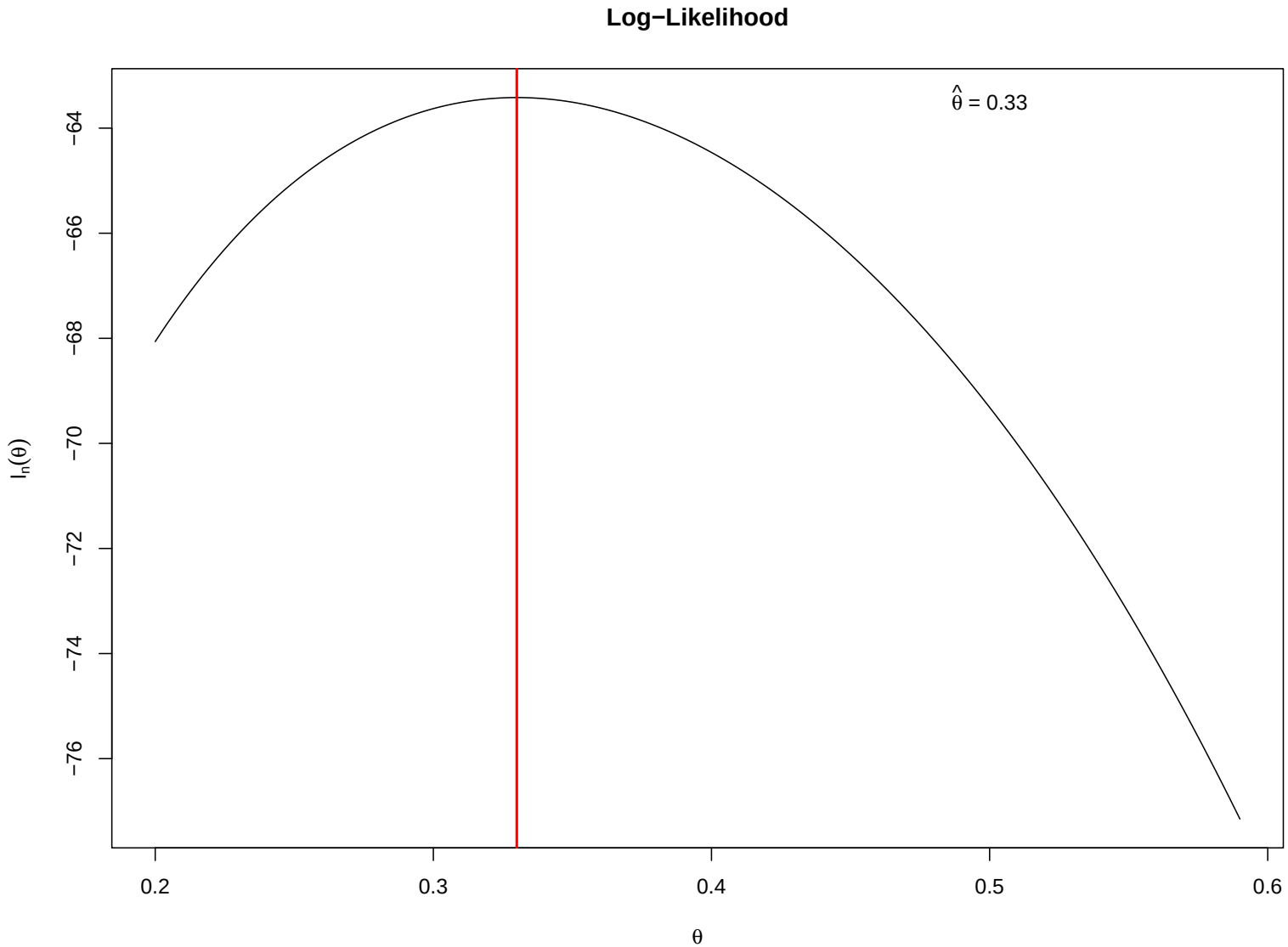
Likelihood (contd.)



Log Likelihood



Log Likelihood



Log Likelihood

We typically approximate the curvature of the log-likelihood at the mle by using its second derivative, as

$$\ell_n(\theta) \doteq \ell_n(\widehat{\theta}) + \frac{1}{2}\ddot{\ell}_n(\widehat{\theta})(\theta - \widehat{\theta})^2.$$

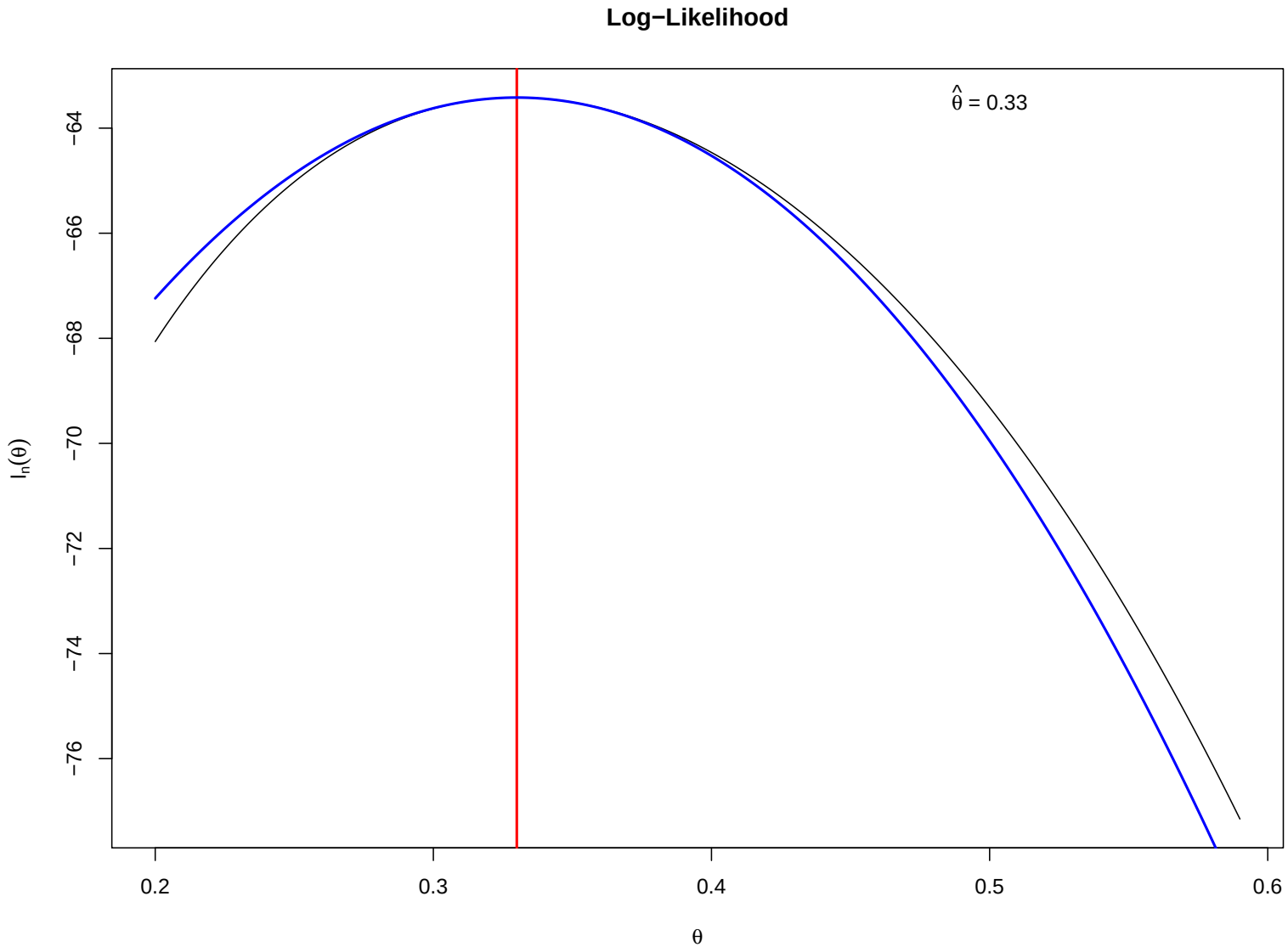
We have

$$\dot{\ell}_n(\theta) = \frac{d\ell_n(\theta)}{d\theta} = \frac{S_n}{\theta} - \frac{(nm - S_n)}{(1-\theta)}$$

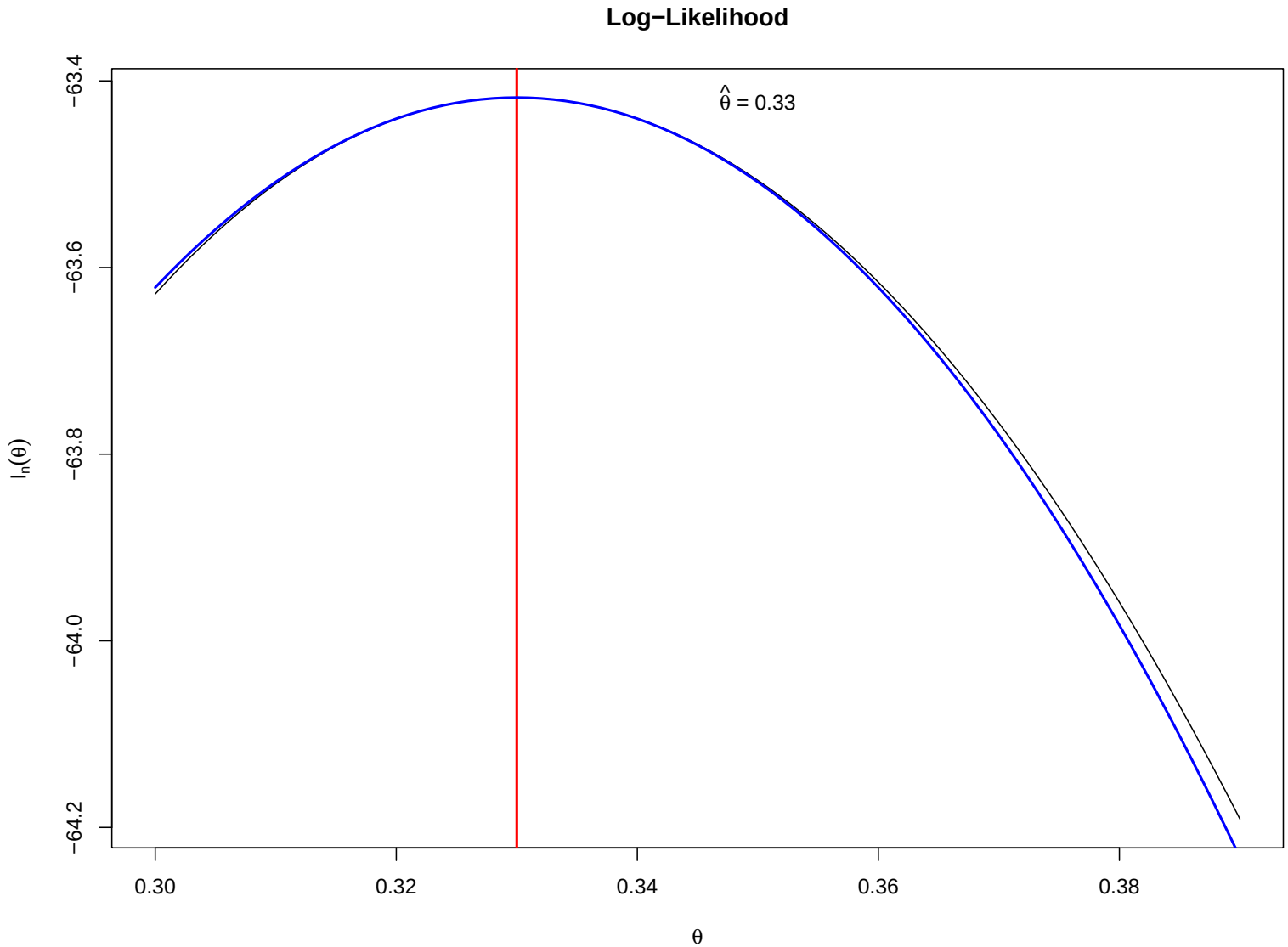
$$\ddot{\ell}_n(\theta) = \frac{d^2\ell_n(\theta)}{d\theta^2} = -\frac{S_n}{\theta^2} - \frac{(nm - S_n)}{(1-\theta)^2}$$

which we evaluate at $\theta = \widehat{\theta}$.

Log Likelihood & Quadratic Approximation



Log Likelihood & Quadratic Approximation



Log Likelihood & Quadratic Approximation

We may deduce from this that, at least for θ near to $\widehat{\theta}$, we have

$$\mathcal{L}_n(\theta) \doteq \mathcal{L}_n(\widehat{\theta}) \exp\left\{ -\frac{\ddot{\ell}_n(\widehat{\theta})}{2}(\theta - \widehat{\theta})^2 \right\}$$

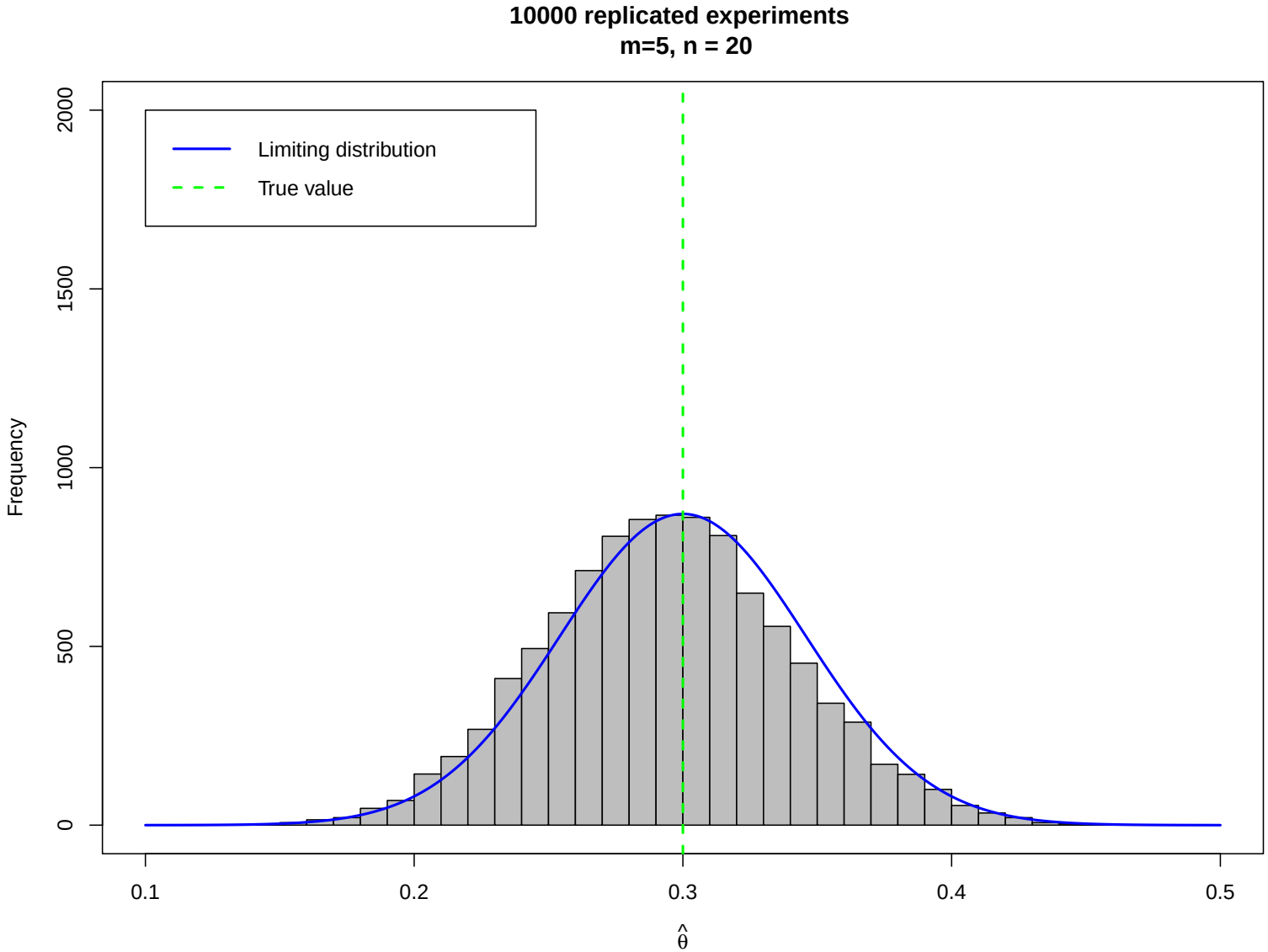
and we may state that for large n the *sampling distribution* of the ML estimator is approximately Normal

$$\widehat{\theta} \; \dot{\sim} \; \text{Normal}(\theta, \{-\ddot{\ell}_n(\theta)\}^{-1})$$

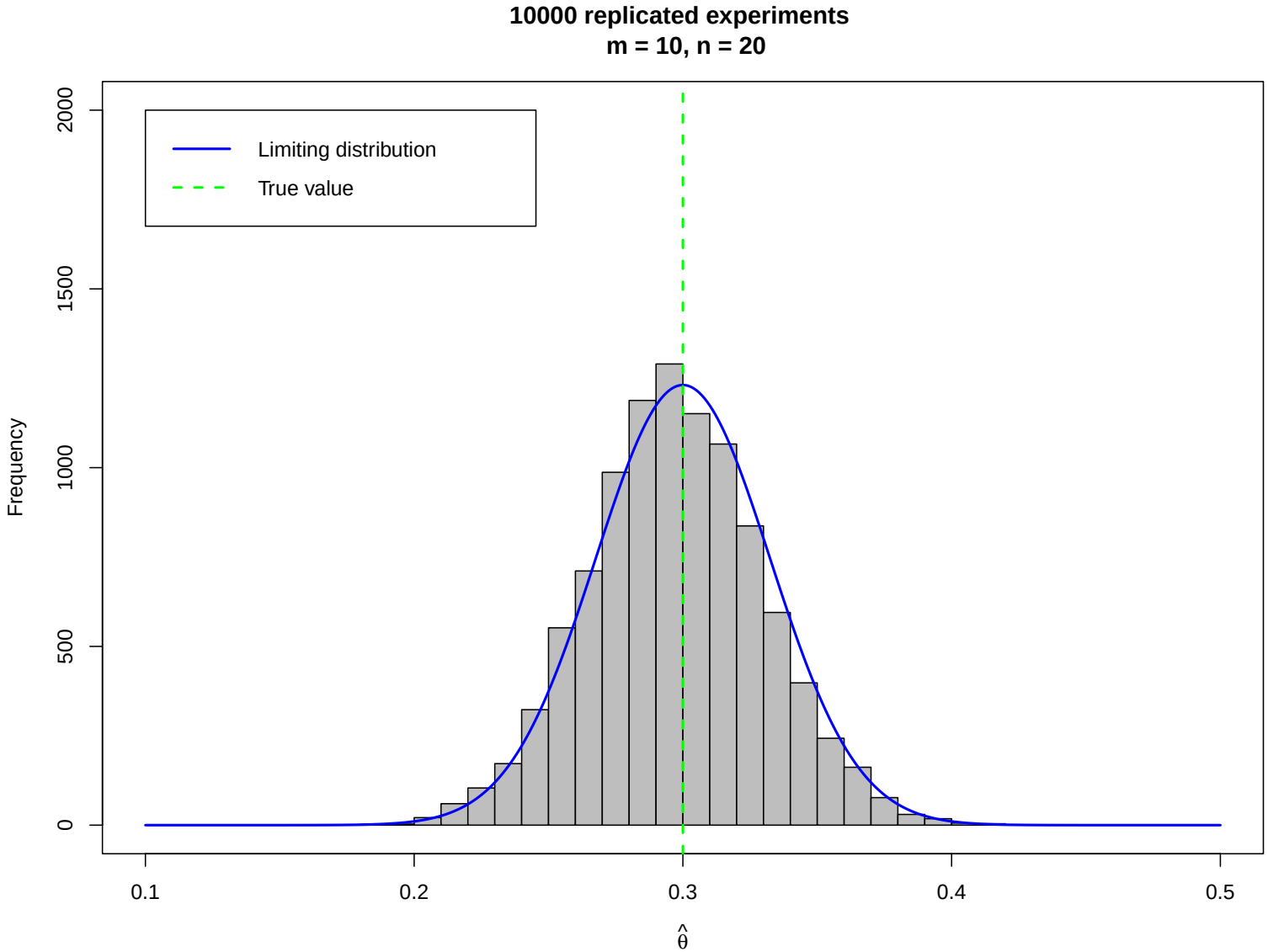
Hence we may construct a (large sample) 95 % confidence interval in the form

$$\widehat{\theta} \pm 1.96\{-\ddot{\ell}_n(\widehat{\theta})\}^{-1/2}$$

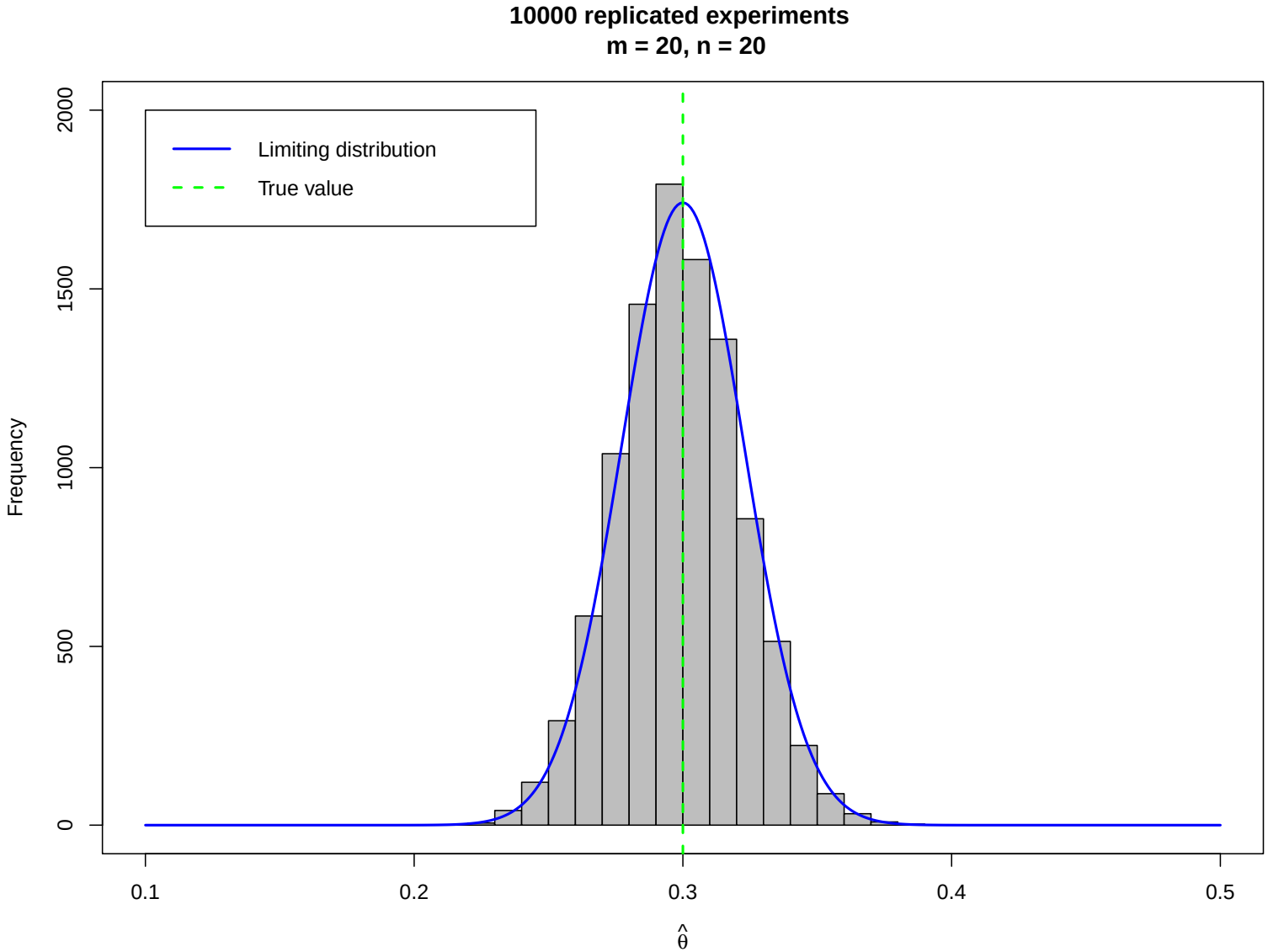
Log Likelihood & Quadratic Approximation



Log Likelihood & Quadratic Approximation



Log Likelihood & Quadratic Approximation



This is Fake Inference !

If $\mathbf{x} = (x_1, \ldots, x_n)$, and $c_L(\mathbf{x})$ and $c_U(\mathbf{x})$ denote the computed lower and upper limits of the confidence interval, then the confidence interval statement is based on the calculation

$$\Pr_\theta[c_L(\mathbf{X}) \leqslant \theta \leqslant c_U(\mathbf{X})] \doteq 0.95$$

that is, **if the true value is** θ, the probability that θ lies between $c_L(\mathbf{X})$ and $c_U(\mathbf{X})$ (two random quantities, defined via the data $\mathbf{X} = (X_1, \ldots, X_n)$, is approximately 0.95.

It is **not** stating that with probability approximately 0.95, θ lies between $c_L(\mathbf{x})$ and $c_U(\mathbf{x})$.

The logical calculation ?

In light of the data we have **observed**, we wish to make probability statements about the quantity that we have **not observed**.

Therefore we are using a *conditional probability* argument:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

where A is the unobserved quantity, and B is the observed quantity.

Bayesian Inference

In Bayesian analysis, θ is treated as a random variable, as its value is *unknown* prior to the data being collected. We write

$$f_{X|\theta}(x|\theta)$$

for the conditional density. A *prior density* then encapsulates the beliefs about the random variable θ *before the data* are collected.

Denoting by $\pi_0(\theta)$ the prior density, by Bayes Theorem, we have that in light of the data,

$$\pi_n(\theta) = \pi_{\theta|\mathbf{X}}(\theta|\mathbf{x}) = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}|\theta)\pi_0(\theta)}{f_{\mathbf{X}}(\mathbf{x})} = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}|\theta)\pi_0(\theta)}{\int f_{\mathbf{X}|\theta}(\mathbf{x}|t)\pi_0(t)dt} \quad (1)$$

Bayesian Inference (contd.)

- The denominator in (1), $f_{\mathbf{X}}(\mathbf{x})$ is sometimes termed the *marginal likelihood* of the data $\mathbf{x}$.
- The term $\pi_{\theta|\mathbf{X}}(\theta|\mathbf{x})$ in (1) is the *posterior distribution* of θ given $\mathbf{x}$, which encapsulates the beliefs about θ in light of the data.
- Note that for the posterior calculation in (1)

$$\pi_n(\theta) \propto f_{\mathbf{X}|\theta}(\mathbf{x}|\theta)\pi_0(\theta)$$

ignoring terms that do not involve θ; $f_{\mathbf{X}}(\mathbf{x})$ for fixed $\mathbf{x}$ acts as a normalizing constant.

Bayesian Inference (contd.)

The Bayesian calculation in (1) acts sequentially: for data $\mathbf{x}_1$

$$\pi_{\theta|\mathbf{x}_1}(\theta|\mathbf{x}_1) = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}_1|\theta)\pi_0(\theta)}{f_{\mathbf{X}}(\mathbf{x}_1)} = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}_1|\theta)\pi_0(\theta)}{\int f_{\mathbf{X}|\theta}(\mathbf{x}_1|t)\pi_0(t)\,dt} \tag{2}$$

If new (independent and identically distributed to $\mathbf{x}_1$) data $\mathbf{x}_2$ become available, then the posterior for θ in light of the combined data $(\mathbf{x}_1, \mathbf{x}_2)$ is

$$\pi_{\theta|\mathbf{x}_1,\mathbf{x}_2}(\theta|\mathbf{x}_1,\mathbf{x}_2) = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}_1,,\mathbf{x}_2|\theta)\pi_0(\theta)}{f_{\mathbf{X}}(\mathbf{x}_1,\mathbf{x}_2)} = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}_1,\mathbf{x}_2|\theta)\pi_0(\theta)}{\int f_{\mathbf{X}|\theta}(\mathbf{x}_1,\mathbf{x}_2|t)\pi_0(t)\,dt}.$$

Bayesian Inference (contd.)

But note also that

$$\pi_{\theta|\mathbf{x}_1,\mathbf{x}_2}(\theta|\mathbf{x}_1,\mathbf{x}_2) = \frac{f_{\mathbf{X}|\theta}(\mathbf{x}_2|\theta)\pi_{\theta|\mathbf{x}_1}(\theta|\mathbf{x}_1)}{f_{\mathbf{X}}(\mathbf{x}_2|\mathbf{x}_1)}$$

where $\pi_{\theta|\mathbf{x}_1}(\theta|\mathbf{x}_1)$ is the posterior for θ from (2), and

$$f_{\mathbf{X}}(\mathbf{x}_2|\mathbf{x}_1) = \frac{f_{\mathbf{X}}(\mathbf{x}_1,\mathbf{x}_2)}{f_{\mathbf{X}}(\mathbf{x}_1)} = \frac{\int f_{\mathbf{X}|\theta}(\mathbf{x}_1|\theta)\pi_0(\theta)\,d\theta}{\int f_{\mathbf{X}|\theta}(\mathbf{x}_1,\mathbf{x}_2|t)\pi_0(t)\,dt} \qquad (3)$$

Construction of Prior Distributions

In the Bayesian formulation, the prior density plays an important role. There are several methods via which the prior can be specified quantitatively;

- from historical or training data;
- by subjective assessment, similar to the subjective assessment of probabilities in elementary probability theory;
- by matching to a desired functional form;
- in a *non-informative* or *vague* specification, where the prior probability is supposedly spread 'evenly' across the parameter space.

Construction of Prior Distributions (contd.)

- *conjugate prior*: a prior that combines with the likelihood to give a tractable posterior.
- *non-informative prior: non-informative prior* attempts to express *prior ignorance* about the parameter of interest.

Construction of Prior Distributions (contd.)

- *Jeffreys' Prior*: Set

$$\pi_0(\theta) \propto \{|\mathcal{I}_\theta(\theta)|\}^{1/2}$$

where $\mathcal{I}_\theta(\theta)$ is the *Fisher Information matrix*,

$$\mathcal{I}_\theta(\theta) = \mathbb{E}_X\left[U(X;\theta)U(X;\theta)^\top;\theta\right] = -\mathbb{E}_X\left[\Psi(X;\theta);\theta\right]$$

and $|\mathcal{I}_\theta(\theta)|$ indicates the absolute value of the determinant of $\mathcal{I}_\theta(\theta)$,

$$U(X;\theta) = \frac{\partial}{\partial\theta}\log f_{X|\theta}(x|\theta)$$

$$\Psi(X;\theta) = \frac{\partial^2}{\partial\theta\partial\theta^\top}\log f_{X|\theta}(x|\theta)$$

Construction of Prior Distributions (contd.)

Binomial(m, θ)

$$\begin{aligned}
\log f_{X|\theta}(x|\theta) &= \log \binom{m}{x} + x \log \theta + (m-x) \log(1-\theta) \\
U(x;\theta) &= \frac{x}{\theta} - \frac{(m-x)}{(1-\theta)} \\
\Psi(x;\theta) &= -\frac{x}{\theta^2} - \frac{(m-x)}{(1-\theta)^2}
\end{aligned}$$

$$\mathcal{I}_\theta(\theta) = -\mathbb{E}_{X|\theta}\left[-\frac{X}{\theta^2} - \frac{(m-X)}{(1-\theta)^2}\right] = \frac{m\theta}{\theta^2} + \frac{m(1-\theta)}{(1-\theta)^2} = \frac{m}{\theta(1-\theta)}$$

Construction of Prior Distributions (contd.)

Binomial(m, θ)

$$\pi_0(\theta) \propto |\mathcal{I}_\theta(\theta)|^{1/2} = \{\theta(1-\theta)\}^{-1/2}$$

Jeffreys' prior is *invariant* under 1-1 transformations, that is, if $\phi = \phi(\theta)$, then the prior for ϕ obtained by reparameterization from θ to ϕ in the prior for θ, is precisely Jeffreys' prior for ϕ.

Construction of Prior Distributions (contd.)

Example: Binomial case

If $X_i \sim \text{Binomial}(m, \theta)$, $i = 1, \ldots, n$, then

$$\mathcal{L}_n(\theta) \propto \theta^{S_n}(1-\theta)^{nm-S_n}$$

where $S_n = \sum_{i=1}^{n} x_i$. We consider $m = 5$, $n = 20$ and observe $S_n = 33$.

The Beta(α, β) prior is conjugate:

$$\pi_0(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}.$$

Then

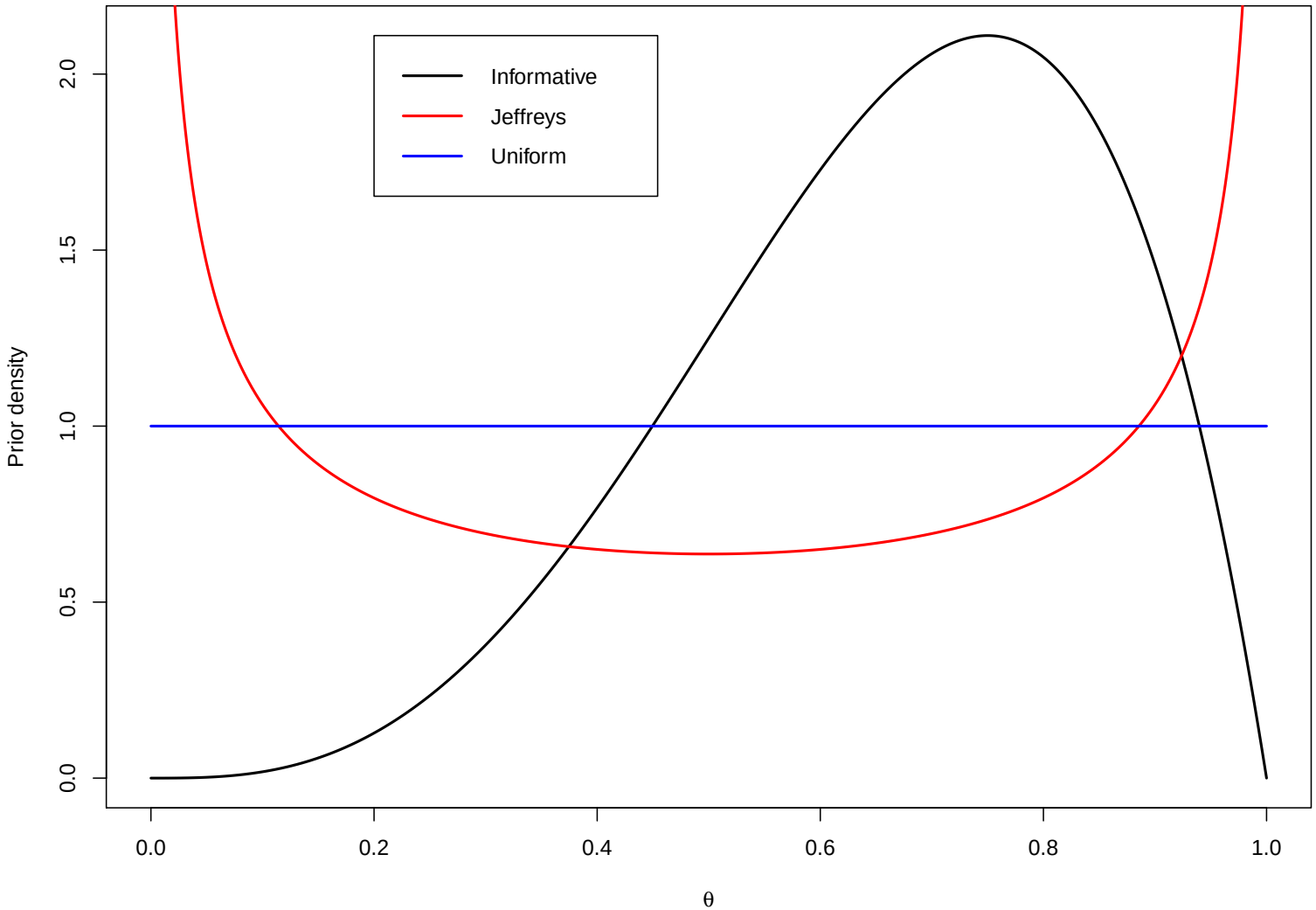
$$\pi_n(\theta) \equiv \text{Beta}(\alpha + S_n, \beta + nm - S_n).$$

Construction of Prior Distributions (contd.)

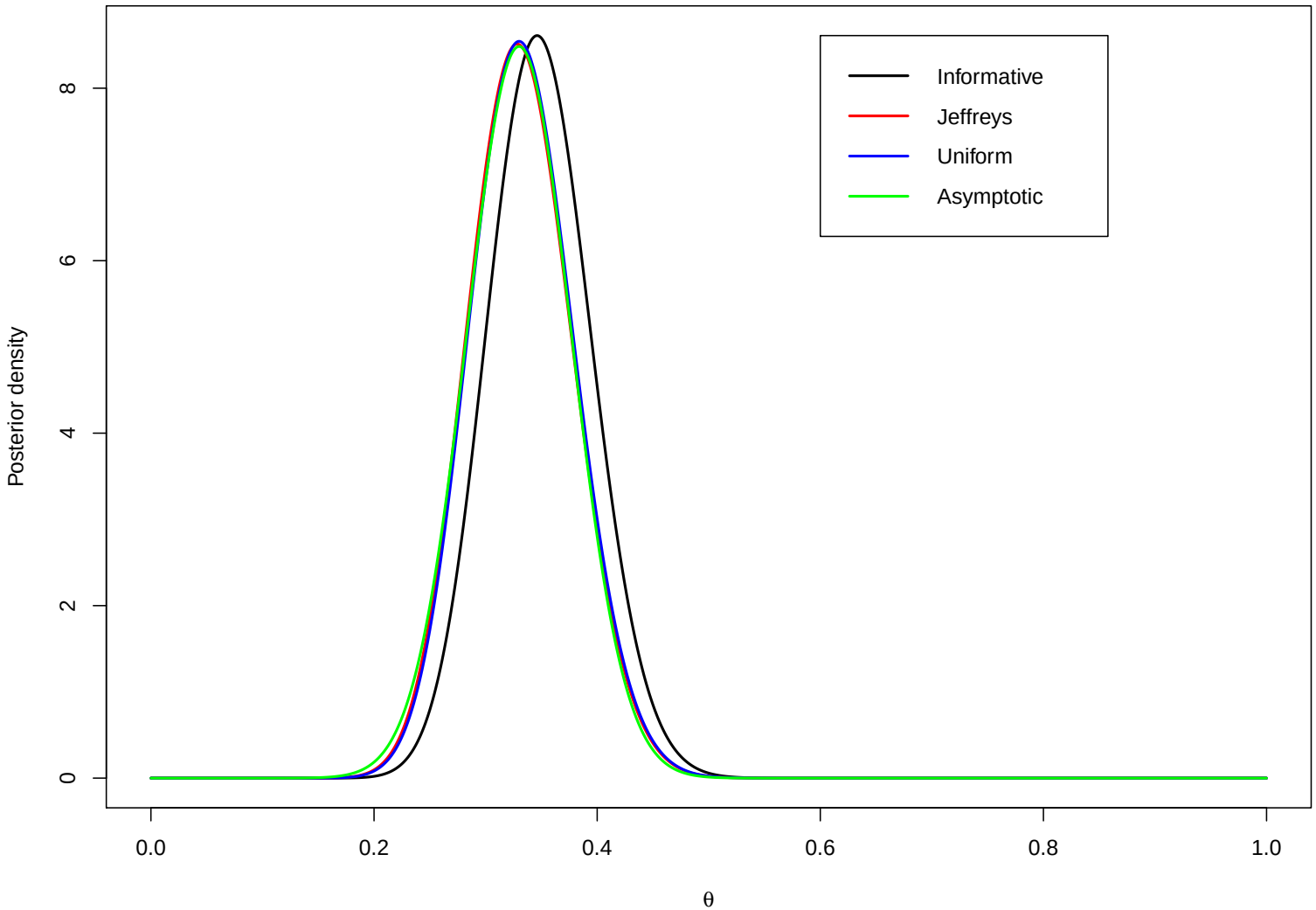
Example: Binomial case

- Informative Prior: Beta$(2, 4)$;
- Jeffreys' Prior: Beta$(1/2, 1/2)$;
- Uniform Prior: Beta$(1, 1)$;

Priors


Informative
Jeffreys
Uniform
Prior density
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
θ

Posteriors


Informative
Jeffreys
Uniform
Asymptotic
Posterior density
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
θ

Beyond the IID case

Example: Linear Regression

In the Gaussian linear model with predictors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$,

$$y_i = \mathbf{x}_i\beta + \epsilon_i \qquad \epsilon_i \sim \text{Normal}(0, \sigma^2)$$

the posterior concerned is

$$\pi_n(\beta, \sigma^2)$$

This is a $(p + 1)$ dimensional distribution.

Beyond the IID case (contd.)

Example: Linear Regression

We have that

$$\mathcal{L}_n(\beta, \sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)\right\}$$

and with a conjugate prior

$$\pi_0(\beta, \sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{\alpha/2} \exp\left\{-\frac{1}{2\sigma^2}\left[\frac{\gamma}{2\sigma^2}(\beta - \mathbf{m})^\top \mathbf{V}^{-1}(\beta - \mathbf{m})\right]\right\}$$

that is

$$\pi_0(\beta|\sigma^2) \equiv \text{Normal}(\mathbf{m}, \sigma^2\mathbf{V}) \qquad \pi_n(\sigma^2) \equiv \text{InvGamma}(\alpha/2, \gamma/2)$$

we end up with an analytically available posterior distribution.

Beyond the IID case (contd.)

Example: Linear Regression

We have that

$$\pi_n(\beta|\sigma^2) \equiv \text{Normal}(\mathbf{m}^*, \sigma^2\mathbf{V}^*)$$

and

$$\pi_n(\sigma^2) \equiv \text{InvGamma}(\alpha^*/2, \gamma^*/2)$$

that is, identical to the prior, but with the parameters updated.

Bayesian decisions

Having obtained the posterior distribution, we proceed to

- estimate θ,
- report a *credible interval*,
- process other decisions,

by specifying a *loss function*, and minimizing the *posterior expected loss*.

Could report

- posterior mean
- posterior mode
- posterior median

as potential estimates of θ.

Challenges

- If θ is d-dimensional, with d moderately large, we may be able to compute $\pi_n(\theta)$ up to proportionality, but we may not be able to compute the appropriate normalizing constant.
- If the model is non-standard, then we may have no analytical tractability, and computing estimates, credible intervals, predictions etc. may be very challenging.
- In more elaborate models, such as *hierarchical models*, the calculation is especially challenging.

Beyond the IID case

Example: Logistic Regression

In the *logistic regression model* with predictors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$,

$$\Pr[Y_i = 1 | \mathbf{x}_i; \beta] = \frac{\exp\{\mathbf{x}_i \beta\}}{1 + \exp\{\mathbf{x}_i \beta\}}$$

and the posterior concerned is

$$\pi_n(\beta).$$

This is a p dimensional distribution.

Beyond the IID case (contd.)

Example: Logistic Regression

We have that

$$\mathcal{L}_n(\beta) \propto \prod_{i=1}^{n} \left(\frac{\exp\{\mathbf{x}_i\beta\}}{1+\exp\{\mathbf{x}_i\beta\}} \right)^{y_i} \left(\frac{1}{1+\exp\{\mathbf{x}_i\beta\}} \right)^{1-y_i}.$$

There is no conjugate prior for this model. We might choose

$$\pi_0(\beta) \equiv \text{Normal}(\mathbf{m}, \mathbf{V})$$

but this does not lead to a standard form posterior distribution.

Simulation-based inference

An important realization is that rather than performing analytical calculations, we may instead focus on *simulation-based approaches*:

- rather than compute $\pi_n(\theta)$ directly, we instead produce a large sample from the distribution using *random number generation*;
- this is especially useful if the distribution is *analytically intractable*.

Simulation-based inference (contd.)

Example: Binomial case

Suppose as above

$$\pi_n(\theta) \equiv \text{Beta}(\alpha + S_n, \beta + nm - S_n)$$

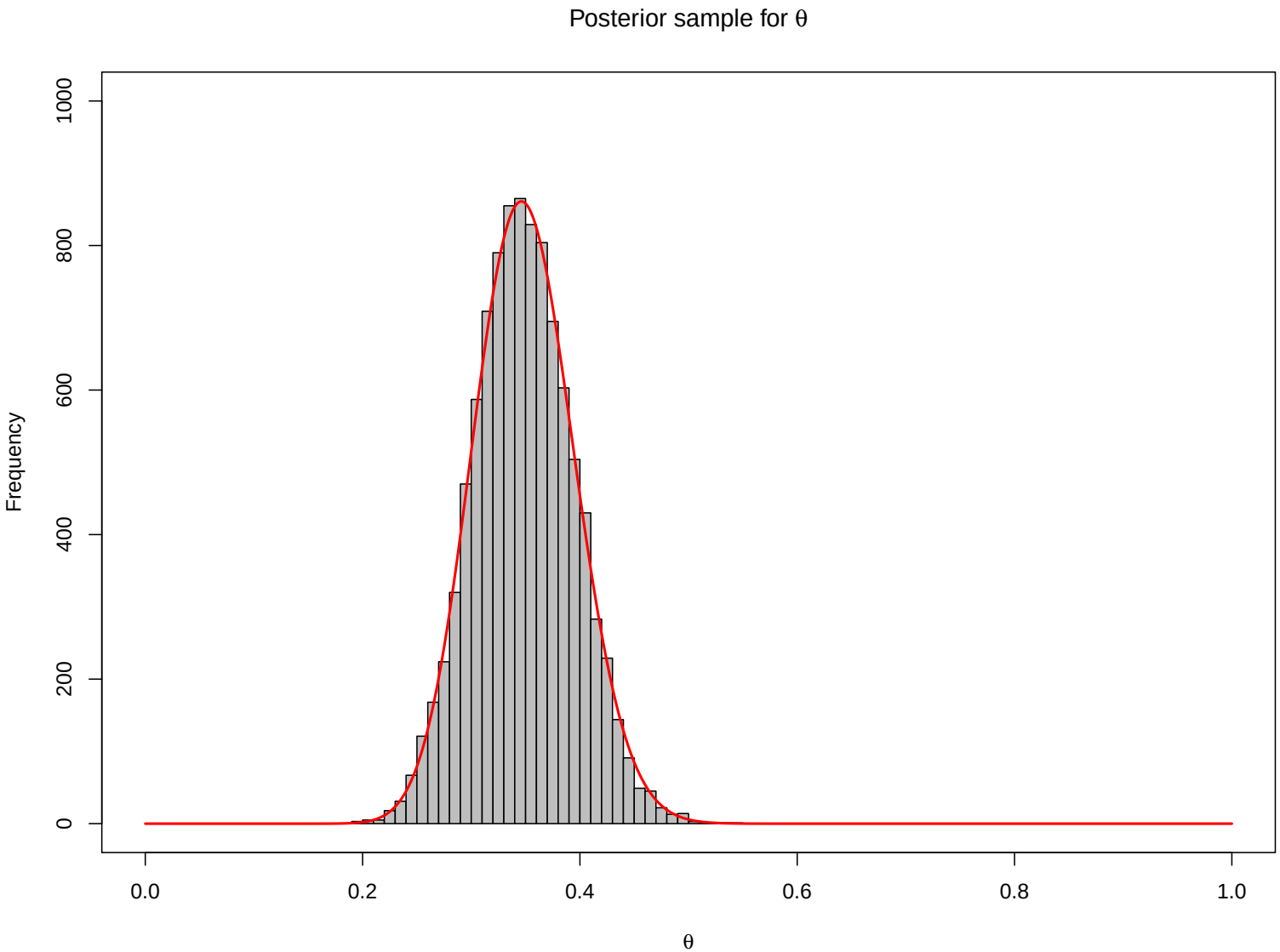
We may sample routinely from $\pi_n(\theta)$ using the `rbeta` function in `R`.

Suppose

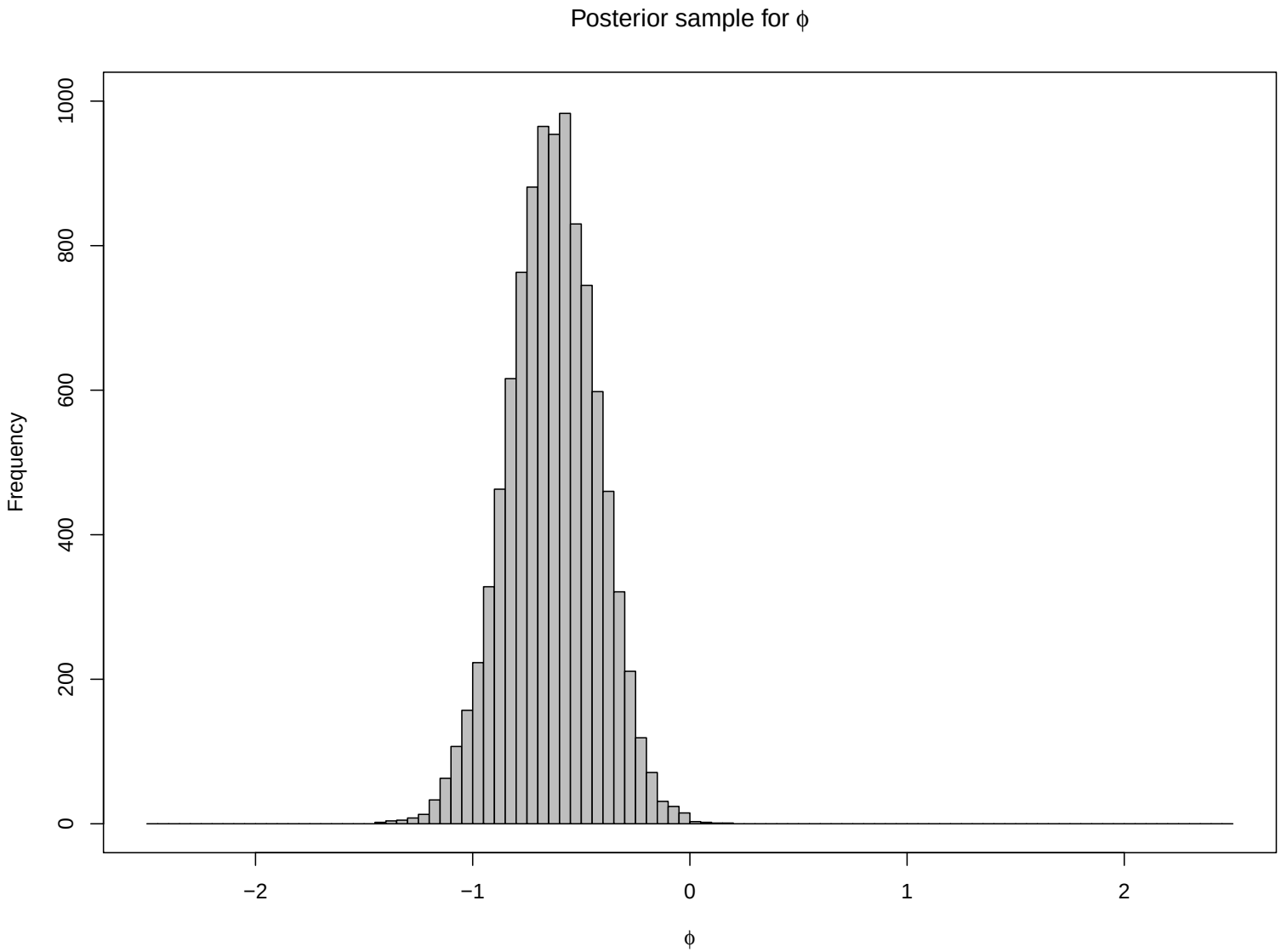
$$\phi = \log\left(\frac{\theta}{1-\theta}\right).$$

The posterior distribution for ϕ is not a standard distribution and is more difficult to compute with. However given a sample from $\pi_n(\theta)$, we may simply *transform the sample* to obtain a sample from $\pi_n(\phi)$.

Posterior for θ



Posterior for ϕ



Transformations

Example: Binomial case

Suppose

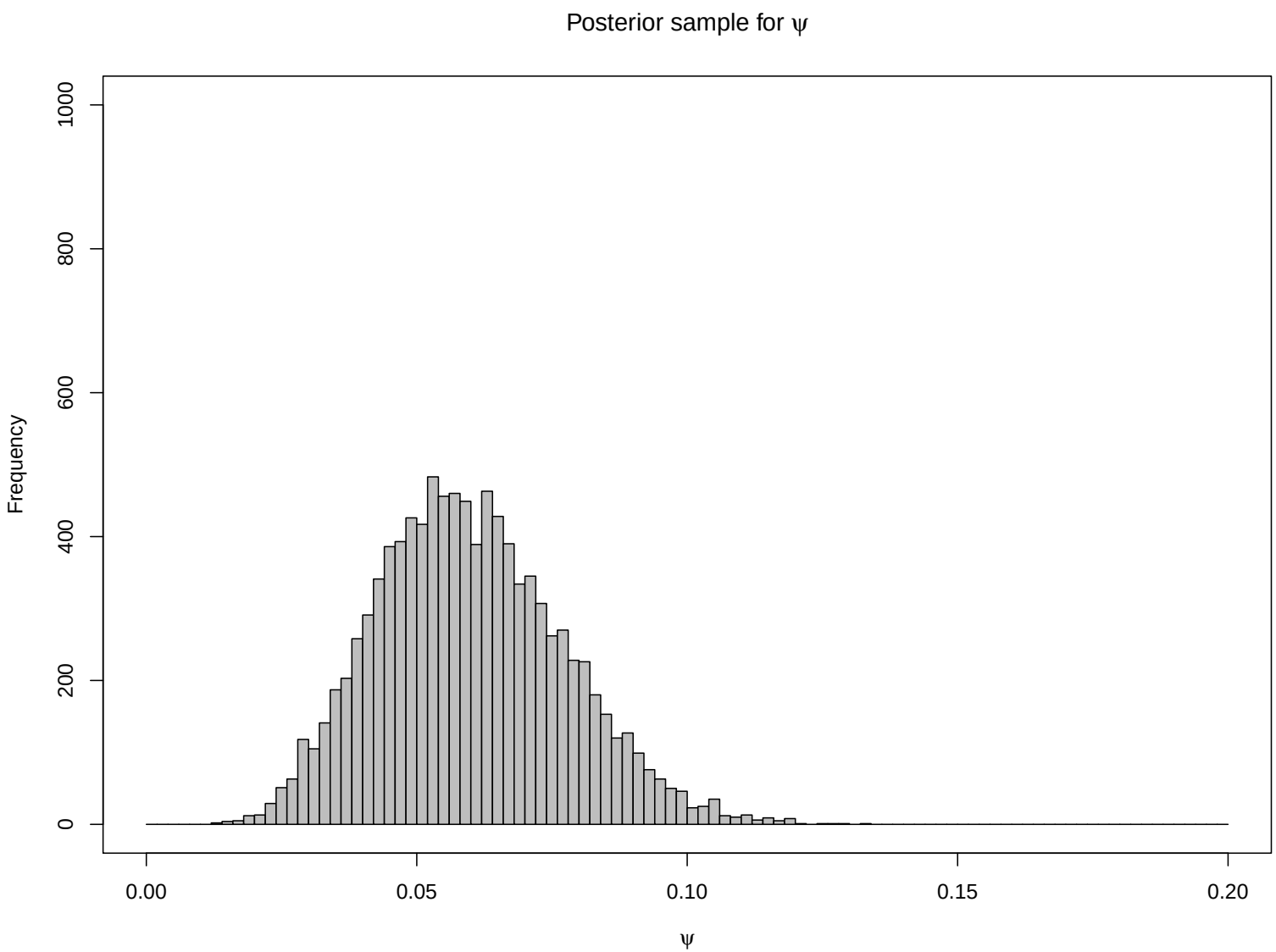
$$\psi = \theta^2\sqrt{(1-\theta)}\Phi^{-1}\left(\frac{1}{2} + \theta(1-\theta)\right)$$

where $\Phi^{-1}(x)$ is the inverse normal cdf.

This is a challenging parameterization to work with if we wish to perform this calculation analytically.

However, transforming the sample is a very straightforward numerical exercise.

Posterior for ψ



Part II

Monte Carlo Methods

Monte Carlo: basic principles

Monte Carlo methods utilize *random number generation*, and classical frequentist probability theory: as $n \longrightarrow \infty$, we have

- The ***weak law of large numbers***: that is,

  sample averages converge to expectations

  $$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} \mu$$

- The ***central limit theorem***: that is,

  sample averages have distributions that are approximately Gaussian

  $$\sqrt{n}(\overline{X}_n - \mu) \xrightarrow{d} \text{Normal}(0, \sigma^2)$$

to construct numerical approximations to integrals.

More explicitly, in the i.i.d. case,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} \int x\, dF(x) = \int x f(x)\, dx$$

where $f(x)$ is the density of the X variables. Recall that

$$\frac{1}{n}\sum_{i=1}^{n} x_i = \int x\, d\widehat{F}_n(x)$$

where $\widehat{F}_n$ is the *empirical cdf*.

In essence, Monte Carlo methods replace integrals with respect to F by integrals with respect to $\widehat{F}_n$, and then rely on the convergence of the latter to the former.

Monte Carlo: basic principles

We consider approximating the integral

$$\int g(x)p(x)\,dx$$

for probability density $p(x)$ by the sum

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

where $x_1, \ldots, x_N$ are an i.i.d sample from p.

That is, we

- **sample** from $p(x)$ to get $x_1, \ldots, x_n$.
- **average** $g(x_1), \ldots, g(x_n)$.

Monte Carlo: basic principles (contd.)

We need to establish

- whether this approximation works
- the accuracy of the approximation
- how the samples from $p(x)$ are obtained.

Importance sampling

The identity

$$\int g(x)p(x)\,dx = \int g(x)p(x)\frac{p_0(x)}{p_0(x)}\,dx = \int \frac{g(x)p(x)}{p_0(x)}p_0(x)\,dx$$

where p_0 is a probability density with support including the support of p, demonstrates that

$$\mathbb{E}_p[g(X)] = \mathbb{E}_{p_0}\left[\frac{g(X)p(X)}{p_0(X)}\right]$$

so that an estimator of the LHS is

$$\widehat{I}_N^{(p_0)}(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{g(X_i)p(X_i)}{p_0(X_i)}$$

where $X_1, \ldots, X_N \sim p_0(.)$.

Importance sampling

$\widehat{I}_N^{(p_0)}$ is termed the *importance sampling* estimator.

p_0 is termed the *importance sampling density*. By careful choice of p_0, the estimator can have better performance than the Monte Carlo estimator in finite samples.

Note that

$$\widehat{I}_N^{(p_0)}(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{p(X_i)}{p_0(X_i)}g(X_i) = \frac{1}{N}\sum_{i=1}^{N} w_0(X_i)g(X_i)$$

say, where

$$w_0(X_i) = \frac{p(X_i)}{p_0(X_i)}$$

is the *importance sampling weight*.

Importance sampling

Note that

$$\mathbb{E}_{p_0}\left[\frac{p(X)}{p_0(X)}\right] = \int p(x)\,dx = 1$$

so

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} w_0(X_i)\right] = 1$$

although for any realization

$$\frac{1}{N}\sum_{i=1}^{N} w_0(X_i) \neq 1$$

in general.

Higher Dimensions

All of the previous results carry across to the case where the target integral is an integral in dimension higher than one.

- $g(\mathbf{x})$ is a scalar function of vector argument $\mathbf{x}$,
- $p(\mathbf{x})$ and $p_0(\mathbf{x})$ are multivariate densities.

In higher dimensions, in general, many more random samples are needed to obtain sufficient accuracy than in the univariate case.

Random number generation

Monte Carlo and Importance sampling both require ready access to random samples from univariate or multivariate distributions.

There are many straightforward techniques available to obtain random samples from standard distributions once a random sample of Uniform random variables is available:

- **CDF inversion:** If $U \sim Uniform(0,1)$, and F_X is a cdf with inverse F_X^{-1}, then

$$X = F_X^{-1}(U) \sim F_X$$

(the *probability integral transform*). If F_X is discrete, define

$$F_X^{-1}(u) = \max\{x : F_X(x) \leqslant u\}$$

Random number generation

- **Transformation:**
  - Exponential, Gamma, Beta
  - Normal
    - Box-Müller
    - Polar Marsaglia
  - Student, Fisher, Multivariate Normal
- **Summation:** The distribution of sums of independent identically distributed random variables.

R has many random number generation tools for standard distributions.

Random number generation

More advanced methods include:

- **Rejection sampling**
- **Adaptive rejection sampling**

Random Number Generators Appendix

Part III

Markov chain Monte Carlo

Markov chain Monte Carlo

Generating variates from a probability distribution $p(x)$ is not straightforward when

- x is high-dimensional (but not Gaussian),
- $p(x)$ is of non-standard form.

These difficulties occur often (but not exclusively) in Bayesian inference.

We seek a way of simulating from $p(x)$ in this situation.

A brief introduction to Markov chains

A *Markov chain* is a countable sequence of random variables, $\{X_t\}$, for which

$$\Pr[X_t \in B | X_1 = x_1, \ldots, X_{t-1} = x_{t-1}] = \Pr[X_t \in B | X_{t-1} = x_{t-1}]$$

that is, X_t is conditionally independent of $X_0, \ldots, X_{t-2}$ given X_{t-1}.

In the simplest case, the $\{X_t\}$ takes values on the finite (discrete) state space

$$\mathcal{S}_X = \{s_1, \ldots, s_d\}$$

and the chain is *homogeneous*, that is, the stochastic properties of $\{X_t\}$ do not change with time.

A brief introduction to Markov chains

A homogeneous, discrete state-space Markov chain is characterized by its initial state X_0, and its *transition matrix* P, a $d \times d$ *stochastic* matrix whose rows sum to one, such that

$$P_{ij} = \Pr[X_t = s_j | X_{t-1} = s_i]$$

describes the entire set of one-step ahead conditional probabilities. Denote by

$$p_{ij}(k) = \Pr[X_{t_0+k} = s_j | X_{t_0} = s_i]$$

the k-step ahead probabilities. Note that if $P(k)$ is the matrix of k-step ahead transition probabilities, then

$$P(k) = P^k$$

A brief introduction to Markov chains

The properties of the chain depend on P. The chain is

- *irreducible* if $p_{ij}(k) > 0$, for all i, j, and at least one k.
- *aperiodic* if all states have period 1 (returns to state i can occur after any number of steps). The *period* of state i is defined as the greatest common divisor of the set of possible return times, $\mathcal{R}$,

$$\mathcal{R} = \{r : \Pr[X_r = s_i | X_0 = s_i] > 0\}$$

- *recurrent* if all states are recurrent, that is, the probability of returning to each state in a finite number of steps is positive. Let $T_i = \inf\{k : X_k = i | X_0 = i\}$. State i is recurrent if and only if $\Pr[T_i = \infty] = 0$, and *transient* otherwise. If $\mathbb{E}[T_i] < \infty$, state i is termed *positive recurrent*, otherwise it is termed *null recurrent*.

The *stationary* or *equilibrium distribution*, π, of a homogeneous Markov chain is the $1 \times d$ vector of probabilities such that

$$\pi = \pi P$$

that is, for each i,

$$\pi_i = \sum_j \pi_j P_{ji}.$$

A brief introduction to Markov chains (contd.)

An irreducible chain has a stationary distribution if and only if all of its states are positive recurrent, in which case π is unique. Given P, π can be computed as

$$\lim_{k \longrightarrow \infty} P^k = \mathbf{1}\pi.$$

where $\mathbf{1}$ is the $d \times 1$ vector of 1s. A matrix P with the stationary distribution π can be computed by solving the system of equations $\pi = \pi P$.

Key aspects of the stationary distribution are that

- As the k-step ahead probability matrix P^k converges to a matrix with d identical rows, the Markov chain can eventually "forget" its initial value X_0.

- Realizations of $\{X_t\}$ exhibit statistical properties that demonstrate (strong) convergence to the stationary distribution, that is, for $i = 1, \ldots, d$,

$$\lim_{n \longrightarrow \infty} \frac{\sum_{k=1}^{n} \mathbb{1}_{\{s_i\}}(X_k)}{n} = \pi_i$$

A homogeneous Markov chain with stationary distribution π is *reversible* if

$$\pi_i P_{ij} = \pi_j P_{ji}$$

for all states i and j. This is also termed the *detailed balance* condition.

Note that **if** this equation holds for a specified π, **then** this implies that the P has been specified so as to have stationary distribution π.

Note that the $\{X_t\}$ are **dependent** random variables, so the standard frequentist asymptotic laws do not directly apply.

However, the *ergodic theorem* applies for irreducible, aperiodic and positive recurrent Markov chains, in particular

$$\frac{1}{N}\sum_{t=1}^{N} g(X_t) \xrightarrow{a.s.} \mathbb{E}_\pi[g(X)]$$

for all bounded functions g, provided

$$\mathbb{E}_\pi[|g(X)|] < \infty$$

A Central Limit Theorem result also holds under mild regularity conditions, specifically,

$$\sqrt{N}\left(\frac{1}{N}\sum_{t=1}^{N} g(X_t) - \mathbb{E}_\pi[g(X)]\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2(g))$$

where

$$\sigma^2(g) = \mathrm{Var}_\pi[g(X_0)] + 2\sum_{t=1}^{\infty} \mathrm{Cov}_\pi[g(X_0), g(X_t)].$$

Discrete Markov chain: example

$d = 2$

Consider $d = 2$, with

$$P = \begin{bmatrix} 0.3 & 0.7 \\ 0.9 & 0.1 \end{bmatrix}$$

Then $\pi = (9/16, 7/16)$. Here

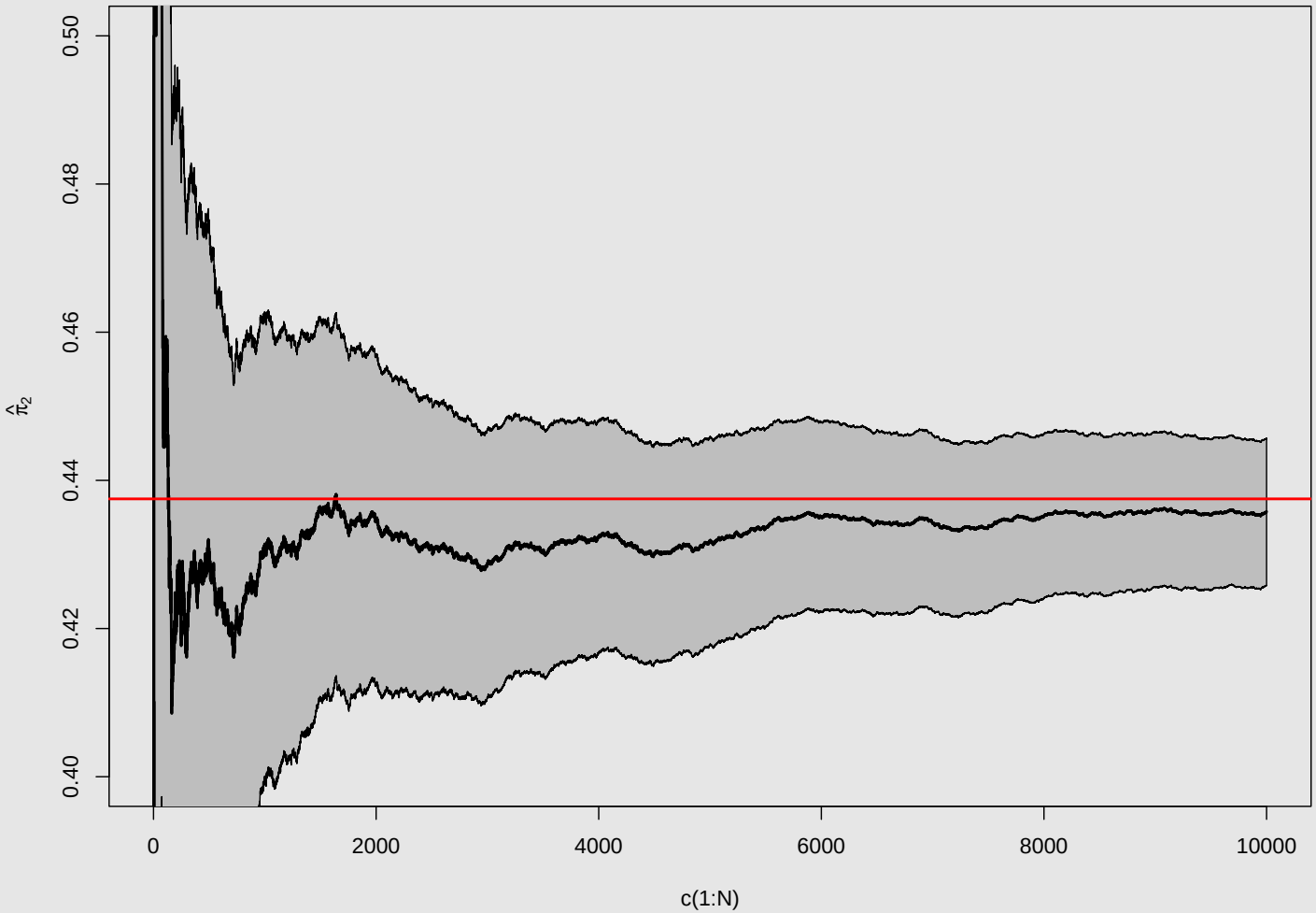
$$\pi_1 P_{12} = \frac{9}{16} \times \frac{7}{10} = \frac{63}{160} \qquad \pi_2 P_{21} = \frac{7}{16} \times \frac{9}{10} = \frac{63}{160}$$

so this chain is reversible.

Discrete Markov chain: example

$d = 2$

Relative frequency of being in state 2 over 10000 steps.



Constructing reversible chains

In the 2×2, for a reversible chain, we require

$$\pi_1 P_{12} = (1 - \pi_1) P_{21}$$

Suppose

$$\begin{aligned} P_{12} &= \min\left\{1, \frac{1-\pi_1}{\pi_1}\right\} \\ P_{21} &= \min\left\{1, \frac{\pi_1}{1-\pi_1}\right\} \end{aligned}$$

Then

$$\begin{aligned} \pi_1 P_{12} &= \pi_1 \min\left\{1, \frac{1-\pi_1}{\pi_1}\right\} = \min\left\{\pi_1, 1-\pi_1\right\} = \min\left\{1-\pi_1, \pi_1\right\} \\ &= (1-\pi_1)\min\left\{1, \frac{\pi_1}{1-\pi_1}\right\} = (1-\pi_1)P_{21} \end{aligned}$$

Constructing reversible chains

The above Markov chain we can think of as acting as follows:

- **If** $X_t = 1$: propose setting $X_{t+1} = 2$, but only accept this move with probability

$$\min\left\{1, \frac{1-\pi_1}{\pi_1}\right\}$$

  otherwise set $X_{t+1} = 1$.
- **If** $X_t = 2$: propose setting $X_{t+1} = 1$, but only accept this move with probability

$$\min\left\{1, \frac{\pi_1}{1-\pi_1}\right\}$$

  otherwise set $X_{t+1} = 2$.

Constructing reversible chains

A generalization of this approach is as follows:

- If $X_t = 1$, simulate Z_t on the set $\{1, 2\}$ with probabilities (q_{11}, q_{12}).
- If $X_t = 2$, simulate Z_t on the set $\{1, 2\}$ with probabilities (q_{21}, q_{22}).

Constructing reversible chains

- If $X_t = 1$, if $Z_t = 2$, set $X_{t+1} = Z_t = 2$ with probability

$$\alpha_{12} = \min\left\{1, \frac{\pi_2}{\pi_1}\frac{q_{21}}{q_{12}}\right\}$$

  otherwise set $X_{t+1} = X_t = 1$.

- If $X_t = 2$, if $Z_t = 1$, set $X_{t+1} = Z_t = 1$ with probability

$$\alpha_{21} = \min\left\{1, \frac{\pi_1}{\pi_2}\frac{q_{12}}{q_{21}}\right\}$$

  otherwise set $X_{t+1} = X_t = 2$.

Constructing reversible chains

In this case, we have the transition probabilities given as follows:

$$\begin{aligned} P_{12} &= \Pr[X_{t+1} = 2 | X_t = 1] = q_{12}\alpha_{12} \\ P_{21} &= \Pr[X_{t+1} = 1 | X_t = 2] = q_{21}\alpha_{21} \end{aligned}$$

to therefore

$$\pi_1 P_{12} = \pi_1 q_{12}\alpha_{12} = \pi_1 q_{12} \min\left\{1, \frac{\pi_2}{\pi_1}\frac{q_{21}}{q_{12}}\right\} = \min\left\{\pi_1 q_{12}, \pi_2 q_{21}\right\}$$

$$\pi_2 P_{21} = \pi_2 q_{21}\alpha_{21} = \pi_2 q_{21} \min\left\{1, \frac{\pi_1}{\pi_2}\frac{q_{12}}{q_{21}}\right\} = \min\left\{\pi_2 q_{21}, \pi_1 q_{12}\right\}$$

and

$$\pi_1 P_{12} = \pi_2 P_{21}.$$

Constructing reversible chains

This allows the generalization to the more general discrete state space $\{1, 2, \ldots, d, \ldots\}$. Suppose that π is an arbitrary discrete distribution. Let matrix Q define the *proposal probabilities*

$$[Q]_{ij} = \Pr[Z_t = j | X_t = i]$$

Define the *acceptance probabilities*

$$\alpha_{ij} = \min \left\{ 1, \frac{\pi_j}{\pi_i} \frac{q_{ji}}{q_{ij}} \right\}$$

If $X_t = i$, set $X_{t+1} = Z_t = j$ with probability α_{ij}. Otherwise set $X_{t+1} = X_t = i$.

Constructing reversible chains

This Markov chain satisfies detailed balance

$$\pi_i P_{ij} = \pi_j P_{ji} \qquad \text{for all } i, j$$

provided it is irreducible, aperiodic and positive recurrent.

Note that the rows of Q must sum to 1 as

$$\sum_{j=1}^{\infty} \Pr[Z_t = j | X_t = i] = 1$$

so Q defines a stochastic proposal (or *transition*) matrix.

Constructing reversible chains: example

Poisson distribution

Suppose, for $\lambda > 0$,

$$\pi_i = \frac{e^{-\lambda}\lambda^i}{i!} \qquad i = 0, 1, 2, \ldots.$$

Suppose

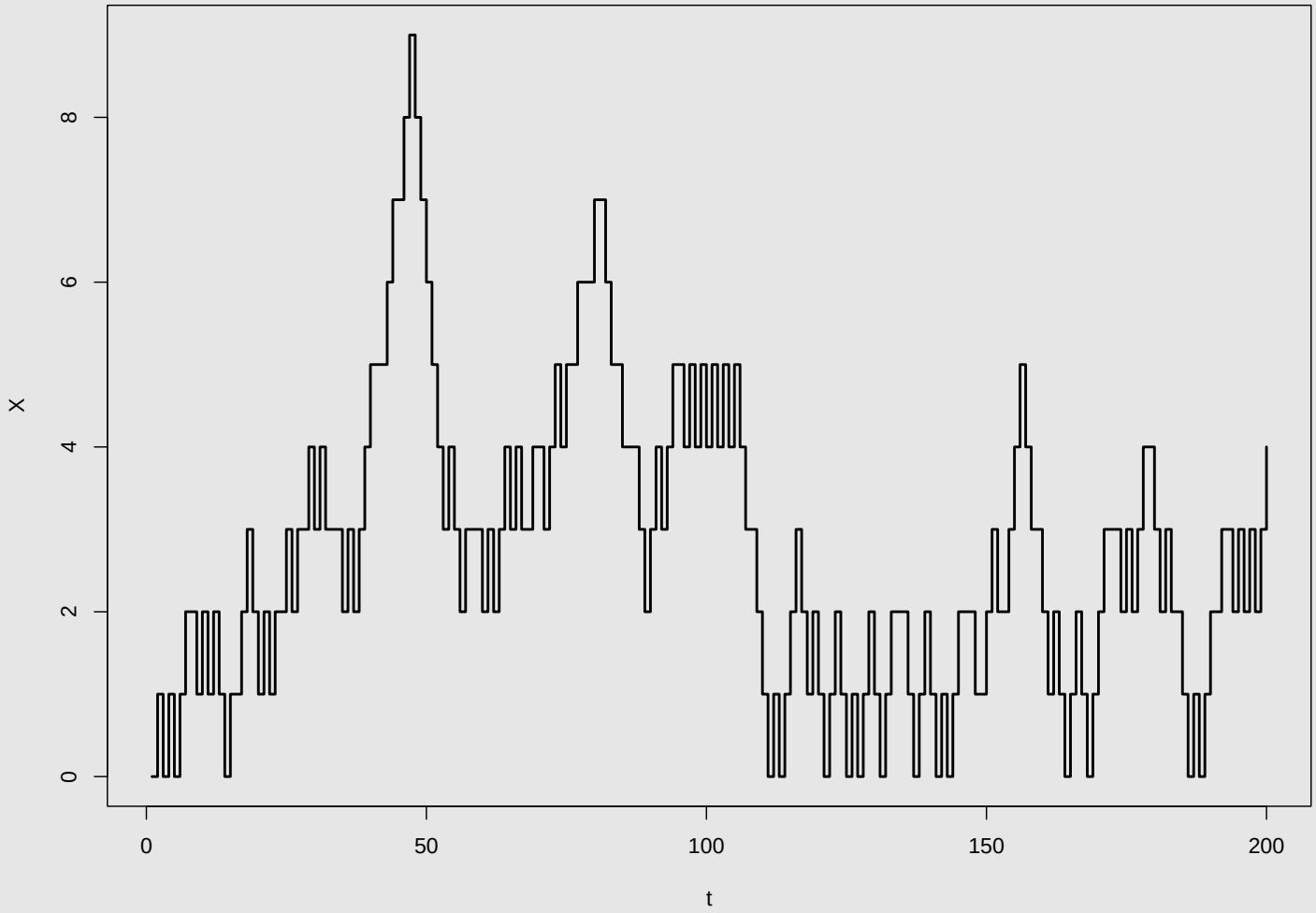
$$q_{ij} = \begin{cases} 1 & i = 0, j = 1 \\ \frac{1}{2} & i \geqslant 1, j = i-1, i+1 \\ 0 & \text{otherwise} \end{cases}$$

That is, Z_t is proposed uniformly on $\{x_t - 1, x_t + 1\}$, unless $X_t = 0$, in which case $Z_t = 1$ is proposed.

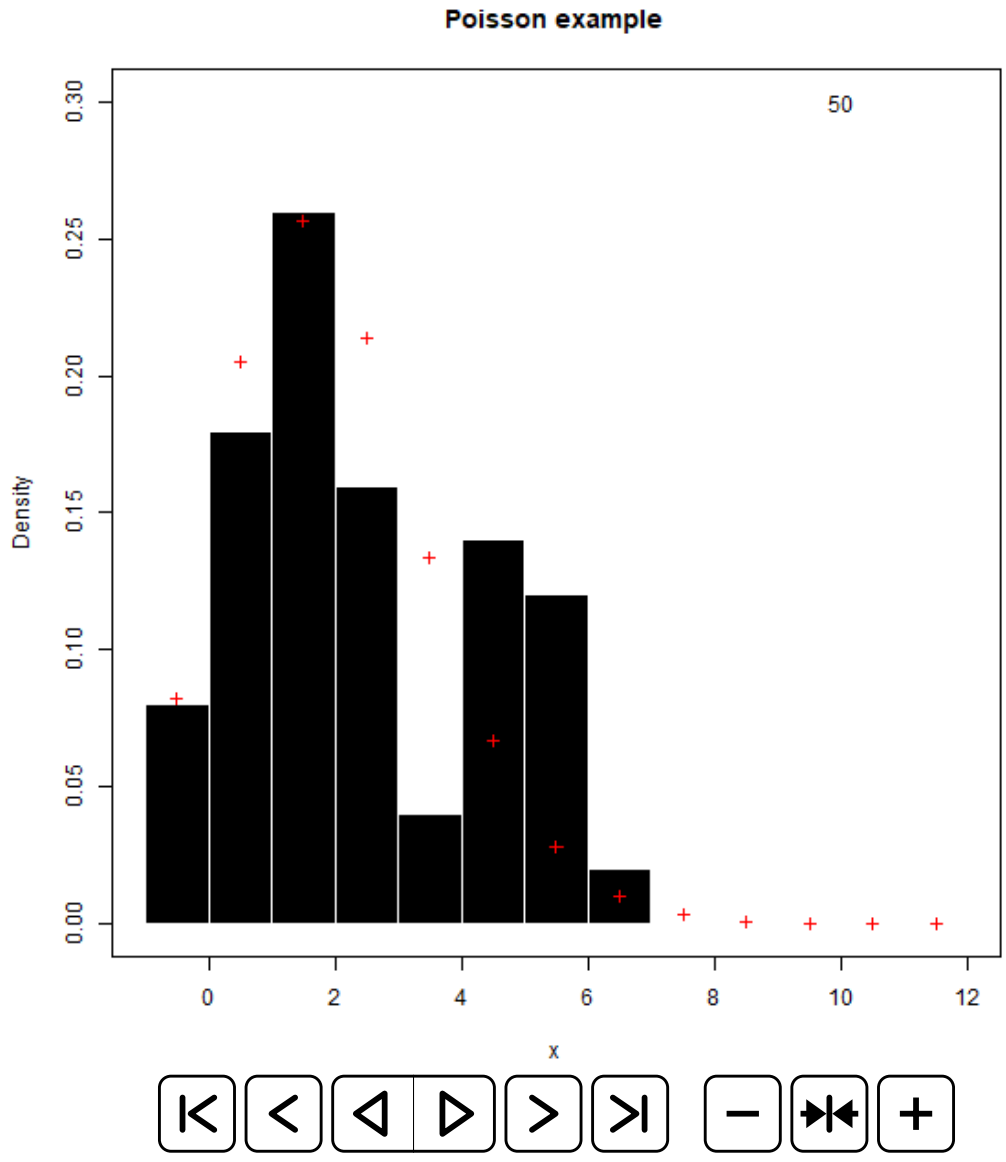
Constructing reversible chains: example

Poisson distribution

First 200 steps of the chain starting at $X_0 = 0$ with $\lambda = 2.5$.



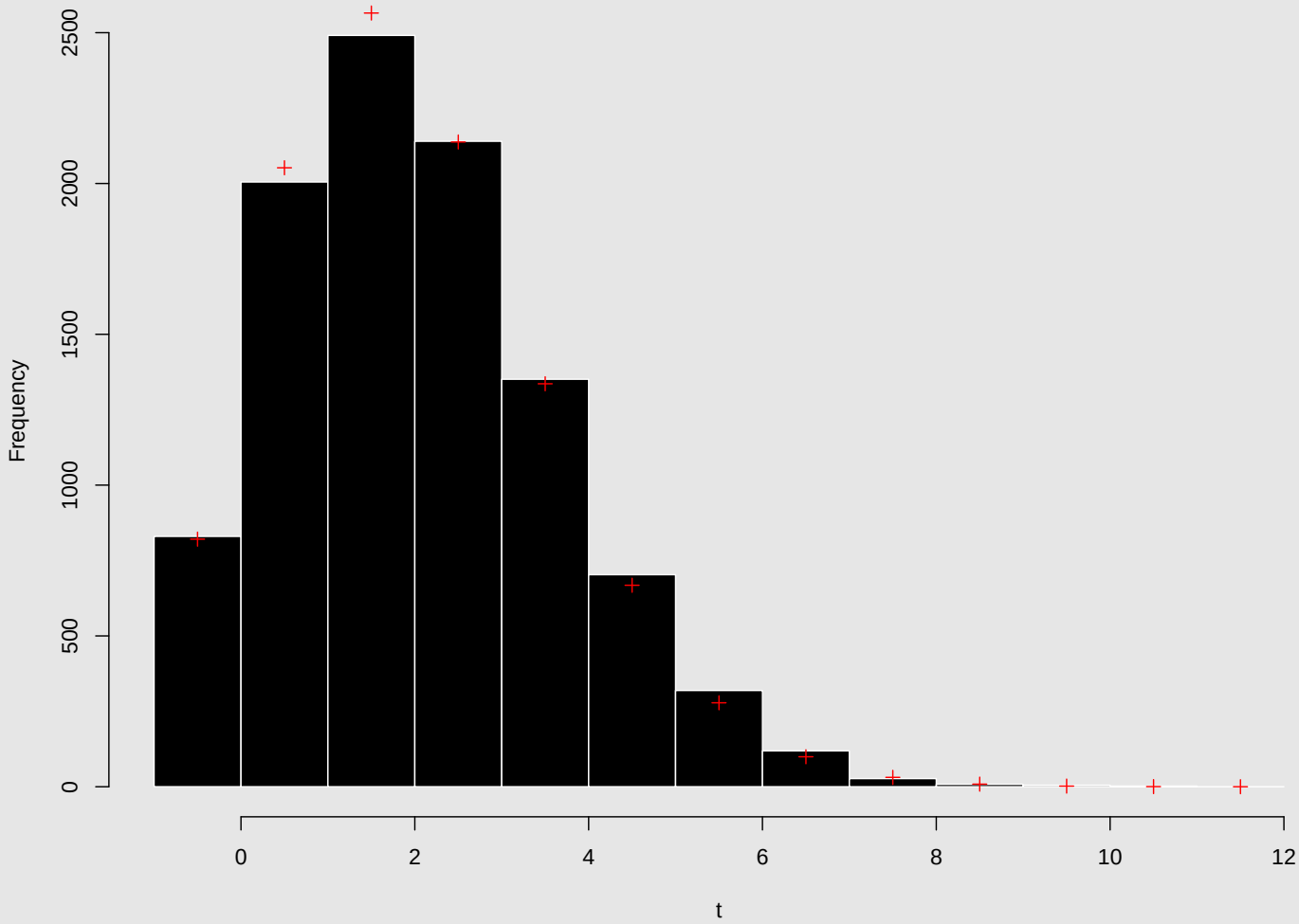
Constructing reversible chains: example



Constructing reversible chains: example

Poisson distribution

Histogram of states visited over $N = 10000$ steps.



Constructing reversible chains: example

Poisson distribution

Relative frequencies ($\widehat{\pi}_i$) of states visited over $N = 10000$ steps, with true Poisson probabilities (π_i).

i	0	1	2	3	4	5
$\widehat{\pi}_i$	0.0830	0.2005	0.2491	0.2140	0.1351	0.0703
π_i	0.0821	0.2052	0.2565	0.2138	0.1336	0.0668

i	6	7	8	9	10	11
$\widehat{\pi}_i$	0.0319	0.0119	0.0027	0.0008	0.0005	0.0002
π_i	0.0278	0.0099	0.0031	0.0009	0.0002	0.0000

Discrete Markov chains: Recap

For an arbitrary discrete distribution, π, we can now construct a Markov chain that has π as its stationary distribution, and can produce dependent samples from the π by running the Markov chain and collecting the generated states.

- specify the stochastic matrix Q
- initialize the chain by setting X_0
- for each t, if $X_t = i$, use the ith row of Q as a discrete distribution for proposing Z_t
- If $Z_t = j$, accept $X_{t+1} = j$ with probability α_{ij}

$$\alpha_{ij} = \min \left\{ 1, \frac{\pi_j}{\pi_i} \frac{q_{ji}}{q_{ij}} \right\}$$

otherwise set $X_{t+1} = X_t = i$.

Continuous State-Space Markov Chains

The theory above extends (reasonably straightforwardly) to continuous state spaces, that is, the countable state set $\{s_1, ..., s_d\}$ is replaced by a continuum of possible values, $\mathbb{X}$.

In this case, instead of having a transition matrix, we have a *transition kernel*

$$P(x, B) = \int_B P(x, z)\, dz$$

$P(x, B)$ determines the probability of making the transition from current value x into the set $B \subset \mathbb{X}$ in any given step.

Continuous State-Space Markov Chains

We retain the *discrete time* nature of the Markov chain, and again consider outcome sequences $\{x_1, x_2, ..., x_n, ...\}$).

Transitions are implemented using a *transition density*

$$P(x, z) \equiv P(x \to z)$$

which specifies a conditional probability density in z, given the current value x, for $x, z \in \mathbb{X}$.

Continuous State-Space Markov Chains

By analogy with the discrete case, the stationary distribution π for the continuous state space chain must satisfy

$$\pi(x) = \int P(z, x)\pi(z)\, dz$$

A reversible chain must satisfy detailed balance

$$\pi(x)P(x, z) = \pi(z)P(z, x)$$

for all x and z. Given P, we can in theory solve for π.

In the context of sampling from probability distributions, we wish to specify π, and then find a P such that its equilibrium distribution is π.

The Metropolis-Hastings Algorithm

We attempt to mimic the construction of a Markov chain with stationary distribution π used in the discrete case.

Let Q be any proposal (transition) kernel suitable for moving (exhaustively) around $\mathbb{X}$, with transition density q such that

$$q\left(z, x\right) = q(z \longrightarrow x) > 0$$

for all x, z.

In fact, this can be relaxed to the condition that requires $Q^n\left(x, z\right) > 0$ for all $x, z \in \mathbb{X}$, separated by n steps in the chain).

The Metropolis-Hastings Algorithm

Then, for $z \neq x$, define

$$P(x, z) = q(x, z)\,\alpha(x, z)$$

where

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

defines an *acceptance probability* for the move from x to z.

The Metropolis-Hastings Algorithm

Under this transition kernel or density P with transition density if the current state of the chain at step n is $X_n = x$, then the next value of the chain is either

- a **new value** $X_{n+1} = z$, generated from the conditional density $q(x, z)$,
- or the **current value** $X_{n+1} = x$.

The value z the proposed or *candidate* state.

Thus, starting from the n^{th} step when $x_n = x$, we have the following algorithm for implementing the continuous state space Markov chain:

The Metropolis-Hastings Algorithm

1. Generate candidate z from the conditional density $q(.,.)$ given x
2. Compute $\alpha(x, z)$
3. Generate u from $Uniform(0, 1)$
   - if $u \leqslant \alpha(x, z)$, **accept** the move to z and set $X_{n+1} = z$
   - if $u > \alpha(x, z)$, **reject** the move to z and set $X_{n+1} = x$
4. Return to 1 to generate X_{n+2}

and so on.

This is the *Metropolis-Hastings algorithm*

The Metropolis Algorithm

The general algorithm above has some special cases of interest. If q is chosen such that

$$q\left(x,z\right)=q\left(z,x\right)$$

so that q is symmetric in its arguments, then

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

and the move to z is accepted with certainty if the target probability density at z is higher than at x.

The Metropolis Algorithm

A simple symmetric transition density has

$$Z|X_n = x \sim \mathcal{N}\left(x, \sigma_q^2\right)$$

Choosing σ_q^2 small encourages many small moves.

This is the original Markov chain simulation algorithm, known as the *Metropolis Algorithm*.

Many such "local" moves can be proposed. Note that it is important to respect any parameter constraints in the proposal.

Independence Metropolis-Hastings

Another common choice is the *independence Metropolis-Hastings algorithm*, where

$$q(x, z) = q(z)$$

that is, independent of the current value of the chain. This still defines a Markov chain as

$$p(x, z) = q(z)\,\alpha(x, z)$$

still depends on x through $\alpha(x, z)$.

A good independence Markov chain (that traverses $\mathbb{X}$ quickly) is more difficult to construct without knowledge of π.

However, if π can be well-approximated by a density q (as in rejection sampling), then this method can work well.

Metropolis-Hastings algorithm

Gamma density

Suppose, for $\gamma > 0$

$$\pi(x) = \frac{1}{\Gamma(\gamma)} x^{\gamma-1} e^{-x} \qquad x > 0.$$

Suppose, first that $q(x, z)$ is specified as a *reflected* normal density, that is, we propose z by simulating

$$Y|X_t = x \sim \mathcal{N}(x, \sigma_q^2),$$

and setting $Z = |Y|$.

Gamma density

Note that

$$\Pr[Z \leqslant z|X = x] = \Pr[|Y| \leqslant z|X = x] = \Pr[-z \leqslant Y \leqslant z|X = x]$$

so therefore

$$\Pr[Z \leqslant z|X = x] = \Phi((z - x)/\sigma_q) - \Phi((-z - x)/\sigma_q)$$

and, on differentiation wrt z,

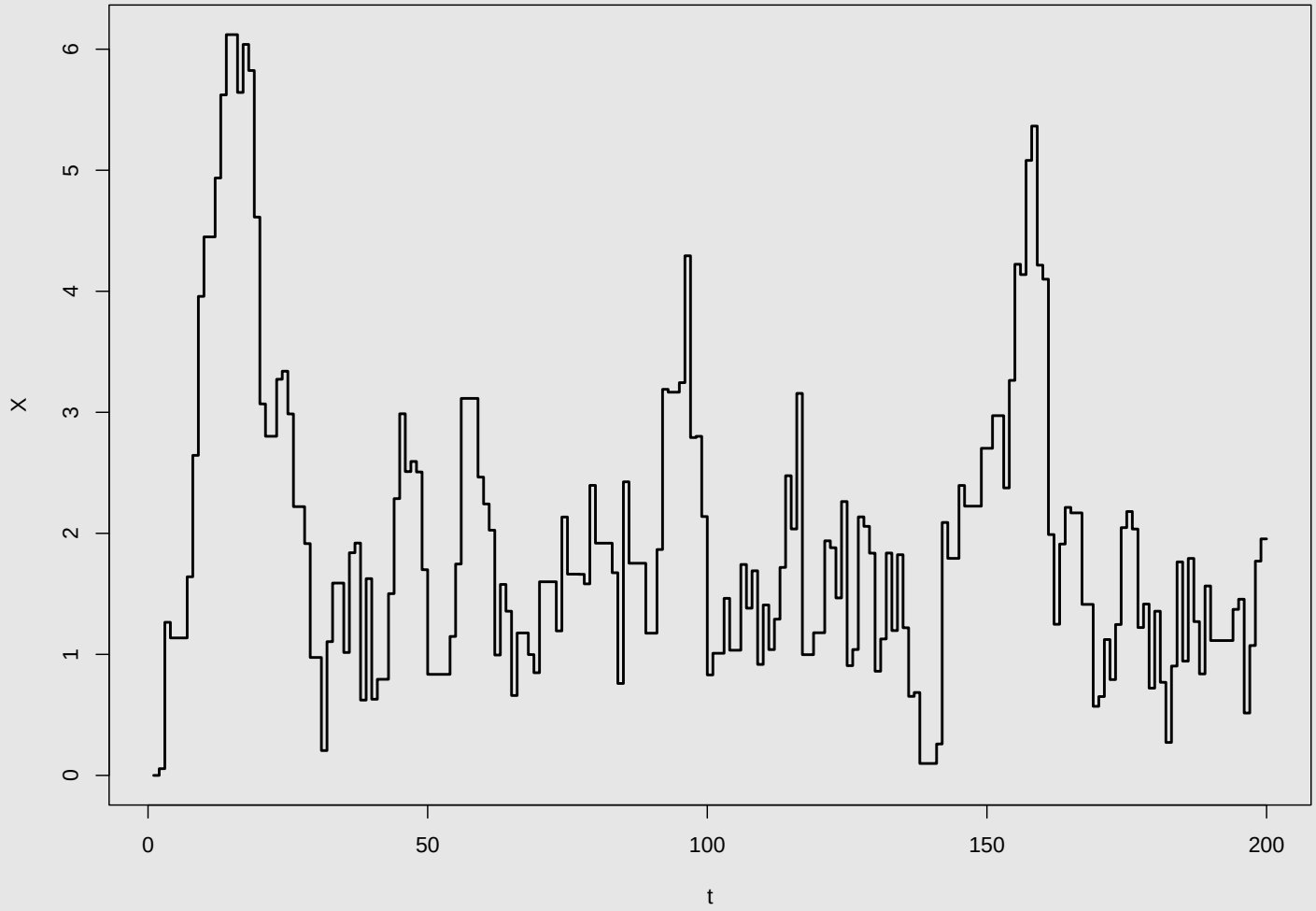
$$q(x, z) = \frac{1}{\sigma_q}(\phi((z - x)/\sigma_q) + \phi((-z - x)/\sigma_q)) = q(z, x)$$

as ϕ is an even function.

Metropolis-Hastings algorithm

Gamma density

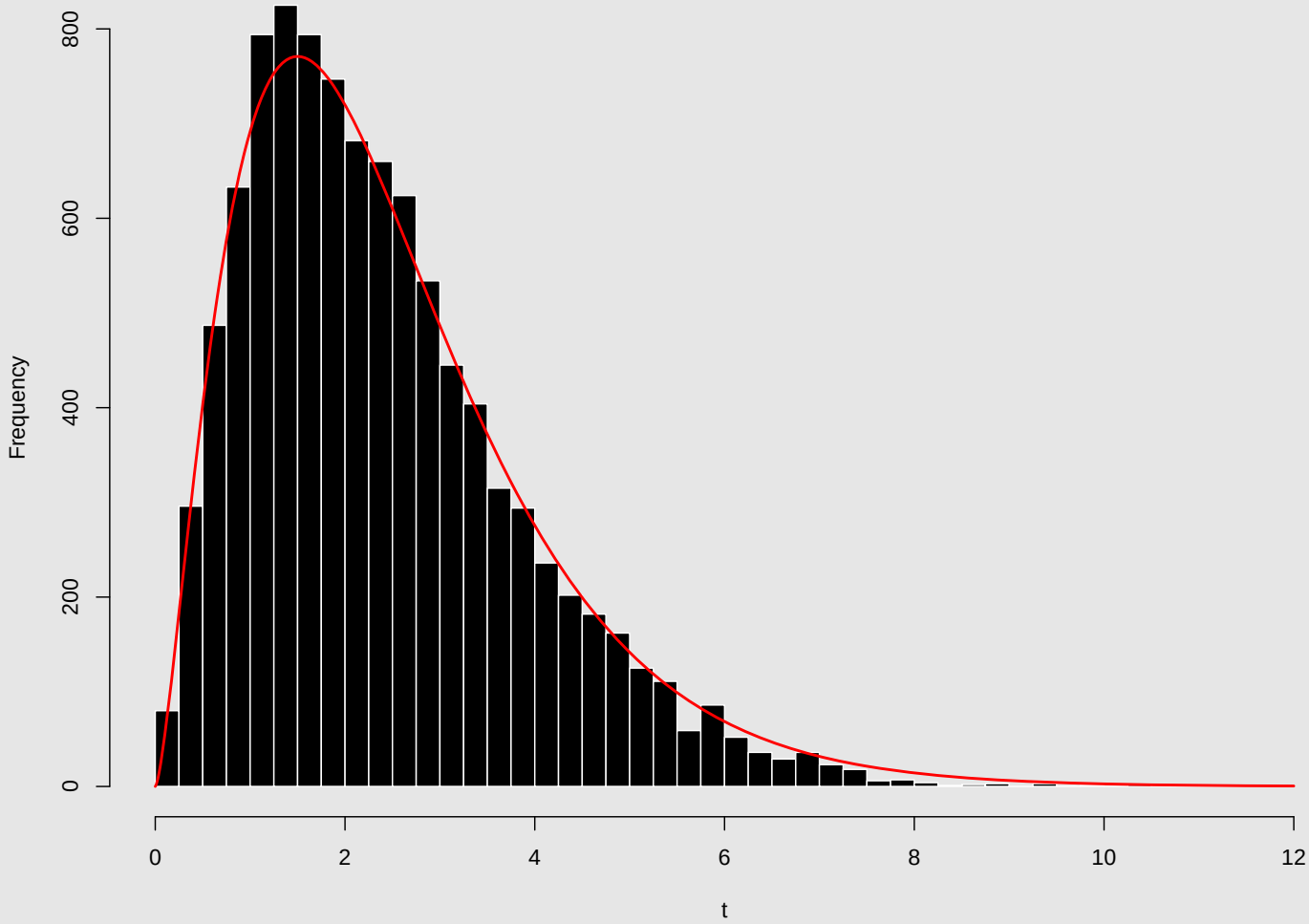
First 200 steps of the chain starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 1$.



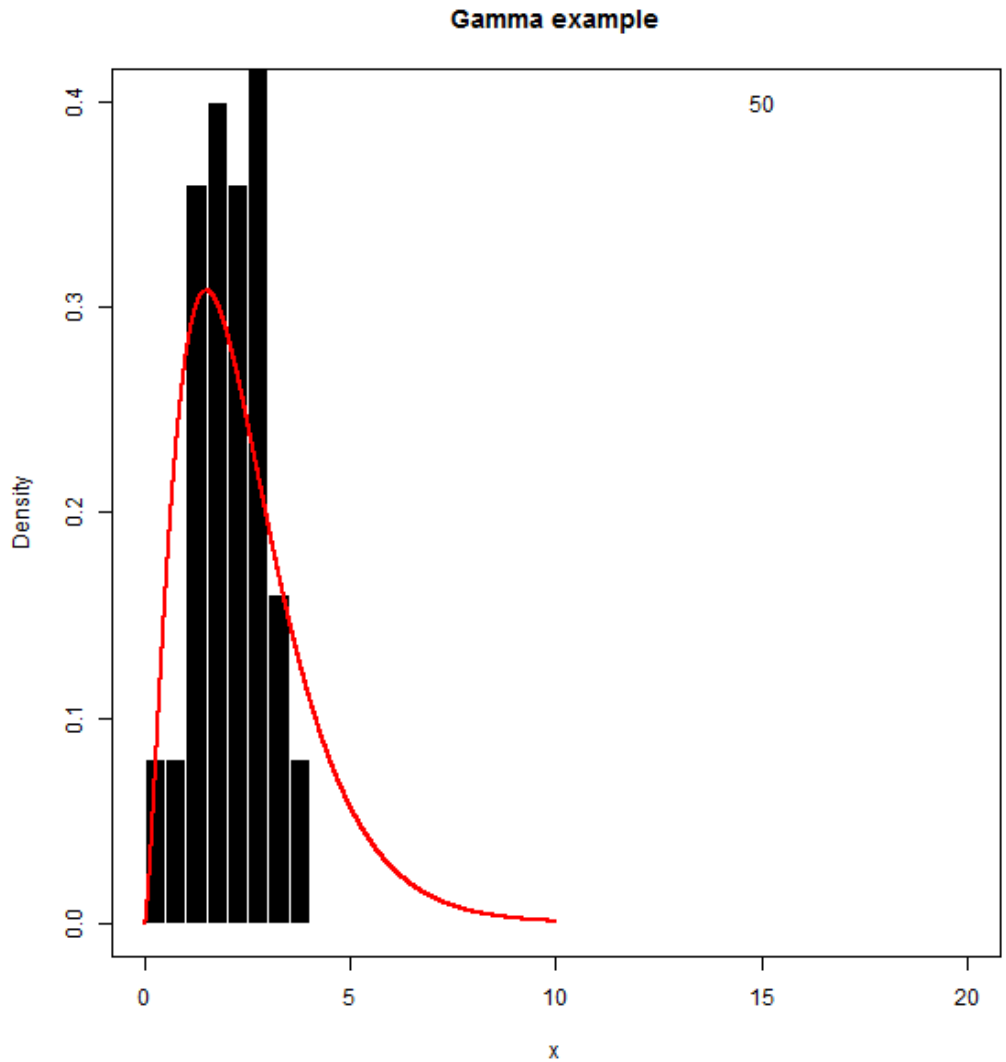
Metropolis-Hastings algorithm

Gamma density

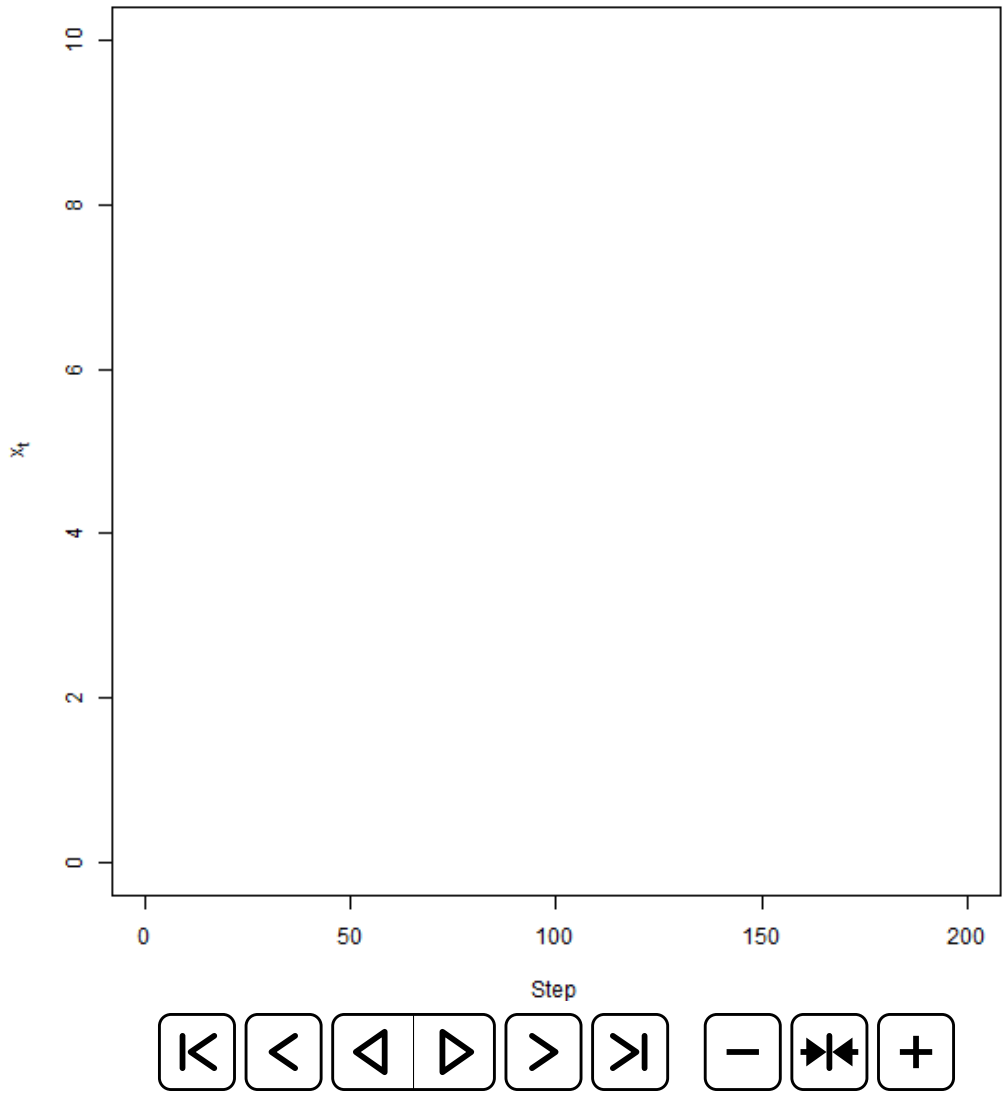
Histogram of states visited over $N = 10000$ steps (with Gamma density (solid))



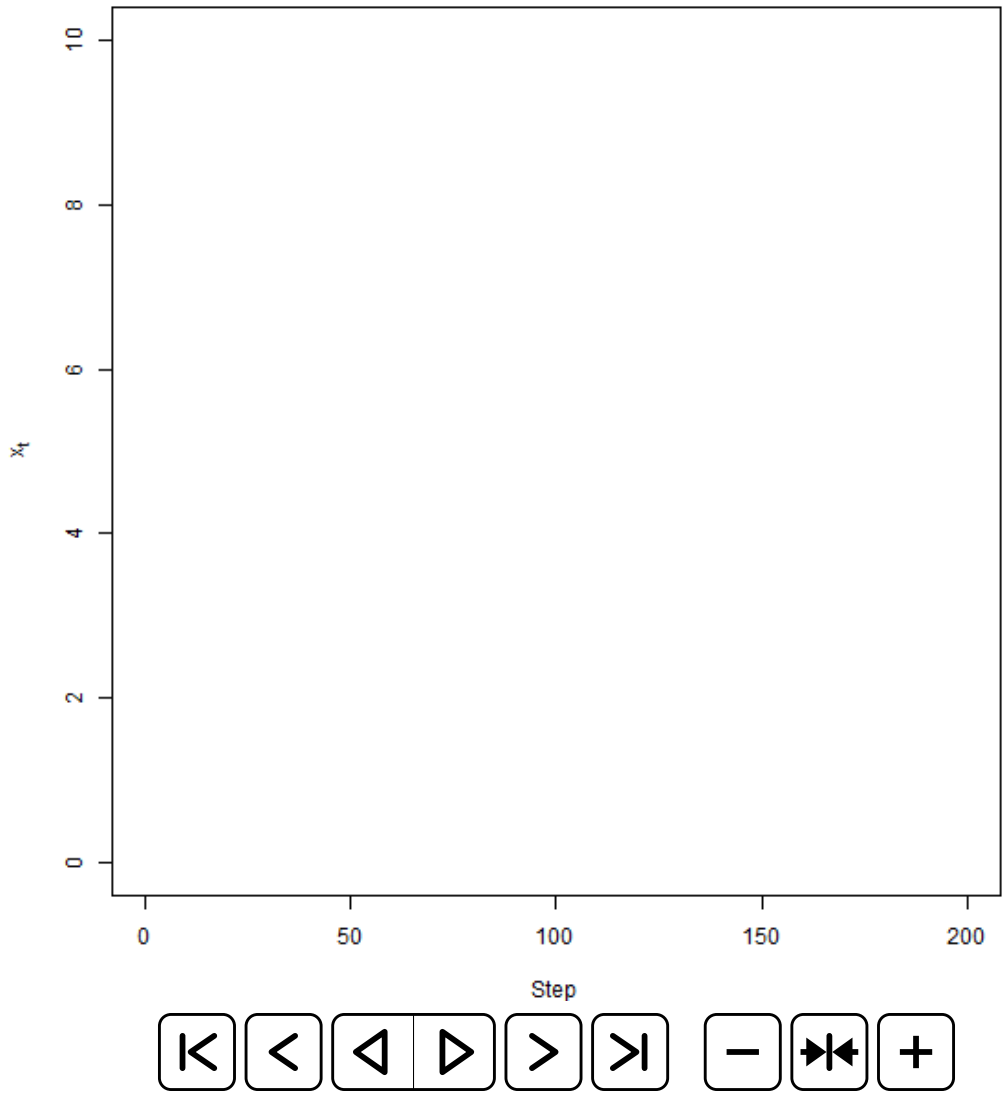
Gamma example: histogram over 10000 steps



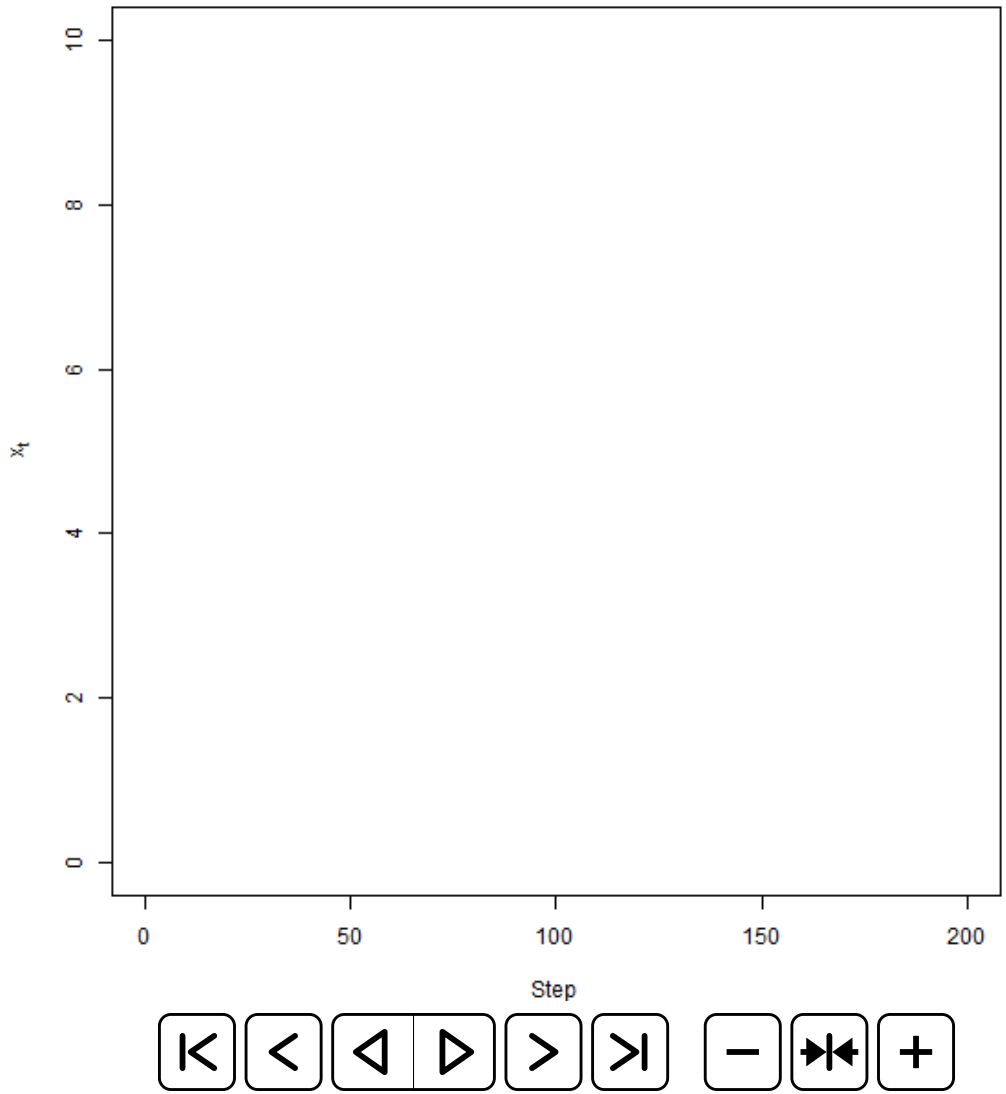
Gamma example: 200 steps, $x_0 = 0$, $\sigma_q = 1$.



Gamma example: 200 steps, $x_0 = 6$, $\sigma_q = 1$.



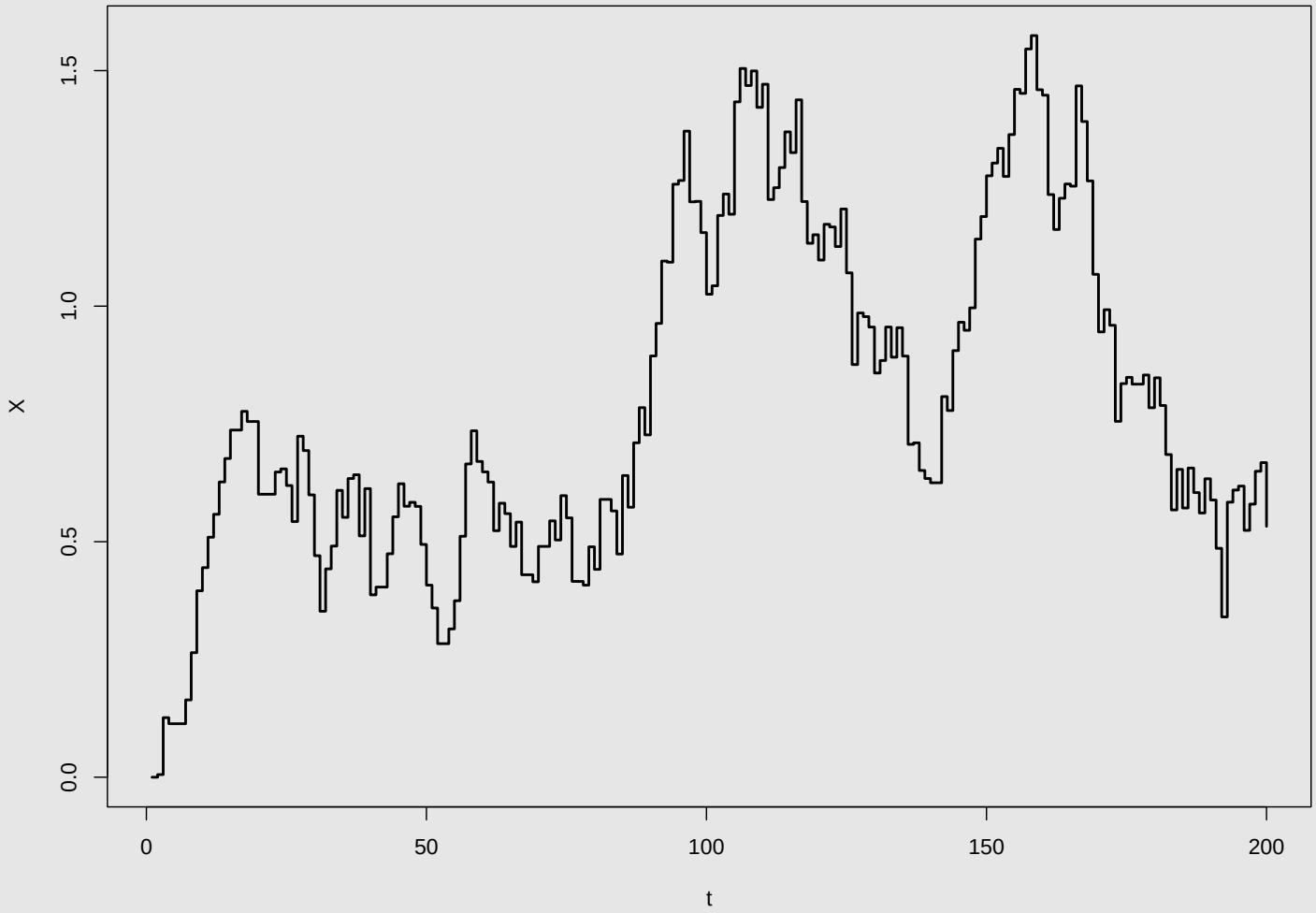
Gamma example: 200 steps, $x_0 = 10$, $\sigma_q = 0.25$.



Metropolis-Hastings algorithm

Gamma density

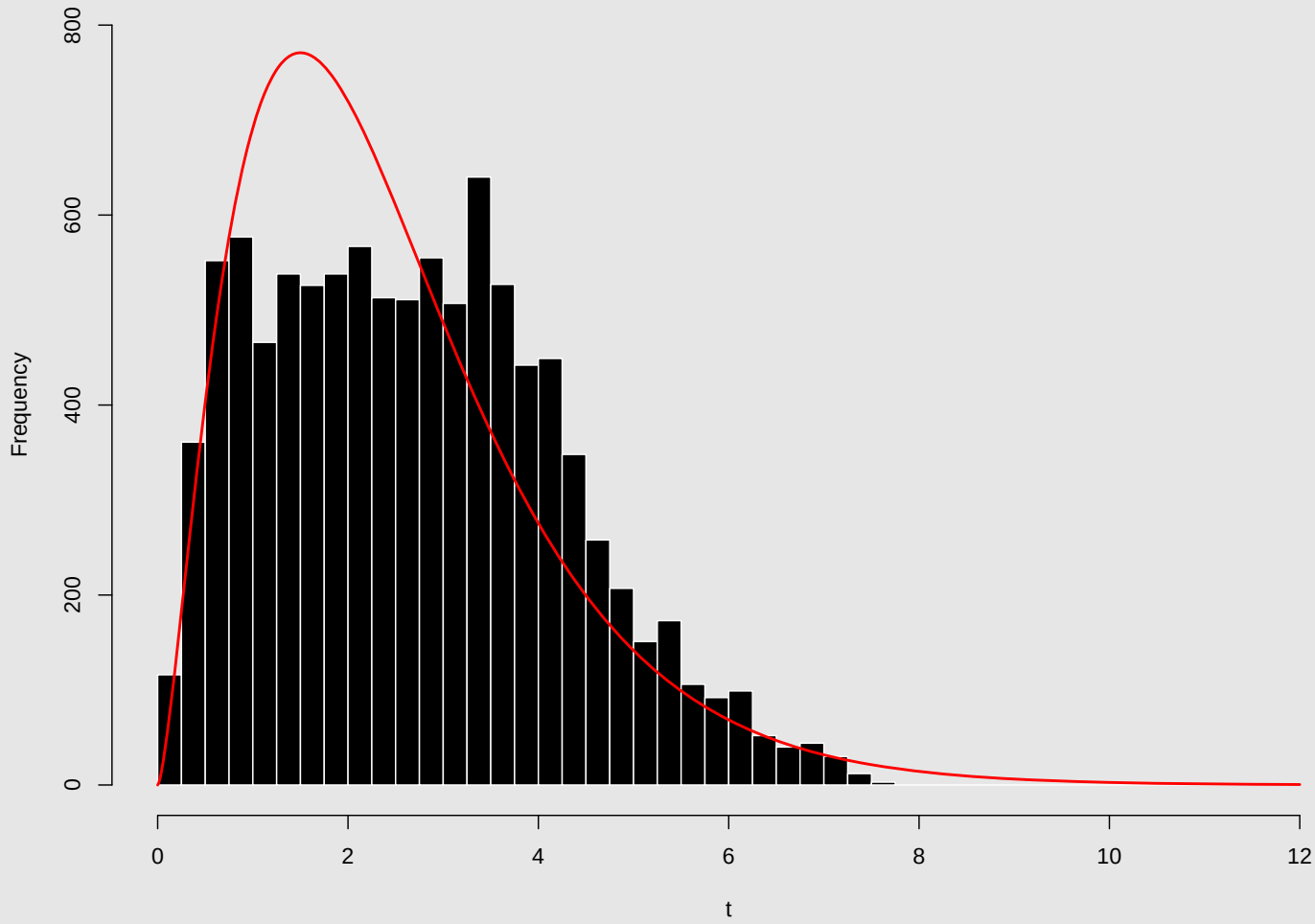
Chain starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 0.1$.



Metropolis-Hastings algorithm

Gamma density

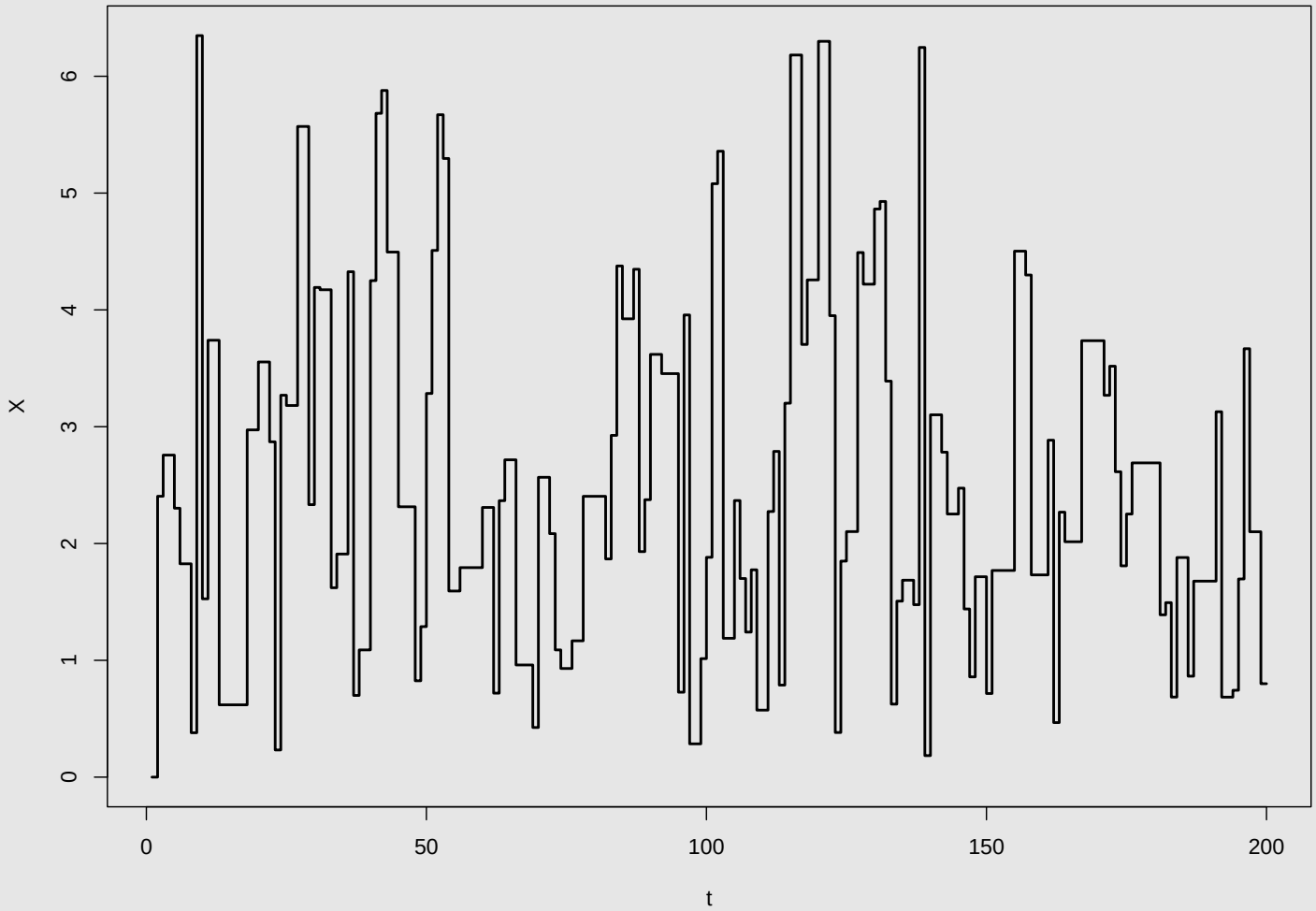
Histogram of states visited over $N = 10000$ steps.



Metropolis-Hastings algorithm

Gamma density

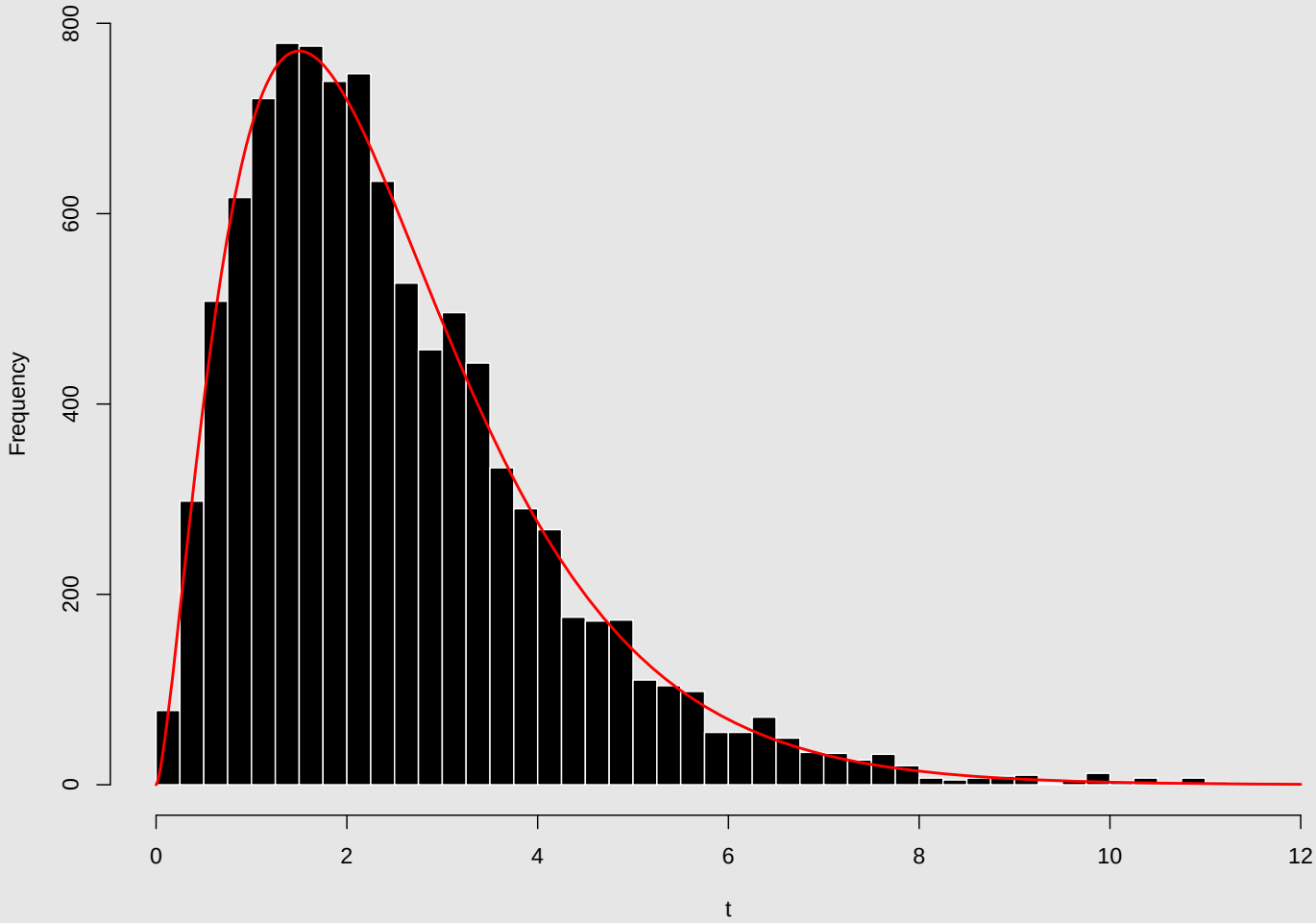
First 200 steps of the chain starting at $X_0 = 0$ with $\gamma = 2.5$, $\sigma_q = 3$.



Metropolis-Hastings algorithm

Gamma density

Histogram of states visited over $N = 10000$ steps (with Gamma density (solid))



Metropolis-Hastings algorithm

The states generated by the Markov chain are correlated; we can assess the performance of the Markov chain by examining the sample autocorrelation function

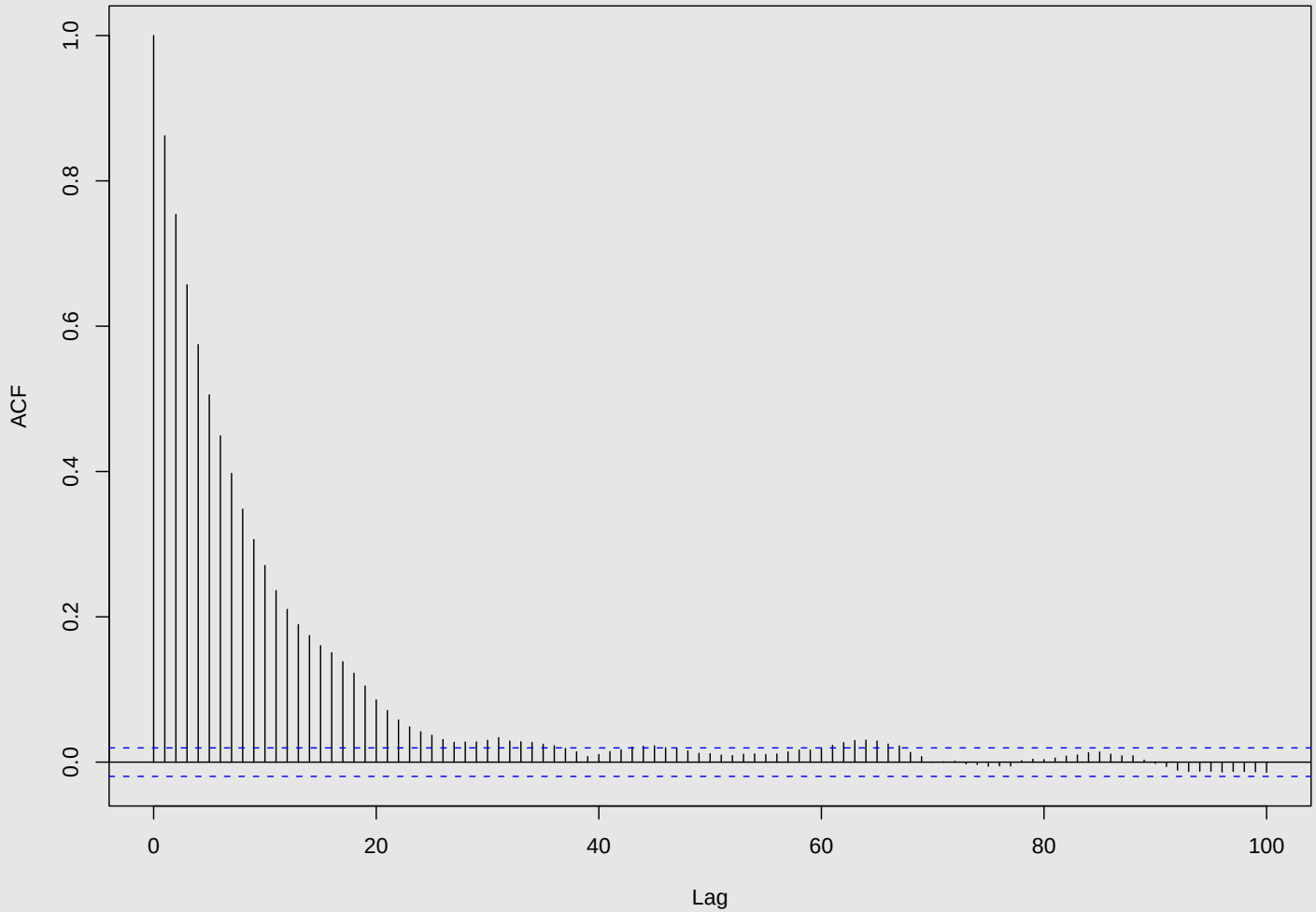
$$r(k) = \left(\frac{N-1}{N-k-1} \right) \frac{\sum_{t=k+1}^{N} (x_t - \overline{x})(x_{t-k} - \overline{x})}{\sum_{t=1}^{N} (x_t - \overline{x})^2}$$

for $k = 0, 1, 2 \ldots$.

A chain with high autocorrelation for large k is typically slow to converge.

Gamma density

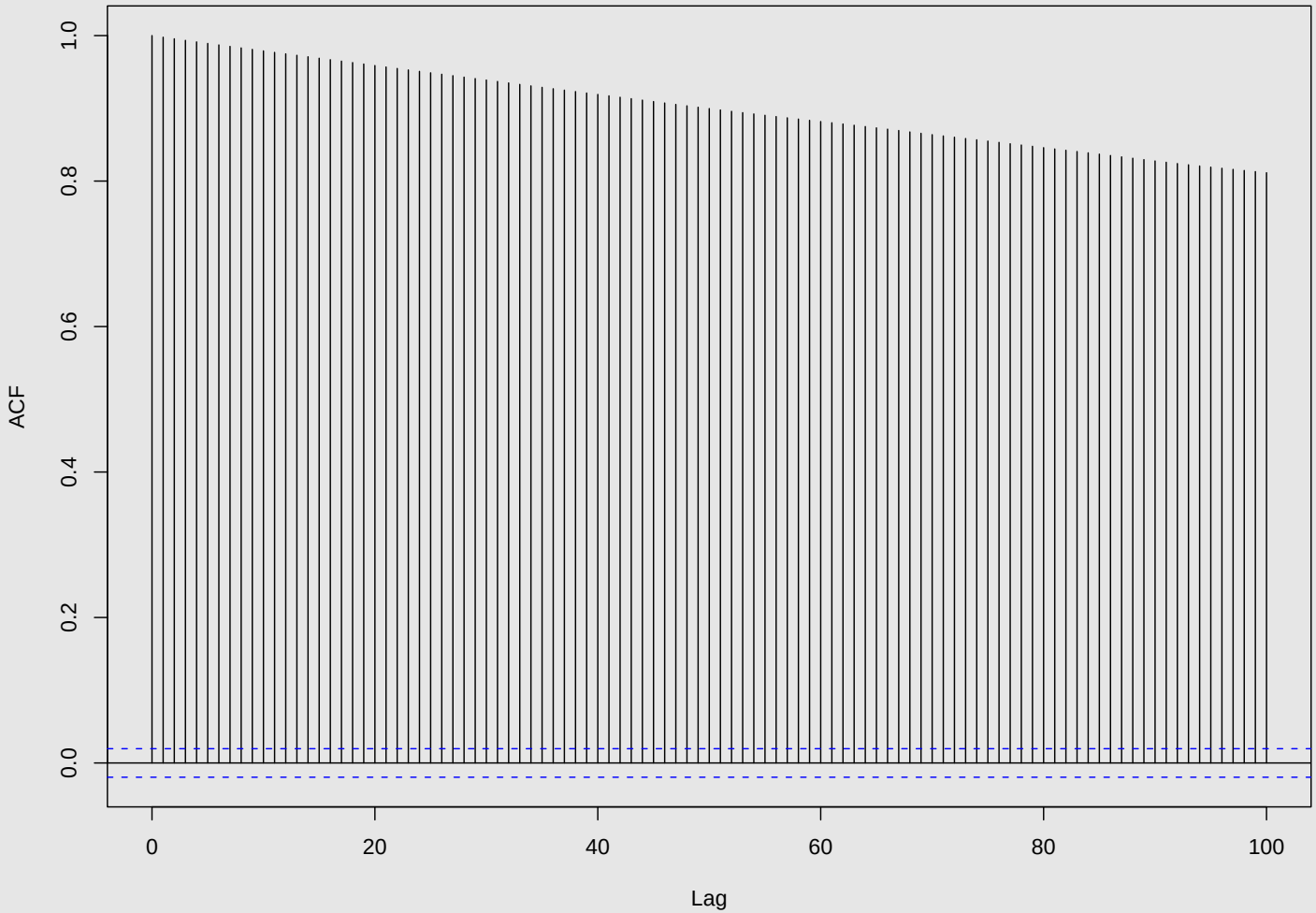
Autocorrelation function for $\sigma_q = 1$.



Metropolis-Hastings algorithm

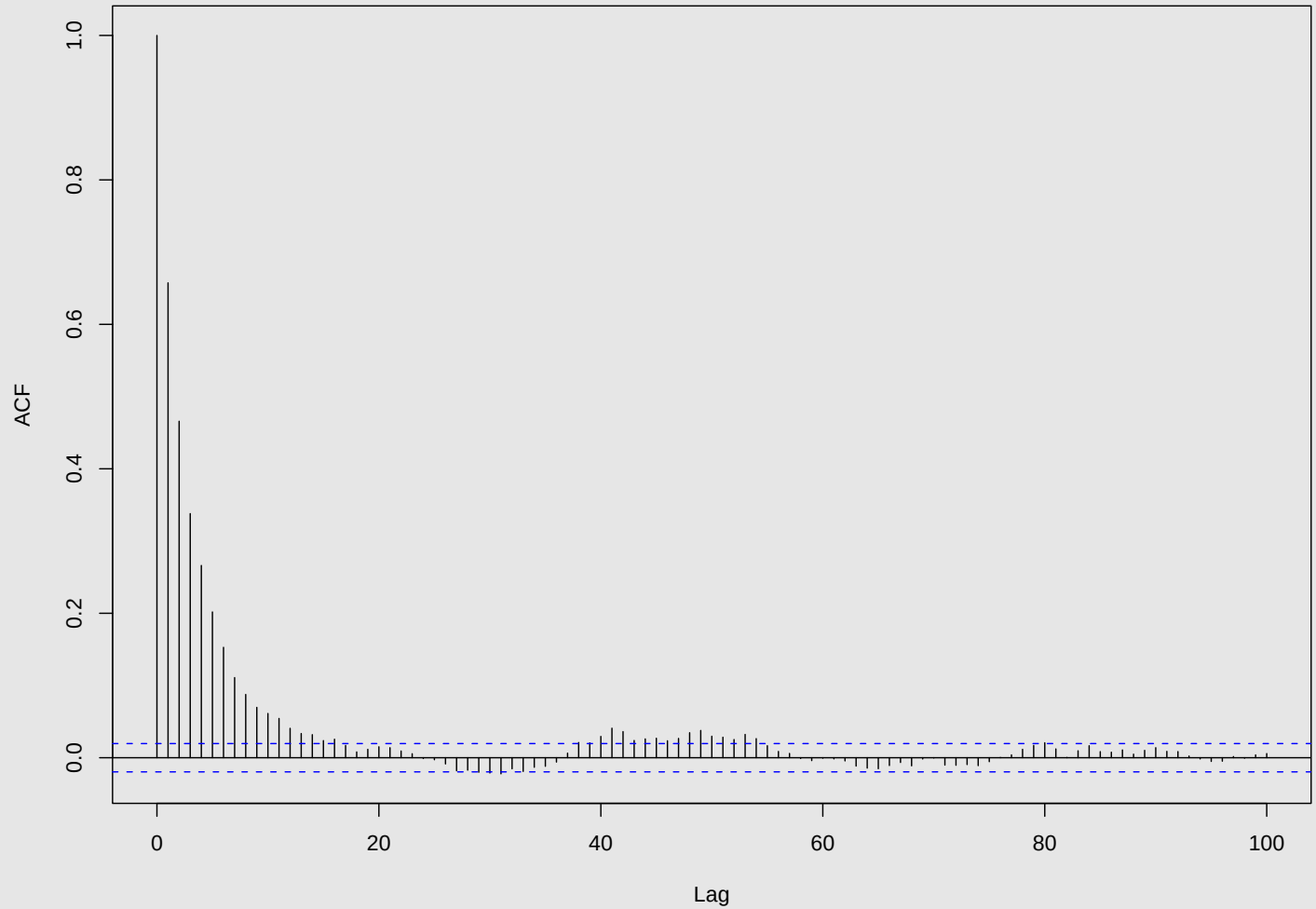
Gamma density

Autocorrelation function for $\sigma_q = 0.1$: inferior performance.



Gamma density

Autocorrelation function for $\sigma_q = 3$: superior performance.



Metropolis-Hastings algorithm

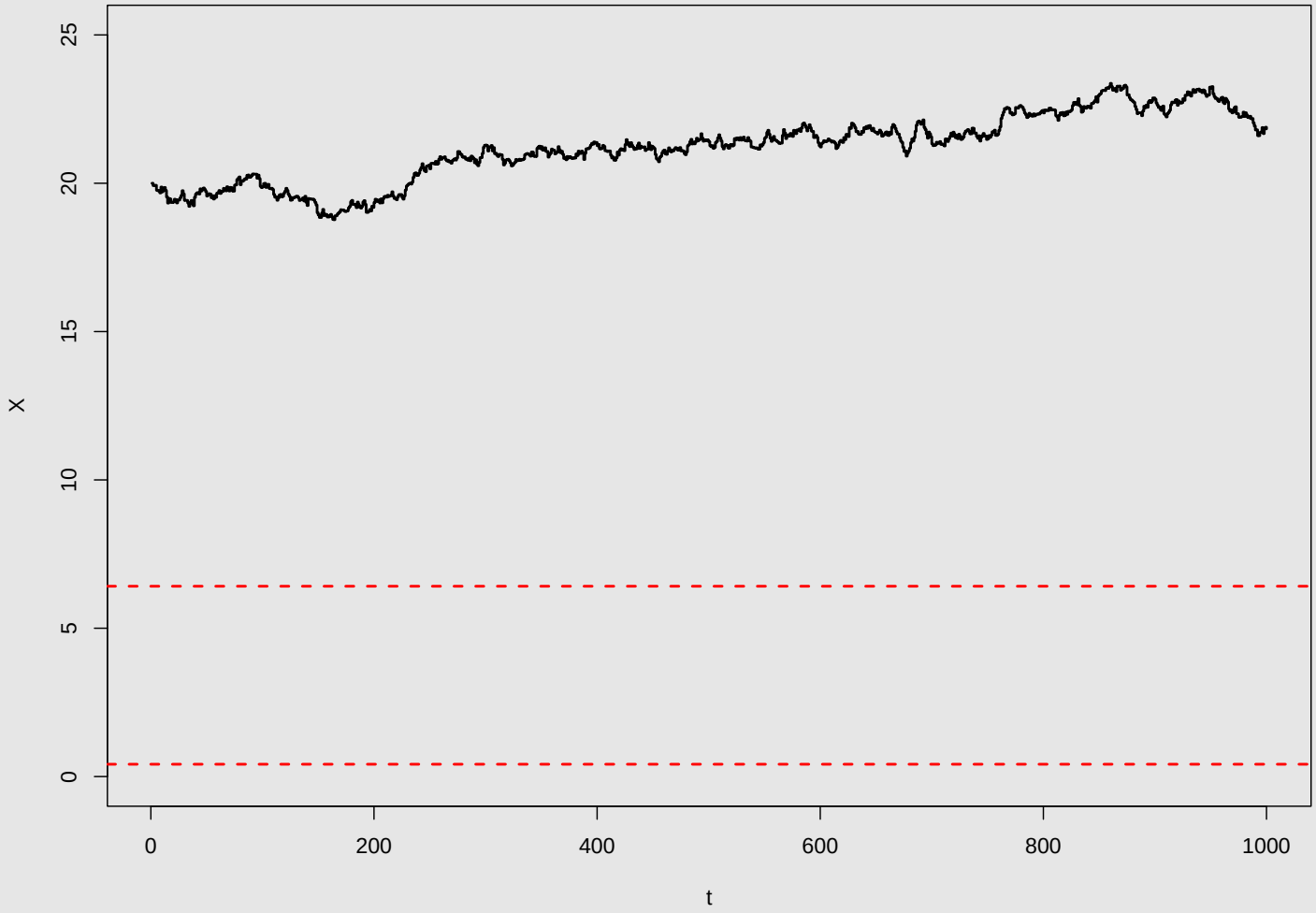
If the chain is started away from the high-probability region of π, then the it can take many steps to return there.

In the following trace plots, the red dashed lines give the 0.025 and 0.975 quantiles of the $Gamma(2.5, 1)$ distribution from the example. The chain is initialized at $X_0 = 20$, and then run for 20000 steps.

Metropolis-Hastings algorithm

Gamma density

Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 1000 steps.



Metropolis-Hastings algorithm

Gamma density

Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 10000 steps.



Metropolis-Hastings algorithm

Gamma density

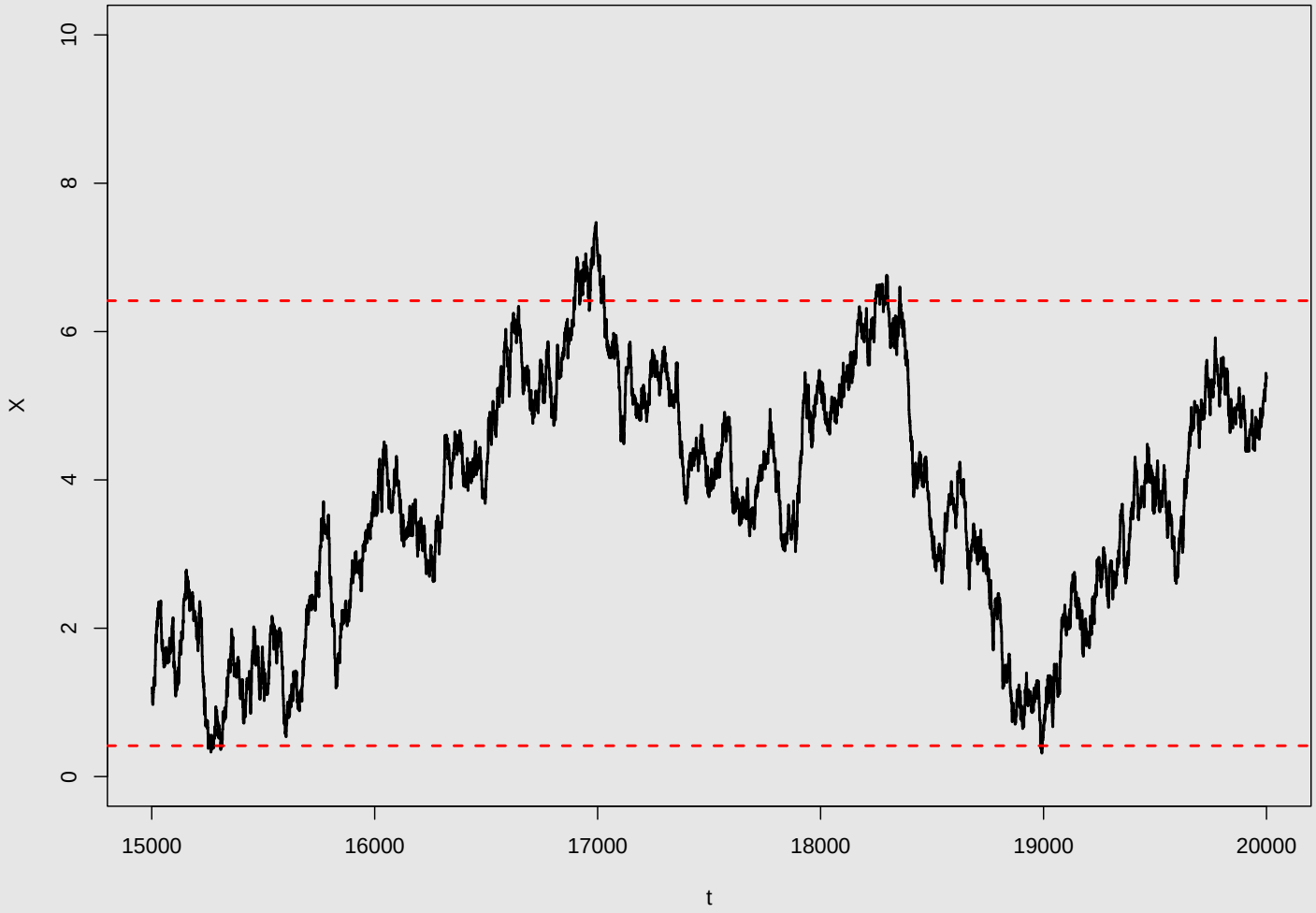
Starting value $X_0 = 20$, $\sigma_q = 0.1$: first 20000 steps.



Metropolis-Hastings algorithm

Gamma density

Starting value $X_0 = 20$, $\sigma_q = 0.1$: steps 15000 to 20000.



Metropolis-Hastings algorithm

In this final section of the chain, the generated values appear to oscillate, despite the fact that the chain has apparently reached the stationary phase. This oscillation is a result of the high autocorrelation present in the chain.

It is often difficult to distinguish such high autocorrelation from the case where a chain has not converged.

The high autocorrelation results here from the choice $\sigma_q = 0.1$; this value is smaller than is optimal.

Gibbs Sampler

The Metropolis-Hastings algorithm above is valid for both univariate and multivariate probability distributions, but is more complicated in high dimensions.

The objective is to choose a transition density q that moves around the space $\mathbb{X}$ quickly, which means that we wish to have the acceptance probability reasonably large.

In high dimensions, this is often difficult to achieve. The *Gibbs Sampler* algorithm attempts to solve this problem by breaking down a high-dimensional problem into several lower dimensional problems that are solved iteratively and simultaneously.

Gibbs Sampler

Suppose that π is a probability density in K dimensions, and let the variables be denoted $(X_1, ..., X_K)$. Define the conditional density $\pi_k(.|.)$ for

$$X_k|X_1, ..., X_{k-1}, X_{k+1}, ...X_K$$

by

$$\pi_k\left(x_k; x_{(k)}\right) = \frac{\pi\left(x_1, ..., x_K\right)}{\pi\left(x_1, ..., x_{k-1}, x_{k+1}, ...x_K\right)} \propto \pi\left(x_1, ..., x_K\right)$$

where the denominator is the marginal distribution of $X_{(k)}$, the $K-1$ variables excluding X_k.

The Gibbs Sampler utilizes this set of K *full conditional distributions* to construct a Markov chain on the support of the joint distribution.

Gibbs Sampler

It is implemented using the following algorithm:

1. Set a vector of starting values for the K variables $(x_{10}, ..., x_{K0})$.
2. Sample in turn from the conditional distributions as follows:

   (i) sample x_{11} from $\pi_1 (x_1; x_{20}, x_{30}, ..., x_{K0})$
   (ii) sample x_{21} from $\pi_2 (x_2; x_{11}, x_{30}, ..., x_{K0})$
   (iii) sample x_{31} from $\pi_3 (x_3; x_{11}, x_{21}, ..., x_{K0})$
   (iv) $\vdots$
   (v) sample x_{K1} from $\pi_K (x_K; x_{11}, x_{21}, ..., x_{K-1\,1})$

   This completes one step of the Gibbs sampler.
3. Return to 2 (i), and repeat to obtain, at step t, the sampled variates $(x_{1t}, x_{2t}, ..., x_{Kt})$

Gibbs Sampler

This Markov chain defines a means of updating the vector x_t to vector x_{t+1}. Each of the steps can be achieved using direct sampling from the conditional distribution if that is possible, but can also involve individual Metropolis-Hastings steps, with acceptance probabilities

$$\alpha_k(x, z) = \min\left\{1, \frac{\pi_k\left(z; x_{(k)}\right)}{\pi_k\left(x; x_{(k)}\right)} \frac{q_k(z, x)}{q_k(x, z)}\right\}$$

for $k = 1, ..., K$. In 2., the steps can also be completed in *random order.*

Gibbs Sampler as Metropolis-Hastings

The Gibbs sampler is in fact a special case of the Metropolis-Hastings algorithm: we can regard the individual updates in 2. as implementing K separate transition kernels $P_1, \ldots, P_K$ that act on the components of the vector $\mathbf{X}$; note that these kernels in isolation yield *reducible* Markov chains.

A more general form of Metropolis-Hastings algorithm is based on a *mixture* transition kernel

$$P(x, B) = \sum_j \omega_j P_j(x, B)$$

where $0 \leqslant \omega_j \leqslant 1$ and $\sum_j \omega_j = 1$, and the P_j are themselves transition kernels. This allows for the possibility of choosing several proposal densities q_j.

Gibbs Sampler: example

Bivariate Normal

Suppose $\mathbf{X} = (X_1, X_2)^\top \sim \mathcal{N}_2(\mathbf{0}, \Sigma)$ where

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

The general result for multivariate normal distribution conditional distributions is that if $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)^\top \sim \mathcal{N}_d(\mu, \Sigma)$, where $\mathbf{X}_1$ is $(d_1 \times 1)$, and

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

then

$$\mathbf{X}_1 | \mathbf{X}_2 = \mathbf{x}_2 \mathcal{N}_{d_1} \sim \left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

Gibbs Sampler: example

Bivariate Normal

Therefore, here $d = 2$, and

$$X_1|X_2 = x_2 \sim \mathcal{N}(\rho x_2, (1-\rho^2))$$

$$X_2|X_1 = x_1 \sim \mathcal{N}(\rho x_1, (1-\rho^2))$$

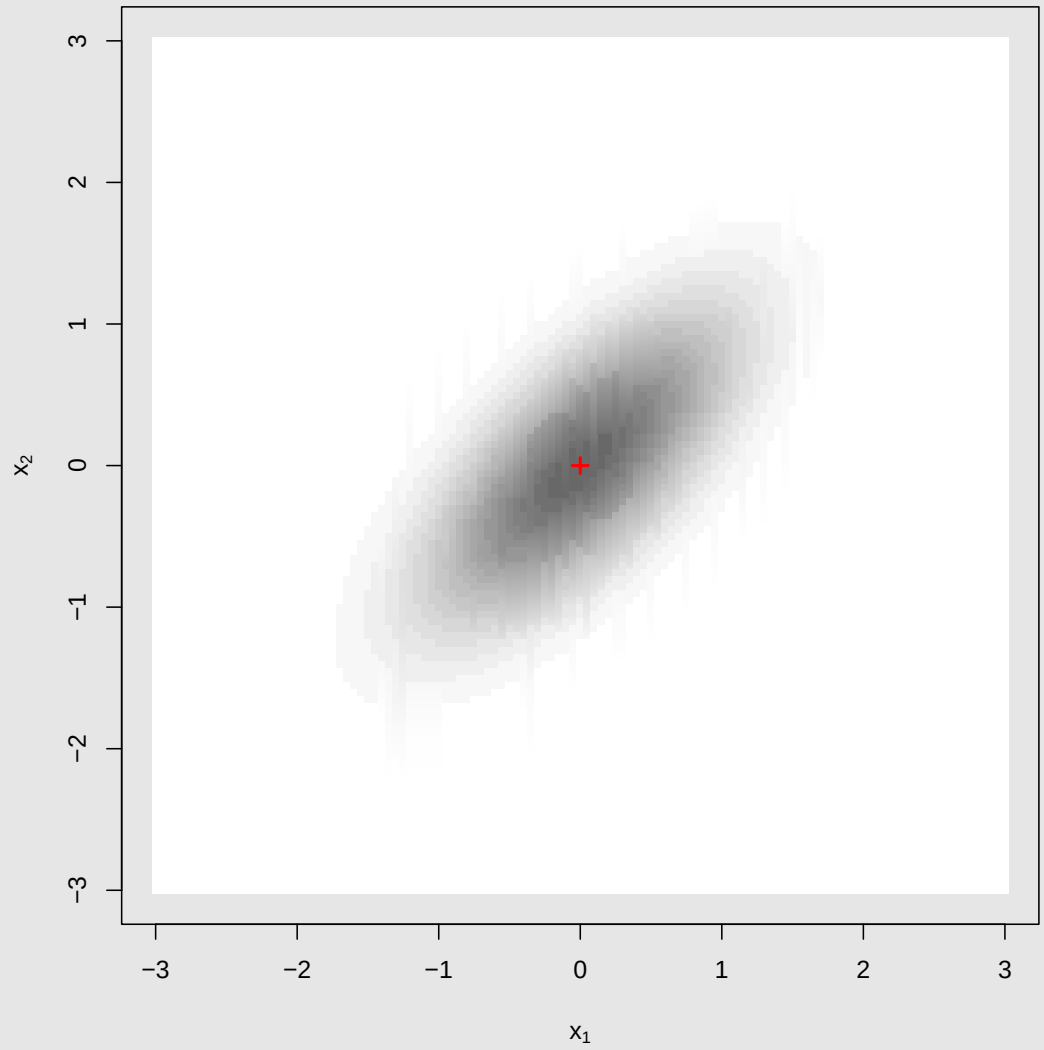
These distributions are sampled repeatedly with updating of the conditioning value after each sampling.

Suppose we start at $(x_{10}, x_{20}) = (0, 0)$.

Gibbs Sampler: example

Bivariate Normal

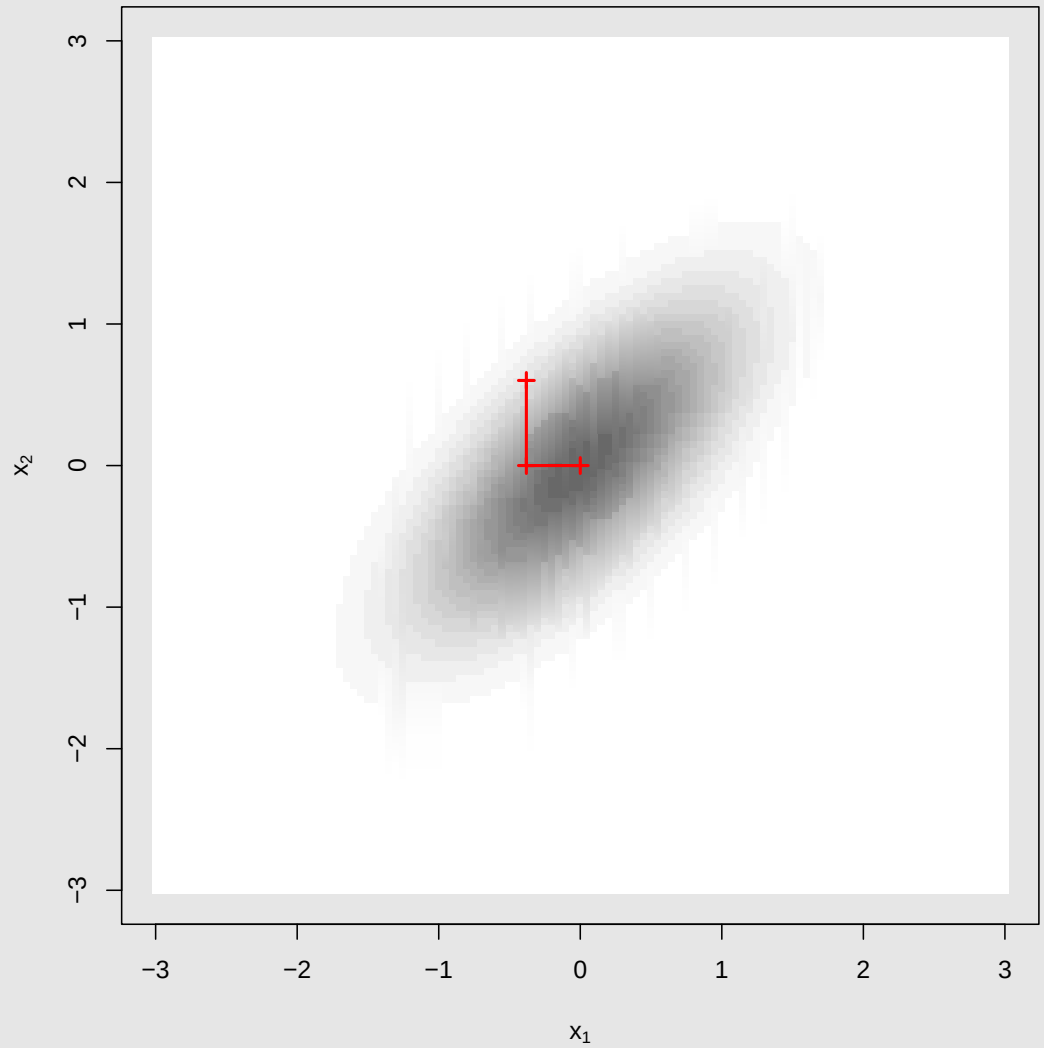
Initial point: (x_{10}, x_{20})



Gibbs Sampler: example

Bivariate Normal

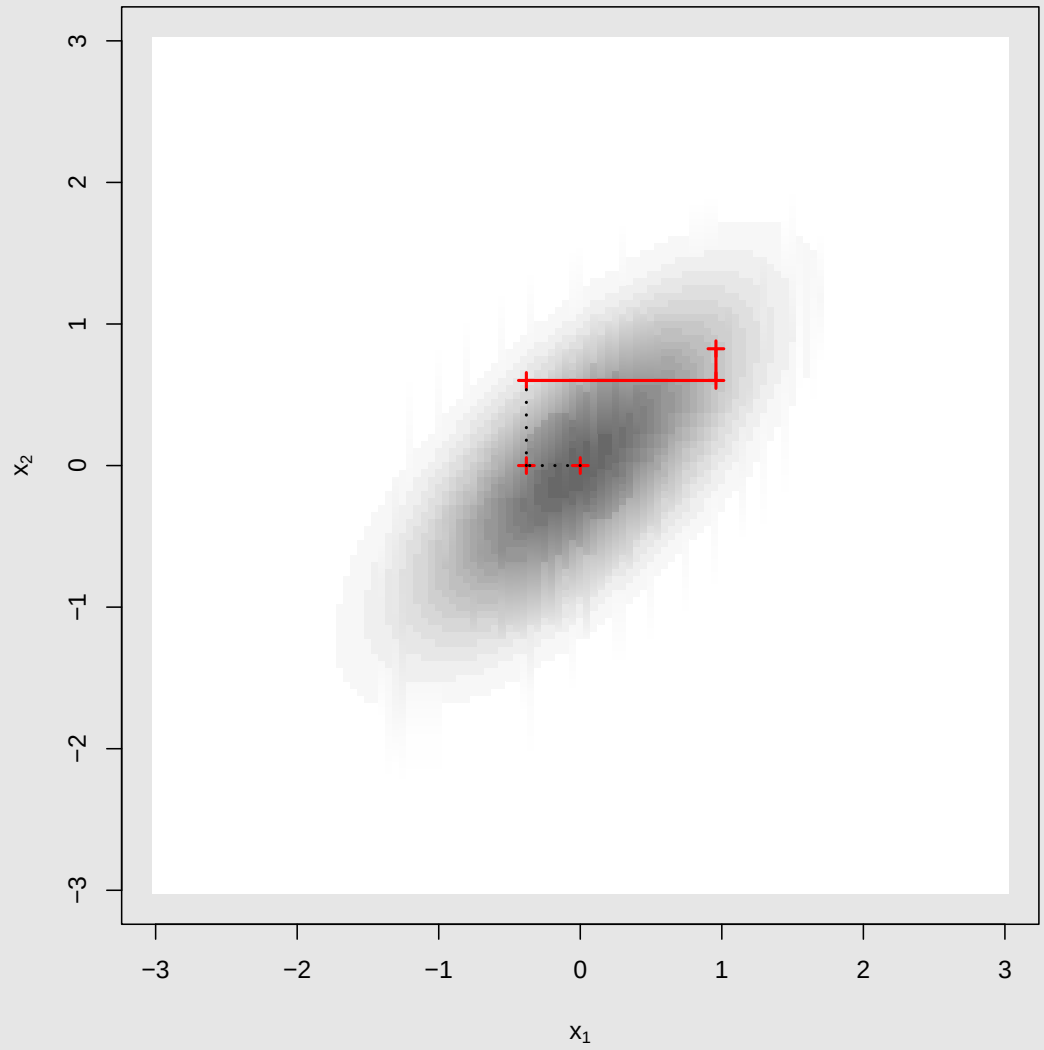
After one update: (x_{11}, x_{21})



Gibbs Sampler: example

Bivariate Normal

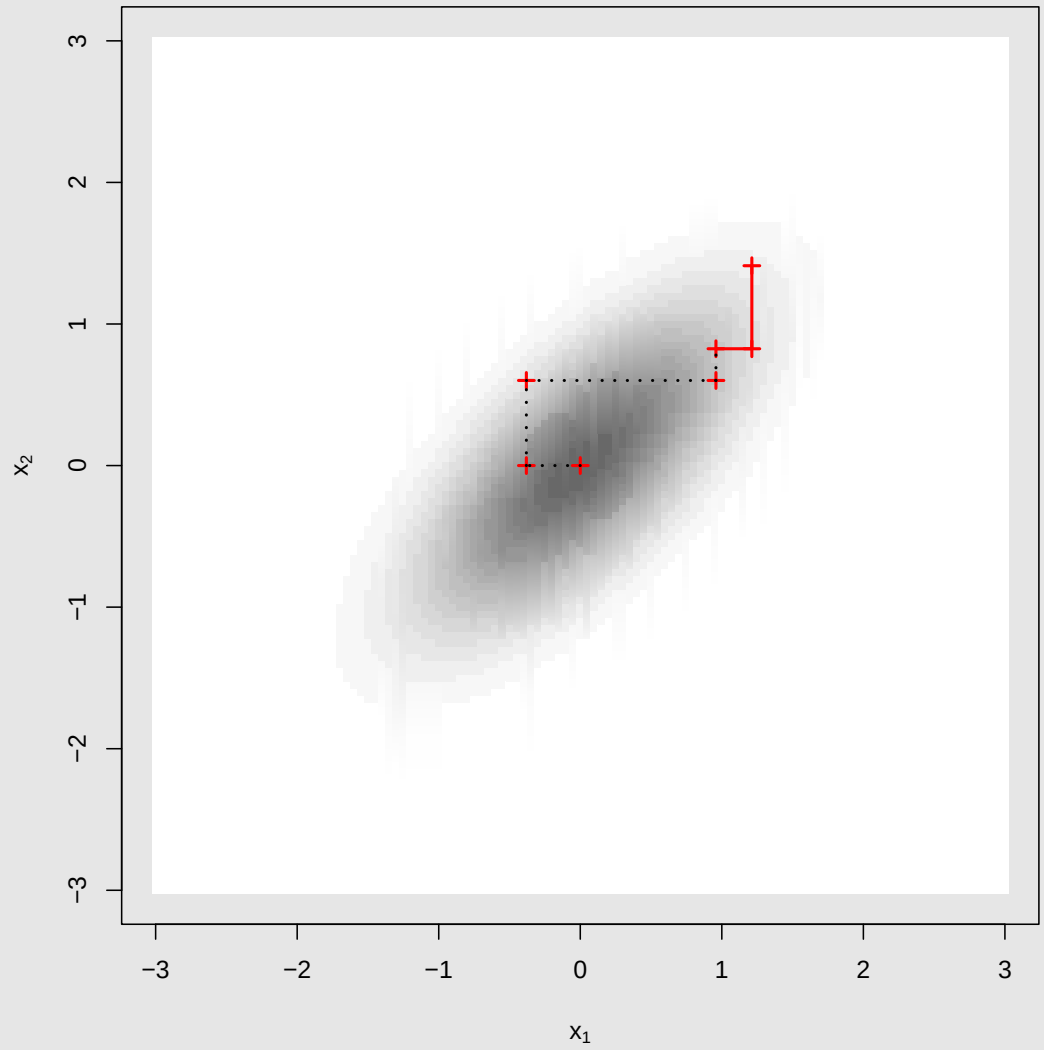
After two updates: (x_{12}, x_{22})



Gibbs Sampler: example

Bivariate Normal

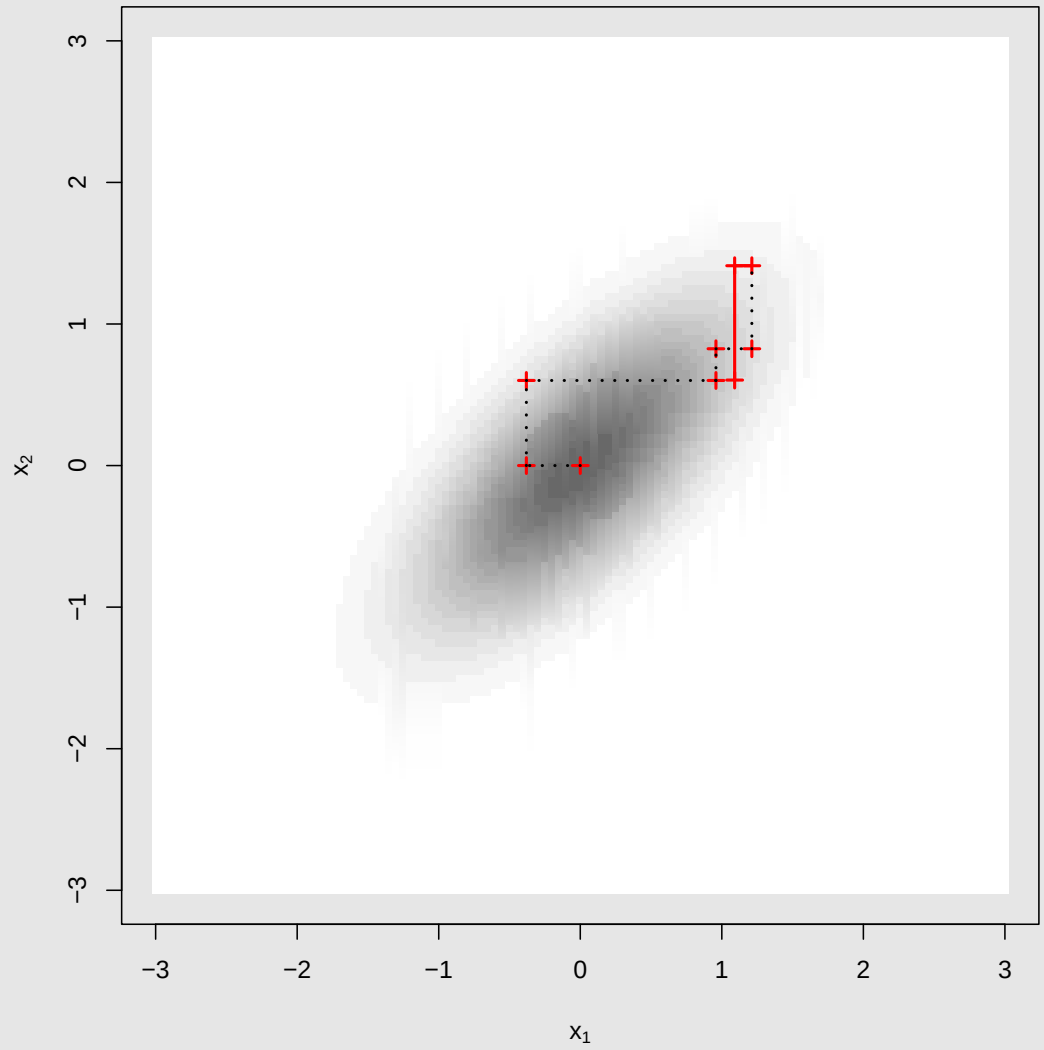
After three updates: (x_{13}, x_{23})



Gibbs Sampler: example

Bivariate Normal

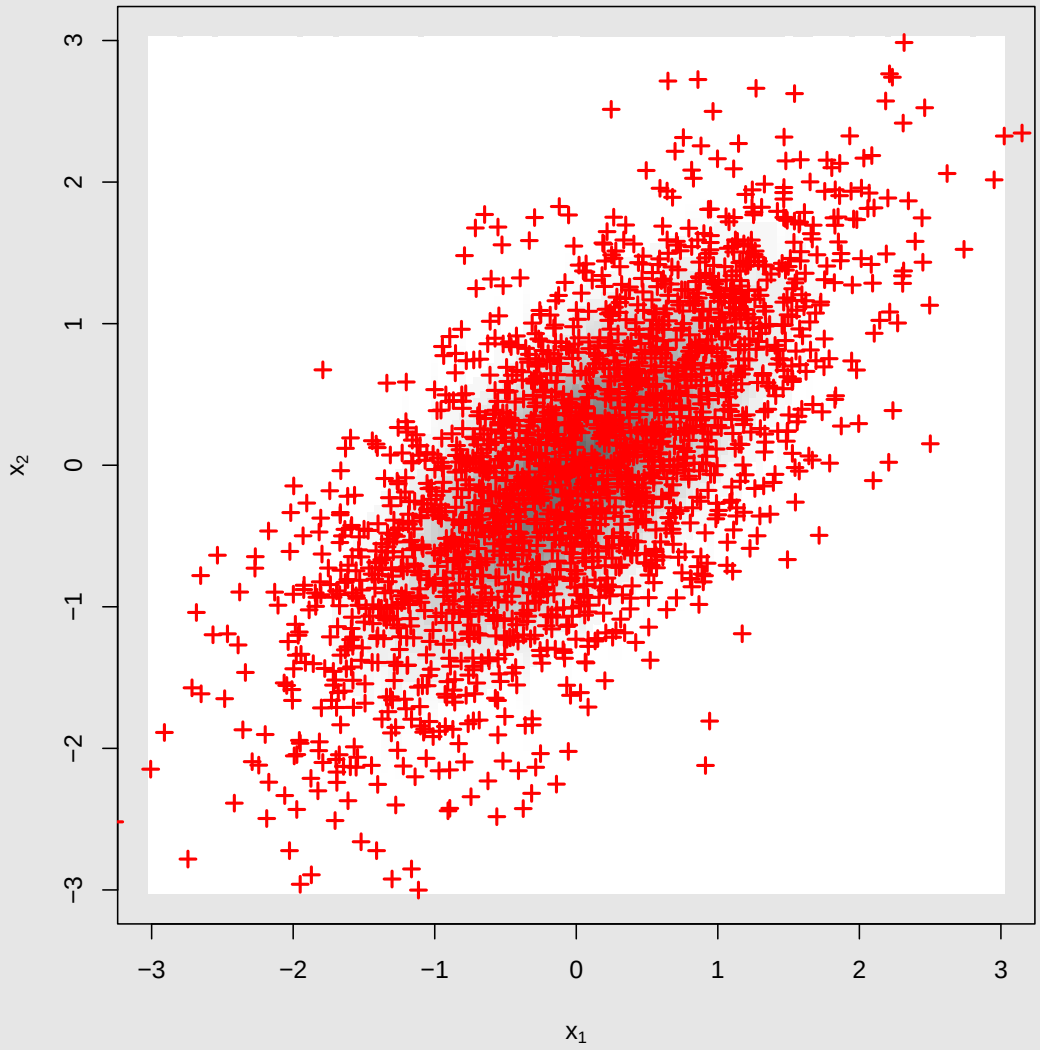
After four updates: (x_{14}, x_{24})



Gibbs Sampler: example

Bivariate Normal

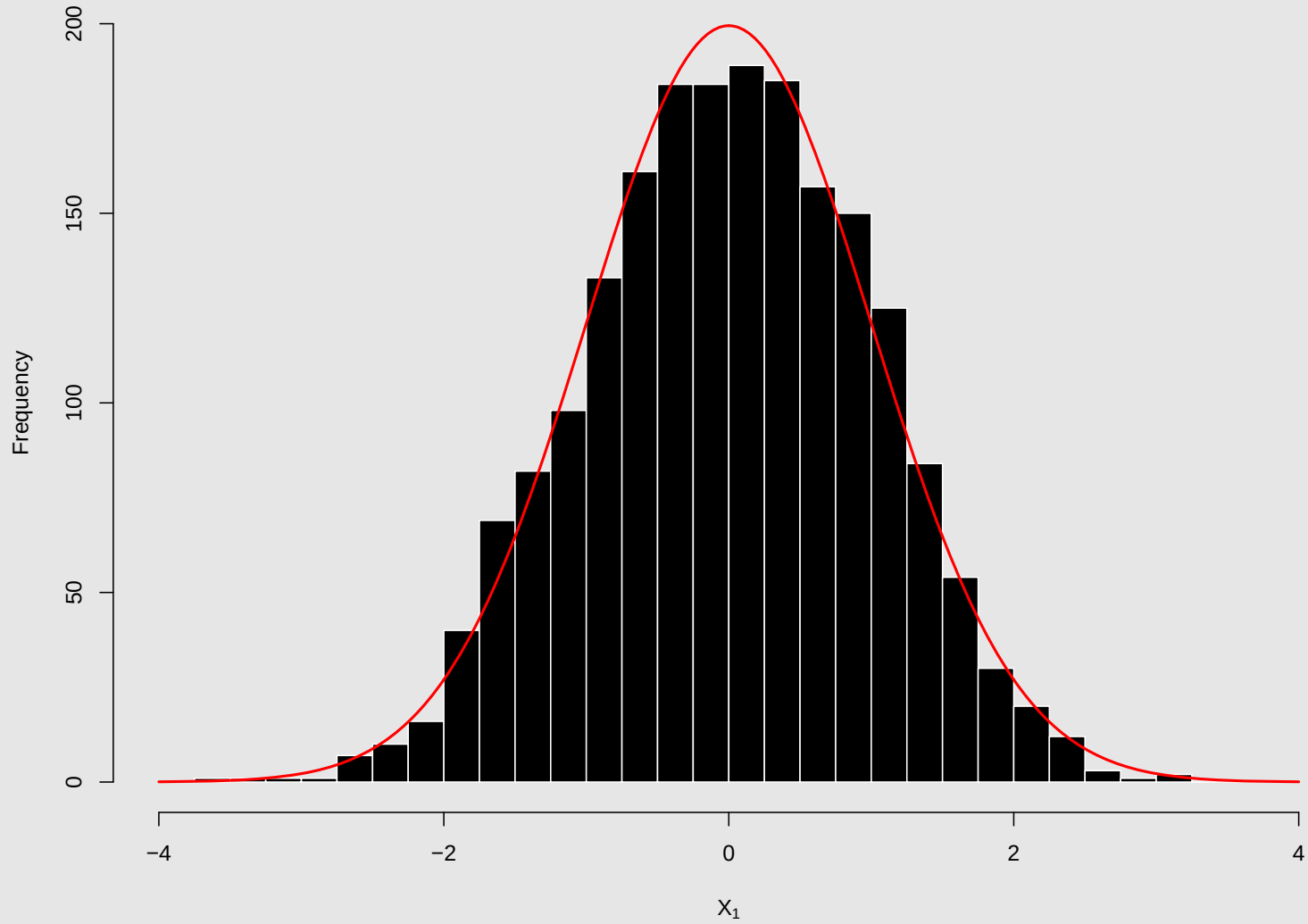
After 2000 updates: entire collected sample



Gibbs Sampler: example

Bivariate Normal

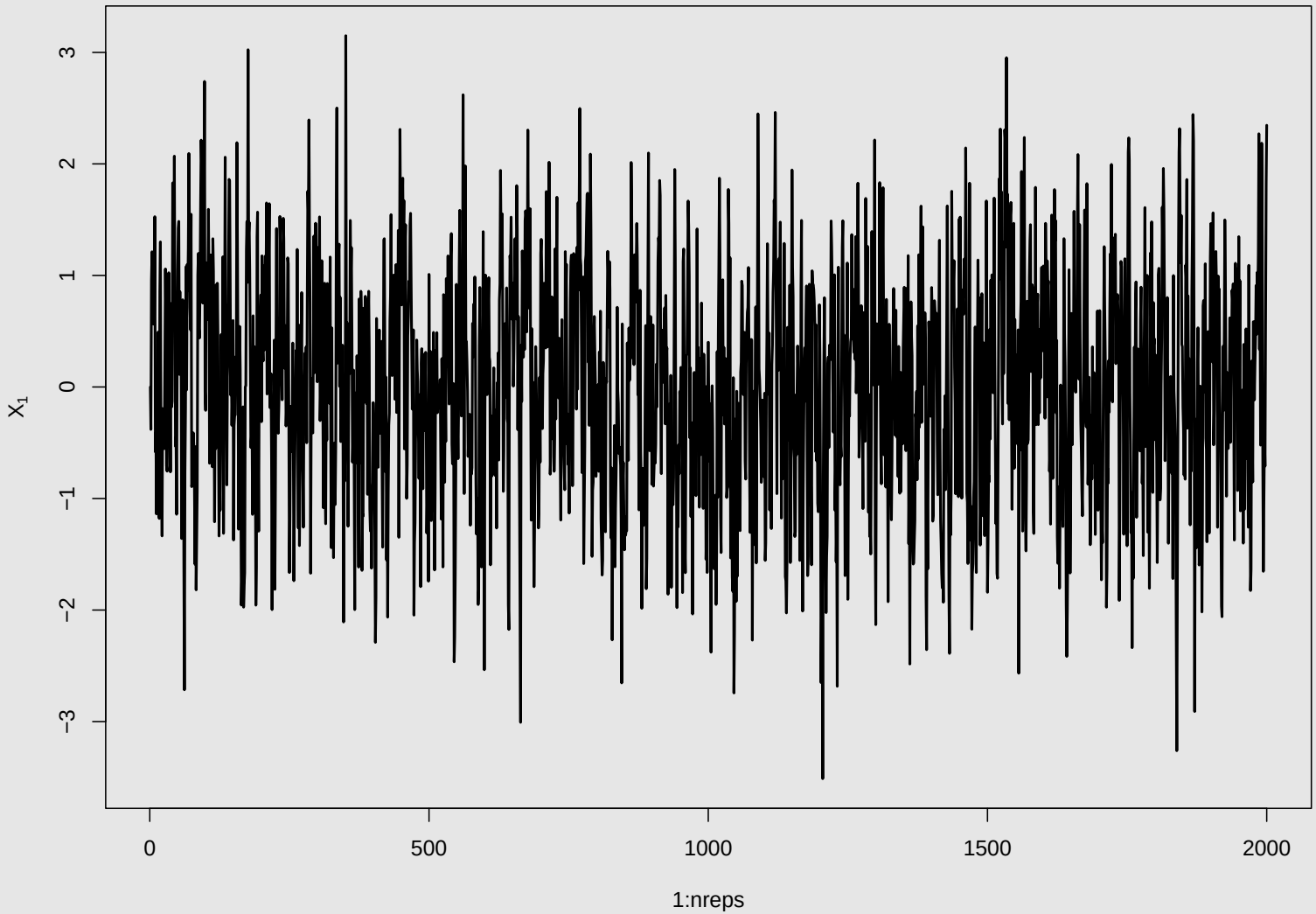
Histogram for X_1 with true marginal density (solid):



Gibbs Sampler: example

Bivariate Normal

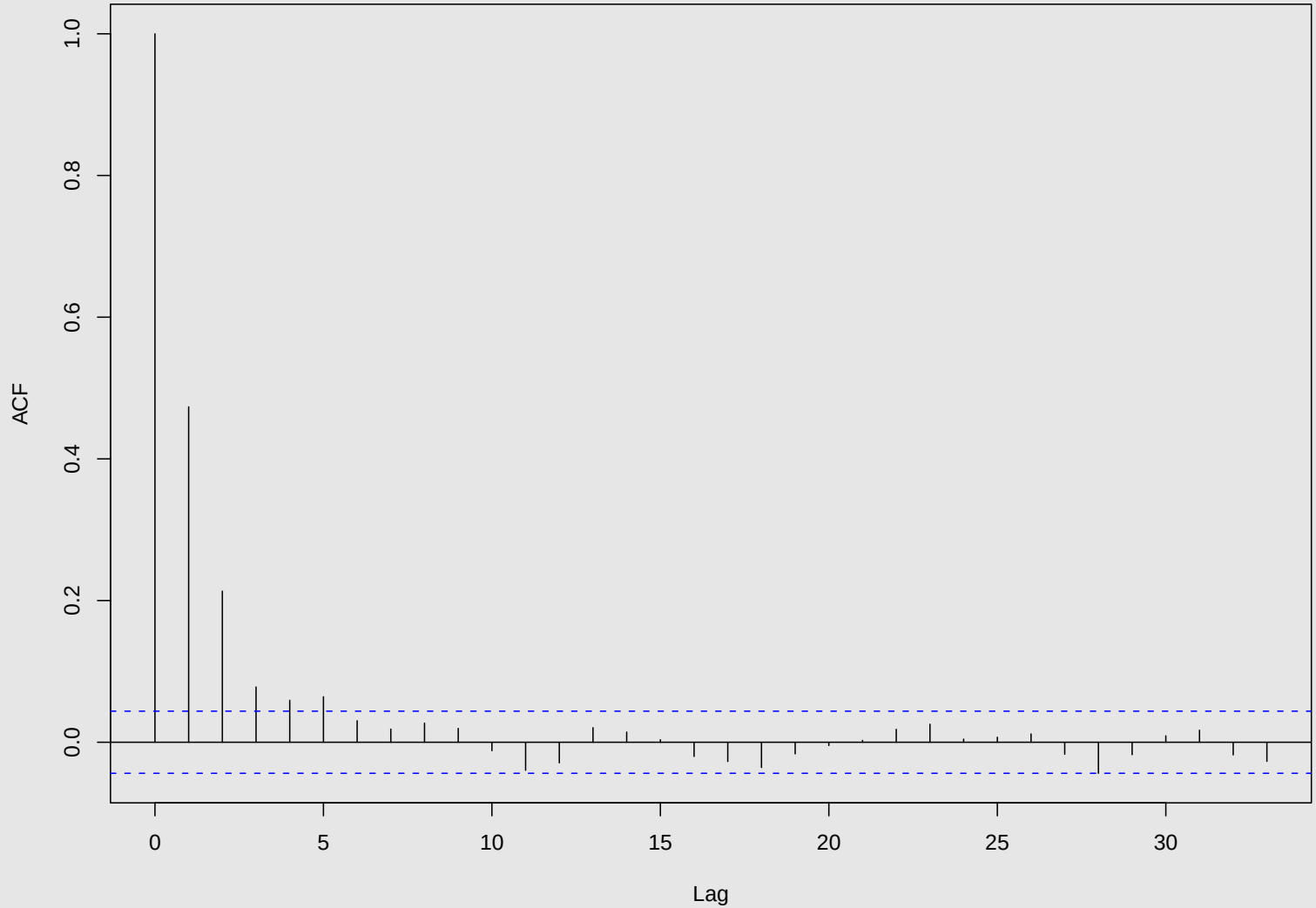
Trace for X_1:



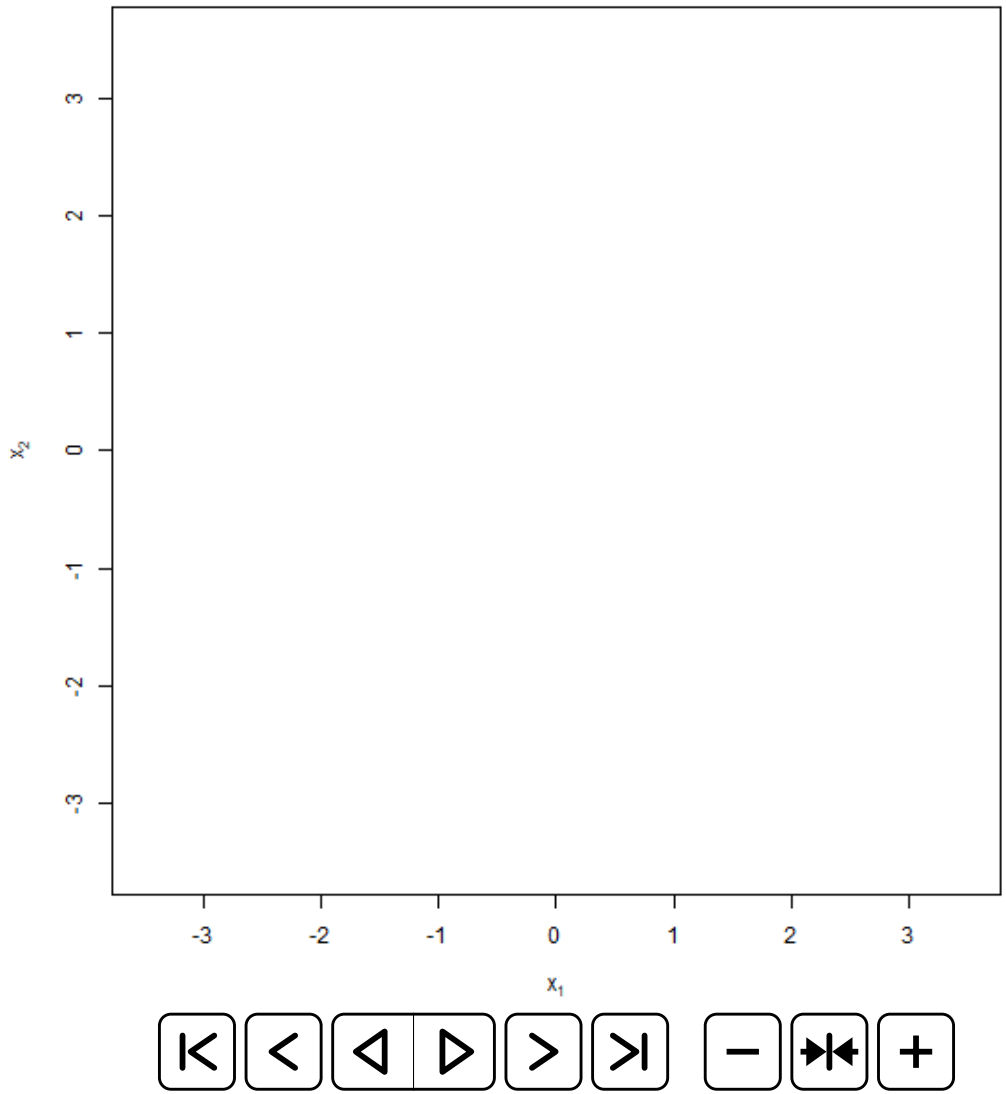
Gibbs Sampler: example

Bivariate Normal

ACF for X_1:



Gibbs Sampler: $\rho = 0.6$



Gibbs Sampler: example

Bivariate Normal

We note that

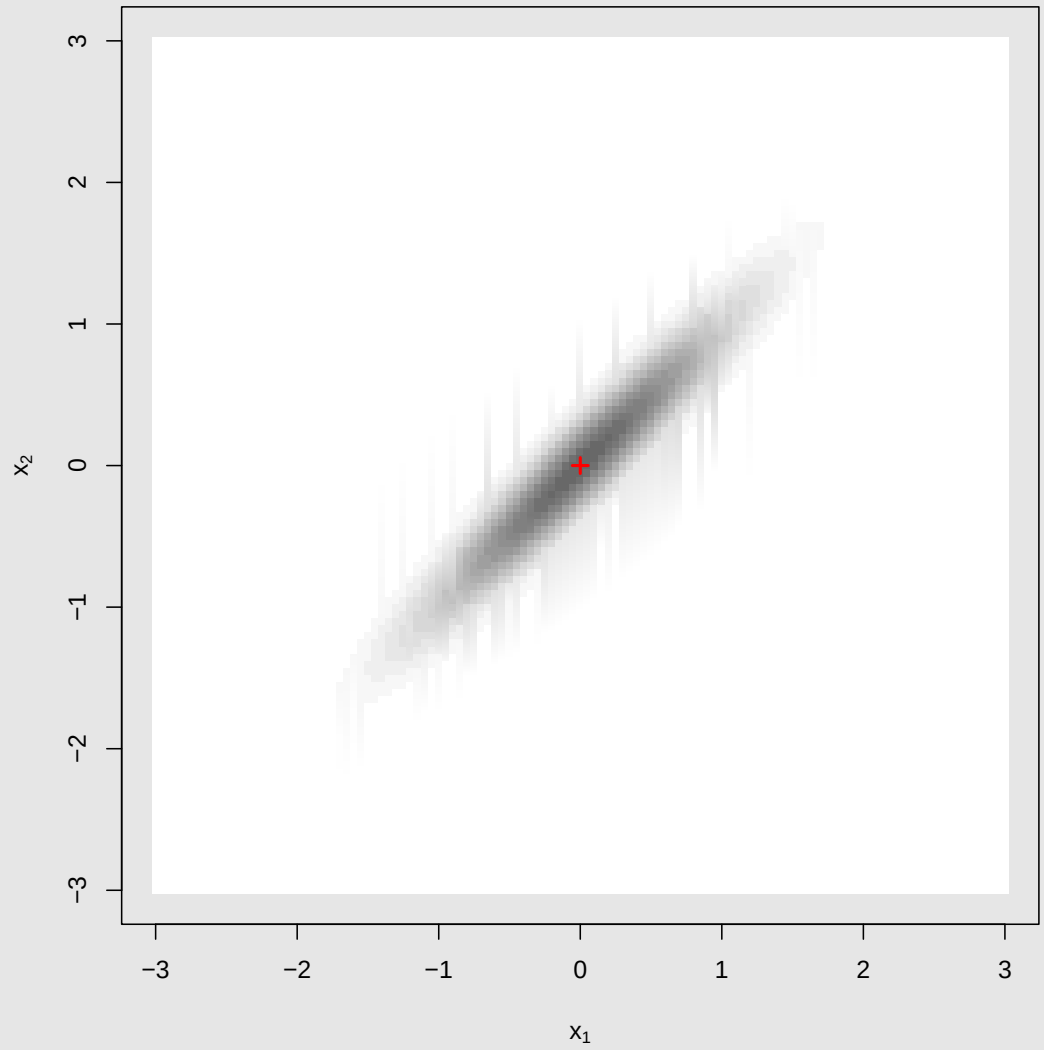
- the Gibbs sampler makes one-step moves along the coordinate axes
- the moves can traverse the support of the joint density fairly well
- there is no "tuning" of a proposal parameter (like σ_q)
- the samples of x_1 that are collected across steps are (dependent) samples from the correct marginal distribution for X_1; the same result holds for X_2

We can re-run the Gibbs sampler from the same starting value, but now with $\rho = 0.95$.

Gibbs Sampler: example

Bivariate Normal

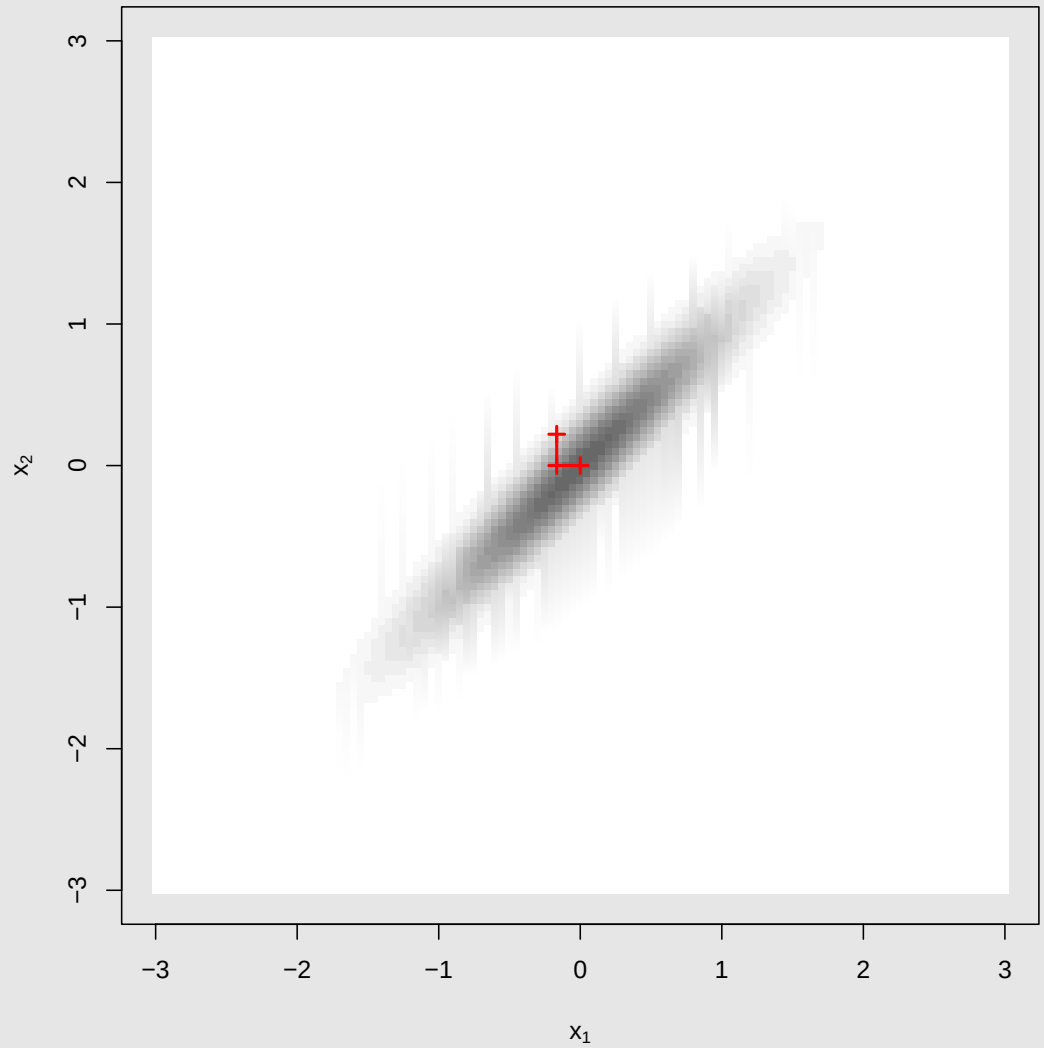
Initial point: (x_{10}, x_{20})



Gibbs Sampler: example

Bivariate Normal

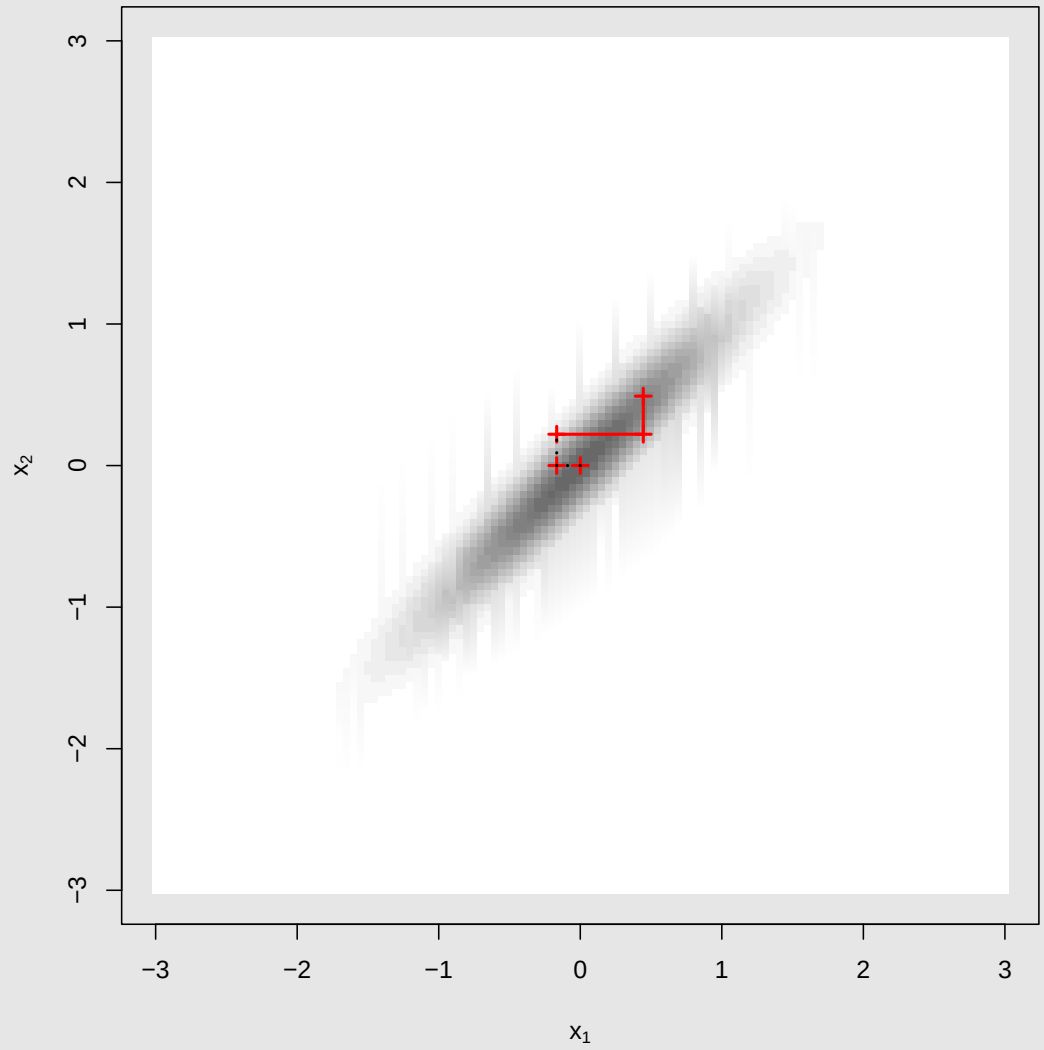
After one update: (x_{11}, x_{21})



Gibbs Sampler: example

Bivariate Normal

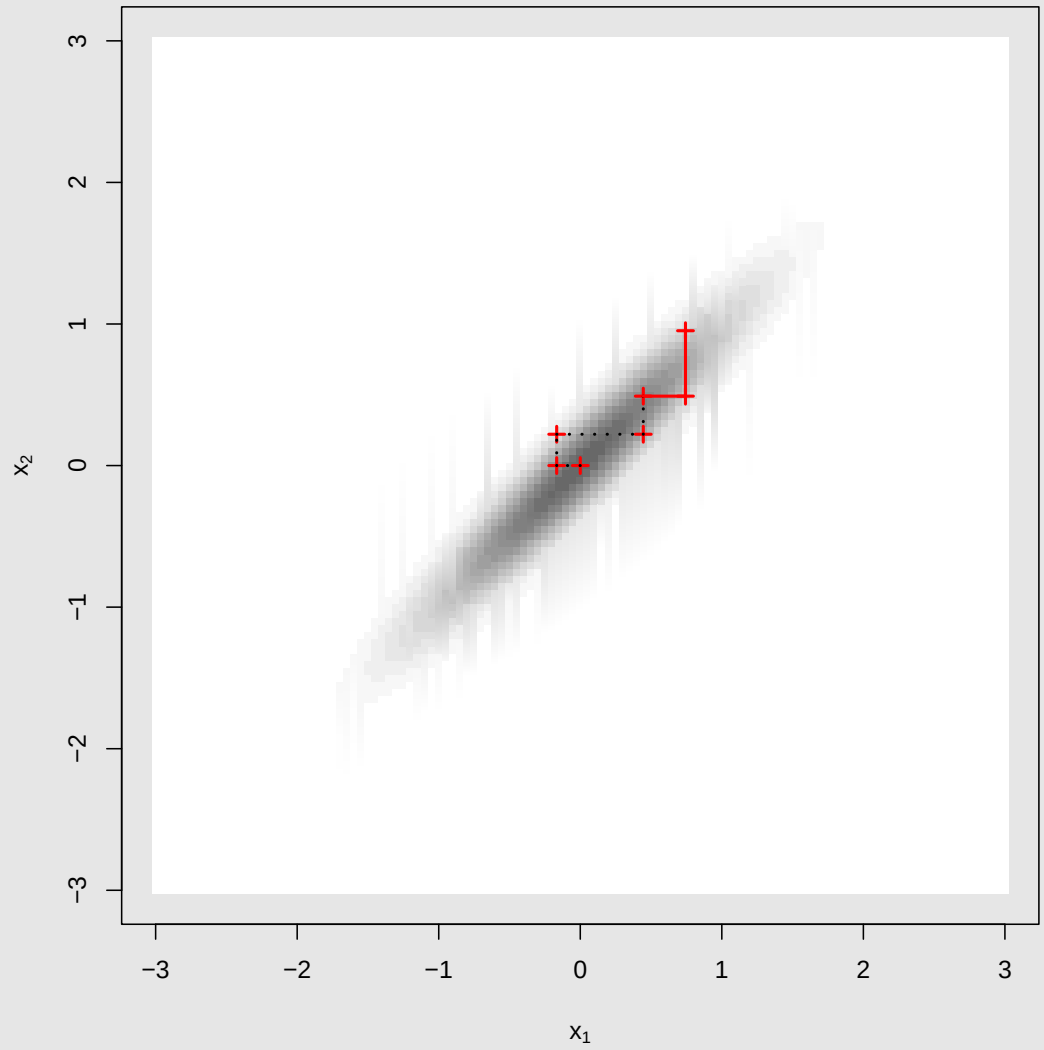
After two updates: (x_{12}, x_{22})



Gibbs Sampler: example

Bivariate Normal

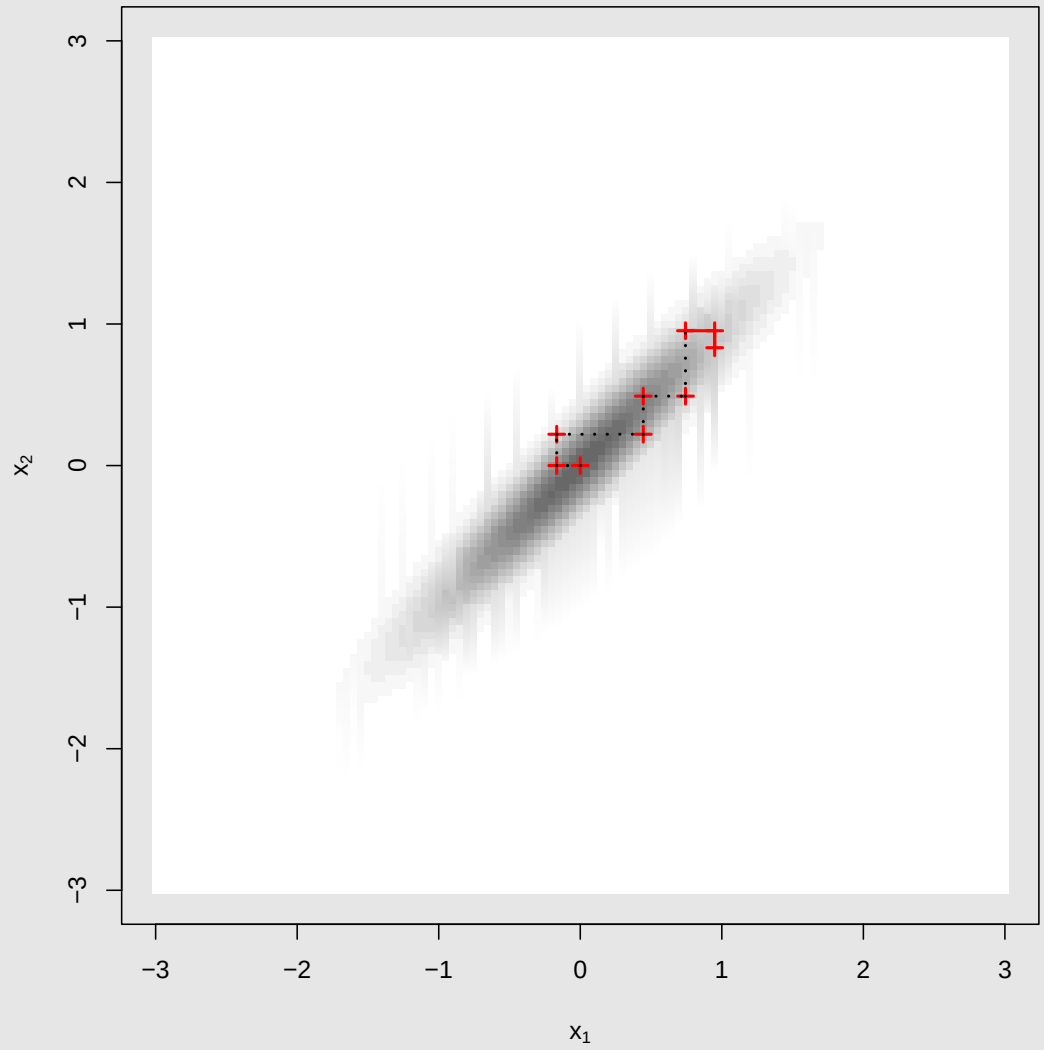
After three updates: (x_{13}, x_{23})



Gibbs Sampler: example

Bivariate Normal

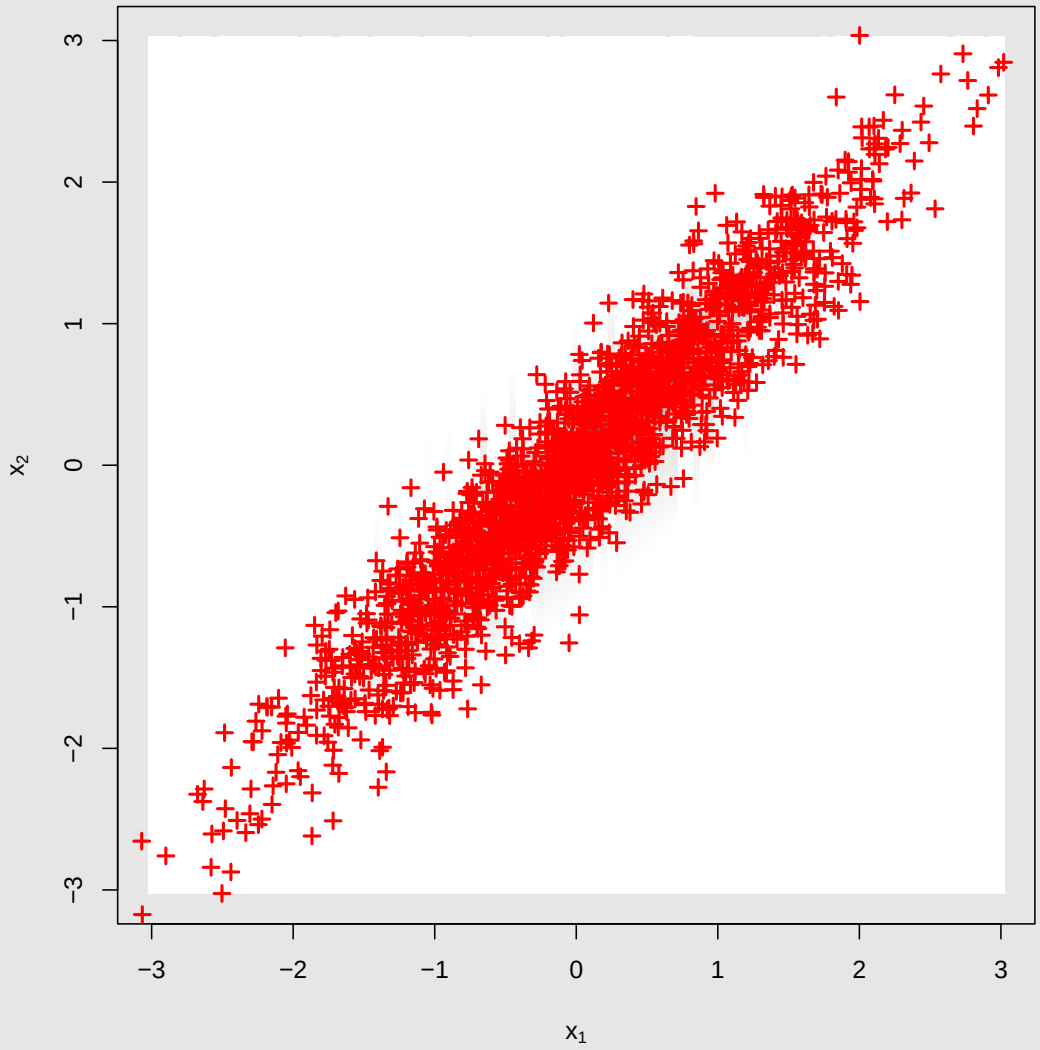
After four updates: (x_{14}, x_{24})



Gibbs Sampler: example

Bivariate Normal

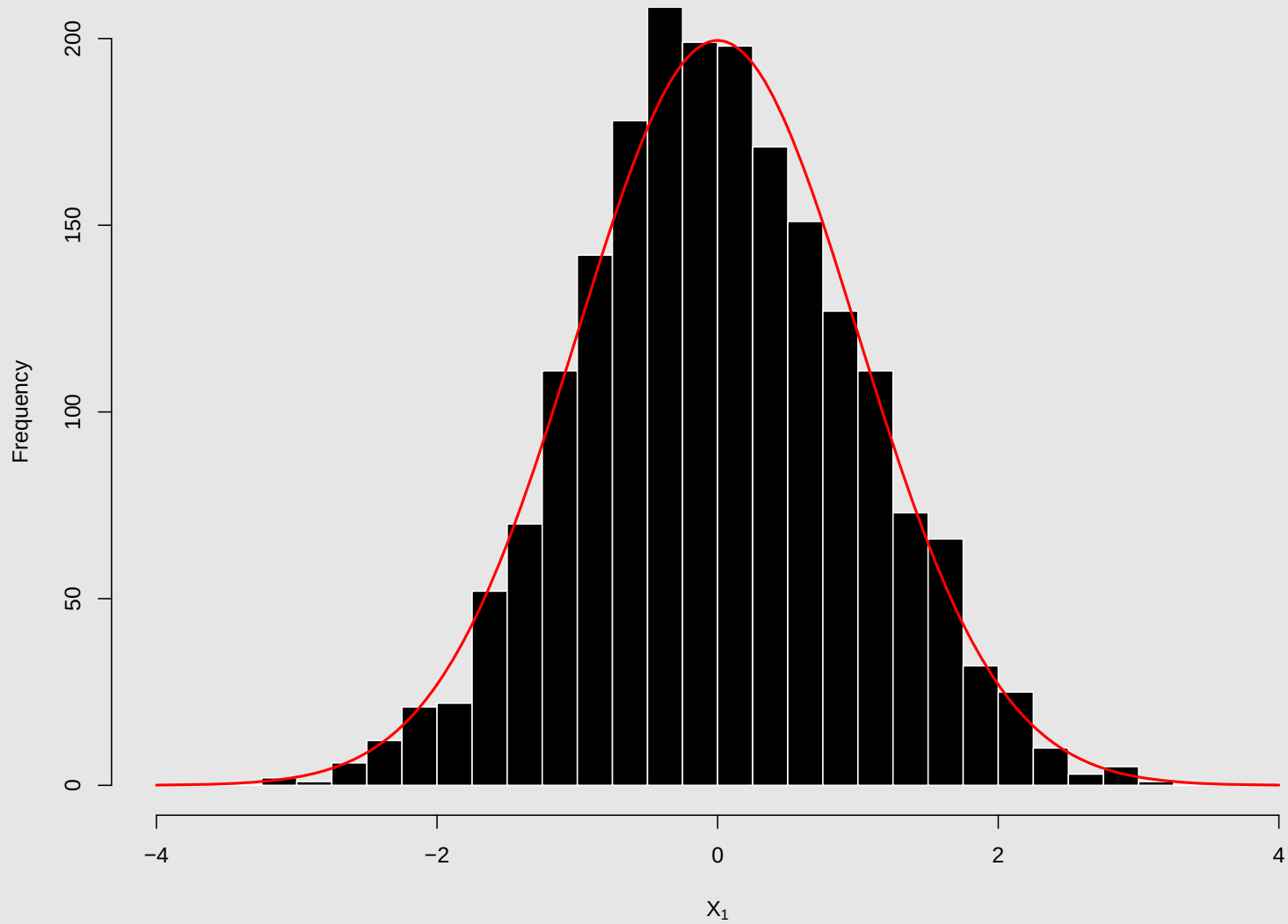
After 2000 updates: entire collected sample



Gibbs Sampler: example

Bivariate Normal

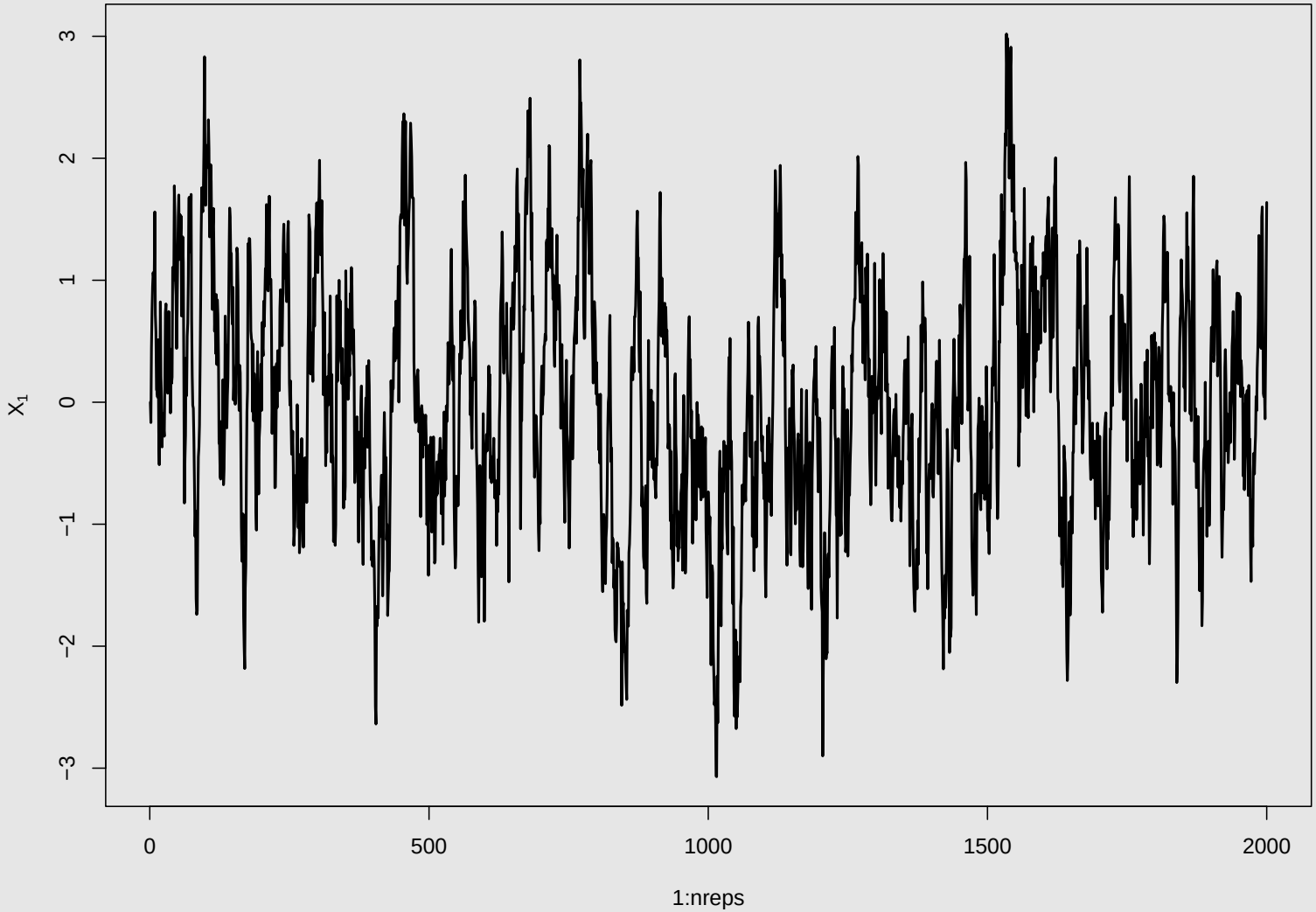
Histogram for X_1 with true marginal density (solid):



Gibbs Sampler: example

Bivariate Normal

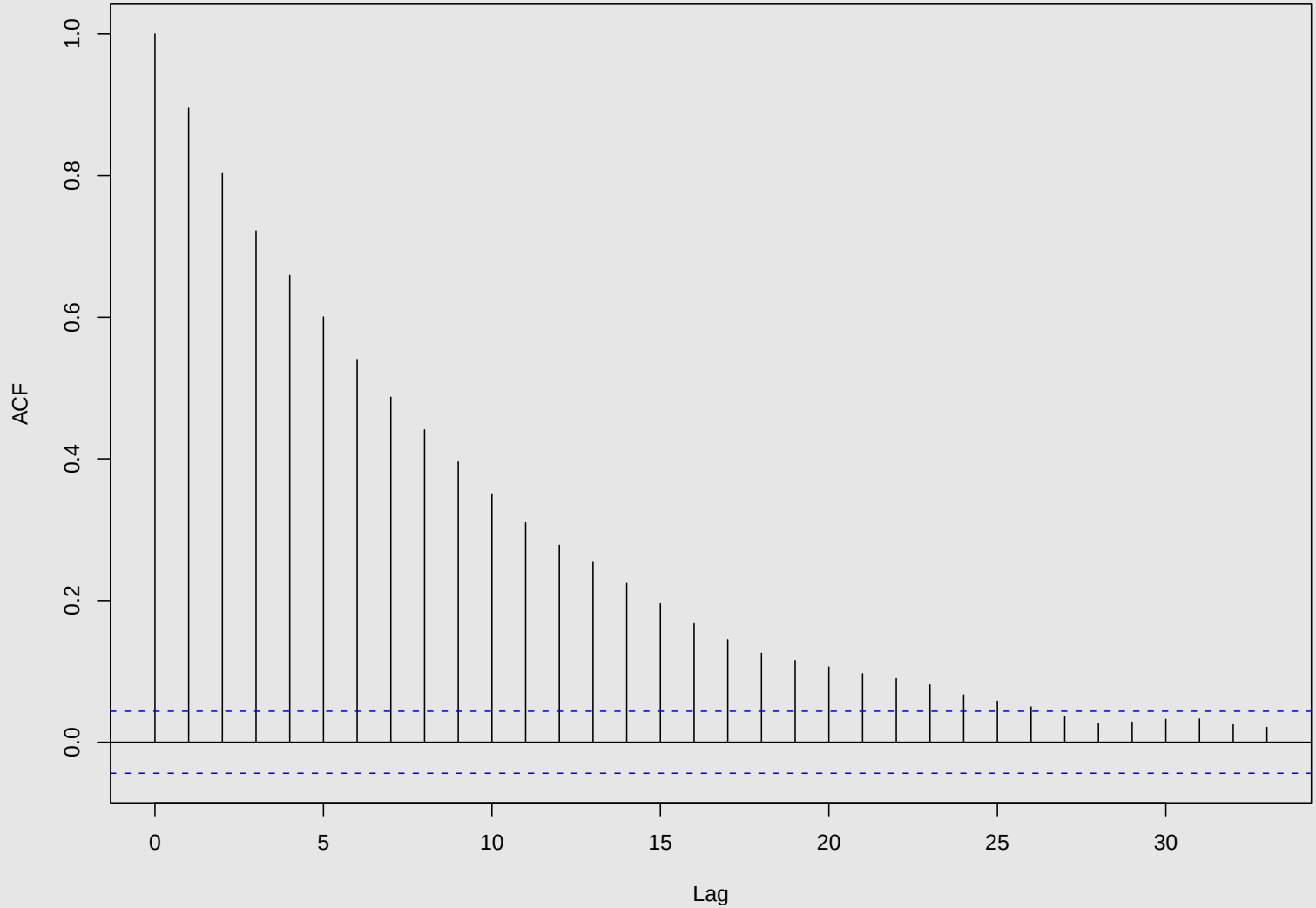
Trace for X_1:



Gibbs Sampler: example

Bivariate Normal

ACF for X_1:



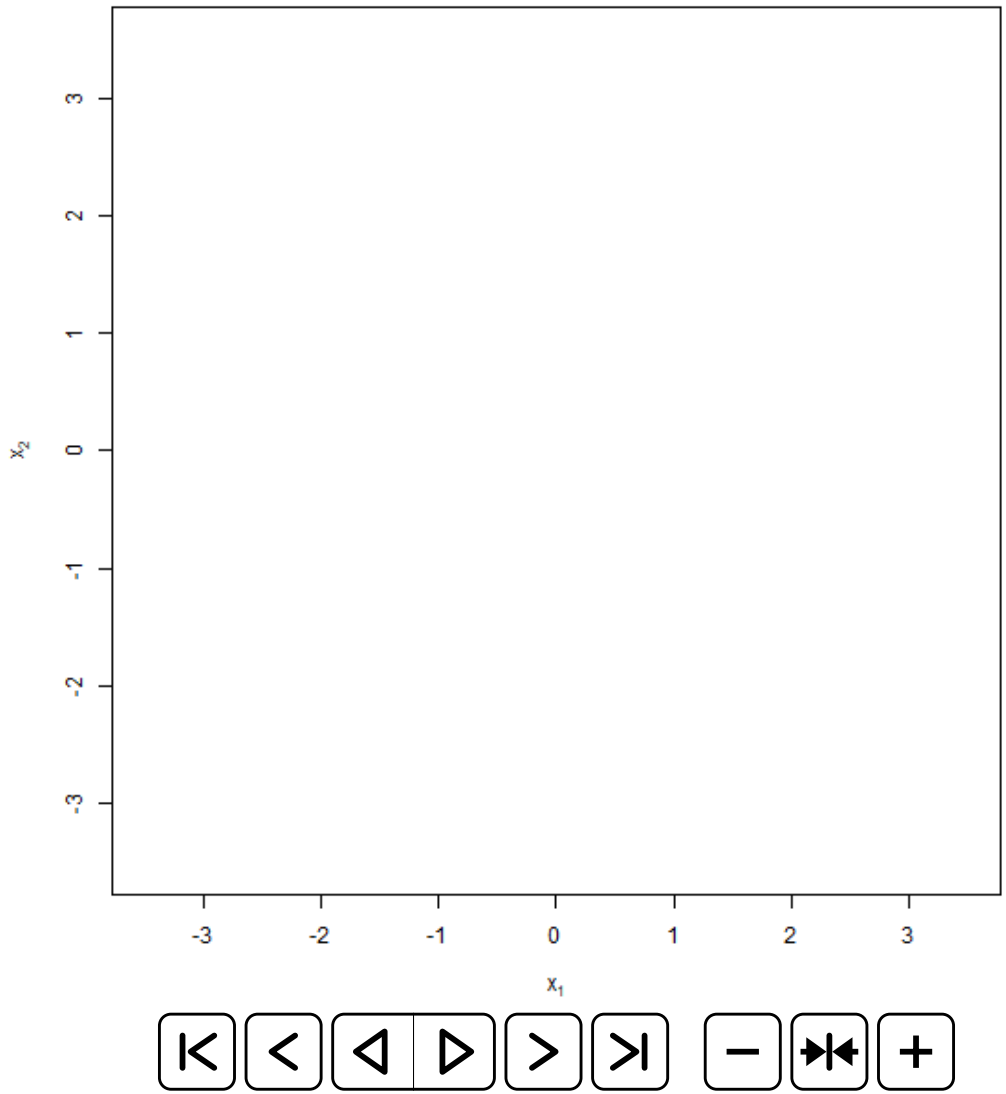
Gibbs Sampler: example

Bivariate Normal

With $\rho = 0.95$, the Gibbs sampler still performs adequately, but the moves made are smaller, and exploring the distribution is much more difficult.

This illustrates a potential general problem with the Gibbs sampler: although it is straightforward to implement as it involves only sampling variates from univariate densities, the restriction to moves along the coordinate axes can cause problems if the variables are **highly correlated**.

Gibbs Sampler: $\rho = -0.95$



Metropolis-Hastings in higher dimensions

The Metropolis-Hastings algorithm can be used for probability distributions in arbitrary dimension. Note that it is always the case that, for the full conditional distributions, if

- $\mathbf{x}_1$ is a sub-vector of the entire vector $\mathbf{x}$ of variables, and
- $\mathbf{x}_{(1)}$ is $\mathbf{x}$ with the components $\mathbf{x}_1$ removed,

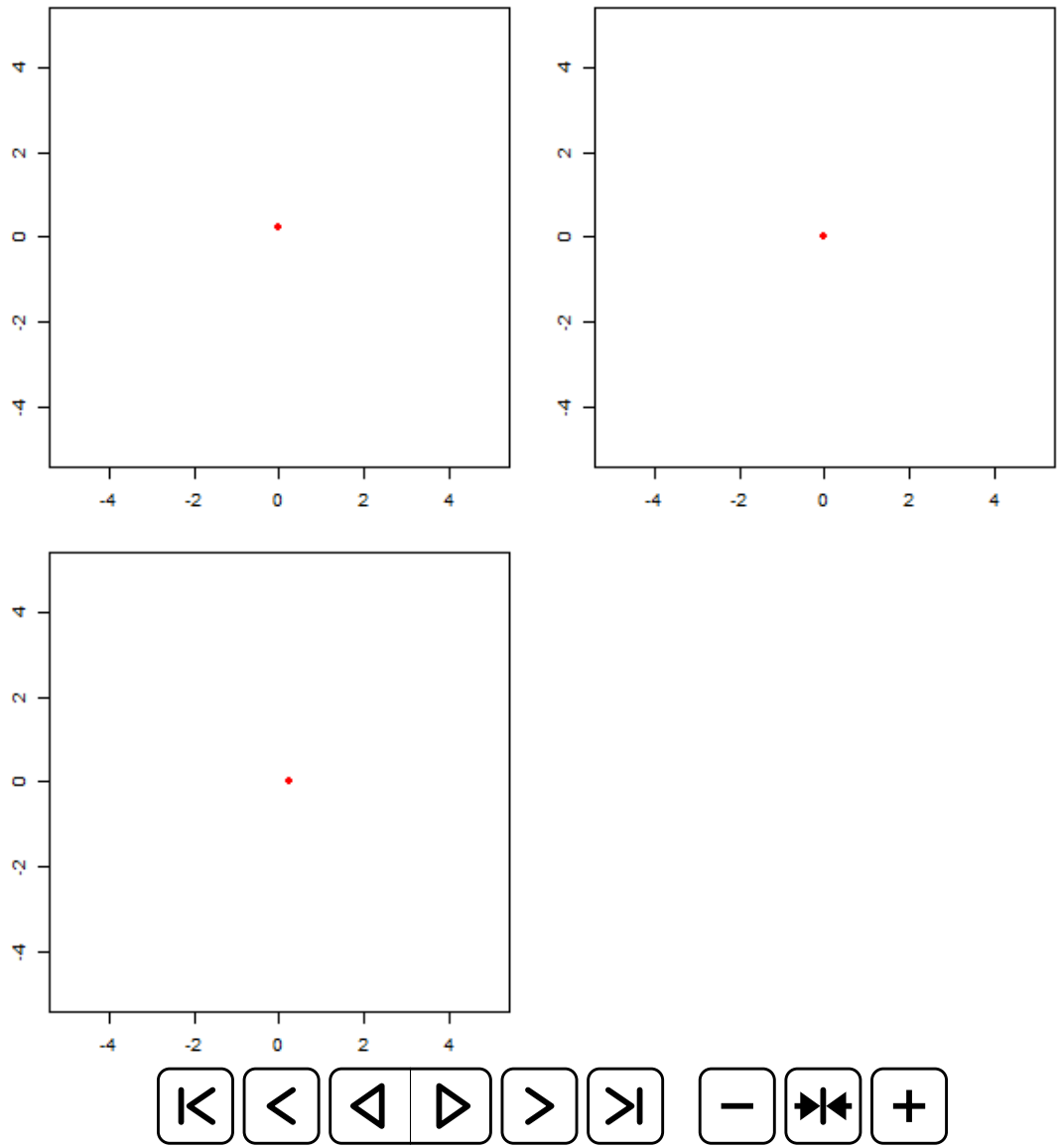
then

$$\pi(\mathbf{x}_1|\mathbf{x}_{(1)}) \propto \pi(\mathbf{x})$$

as the normalizing constant is $\pi(\mathbf{x}_{(1)})$, which does not depend on $\mathbf{x}_1$.

For *Metropolis-within-Gibbs*, knowing $\pi(\mathbf{x}_1|\mathbf{x}_{(1)})$ up to proportionality is sufficient, as the normalizing constant cancels out in the calculation of the acceptance probability.

Example: Logistic Regression $n = 25$, $p = 3$



More Examples

For more details and examples: Appendix

More Examples

Summary

- **Simulation-based inference** is a powerful tool for carrying out Bayesian analysis;
- In complex or high-dimensional models, **Markov chain Monte Carlo** can be used to sample the posterior distribution effectively;
- Algorithm **design** is a crucial research component;
- MCMC facilitates inference in important biostatistical/health settings
  - *causal inference*
  - *spatial statistics*

Part IV

Appendix

Appendix Random Number Generator More Examples

Rejection sampling

Suppose that we wish to sample from an arbitrary density $p(x)$, but this is not straightforward directly. Suppose instead we have easy access to variates from the density $p_0(x)$, where

$$\frac{p(x)}{p_0(x)} < M \qquad \text{for all } x$$

Then the following algorithm produces variates from p:

1. generate x from p_0
2. generate u from Uniform$(0, 1)$
3. if
$$u \leqslant \frac{p(x)}{Mp_0(x)}$$
accept x as a variate from p; if this inequality is not met, return to 1.

Rejection sampling

This is the *rejection sampling* (or *accept-reject*) algorithm. The density p_0 is the *proposal density*.

It works for **any** bound M, but is most efficient (has fewest rejections) when M is as small as possible, for example, if

$$M = \sup_x \frac{p(x)}{p_0(x)}$$

If $p(x)$ is bounded, and has support which is a bounded subset $\mathcal{X}$ of $\mathbb{R}$, then p_0 can be chosen to be the Uniform density on $\mathcal{X}$, although this is not necessarily the optimal choice.

Rejection sampling

Normal mixture

Consider sampling from the normal mixture

$$p(x) = \frac{1}{4}\phi(x+2) + \frac{3}{4}\phi(x-1)$$

where $\phi(.)$ is the standard normal pdf. Consider rejection sampling from this density using

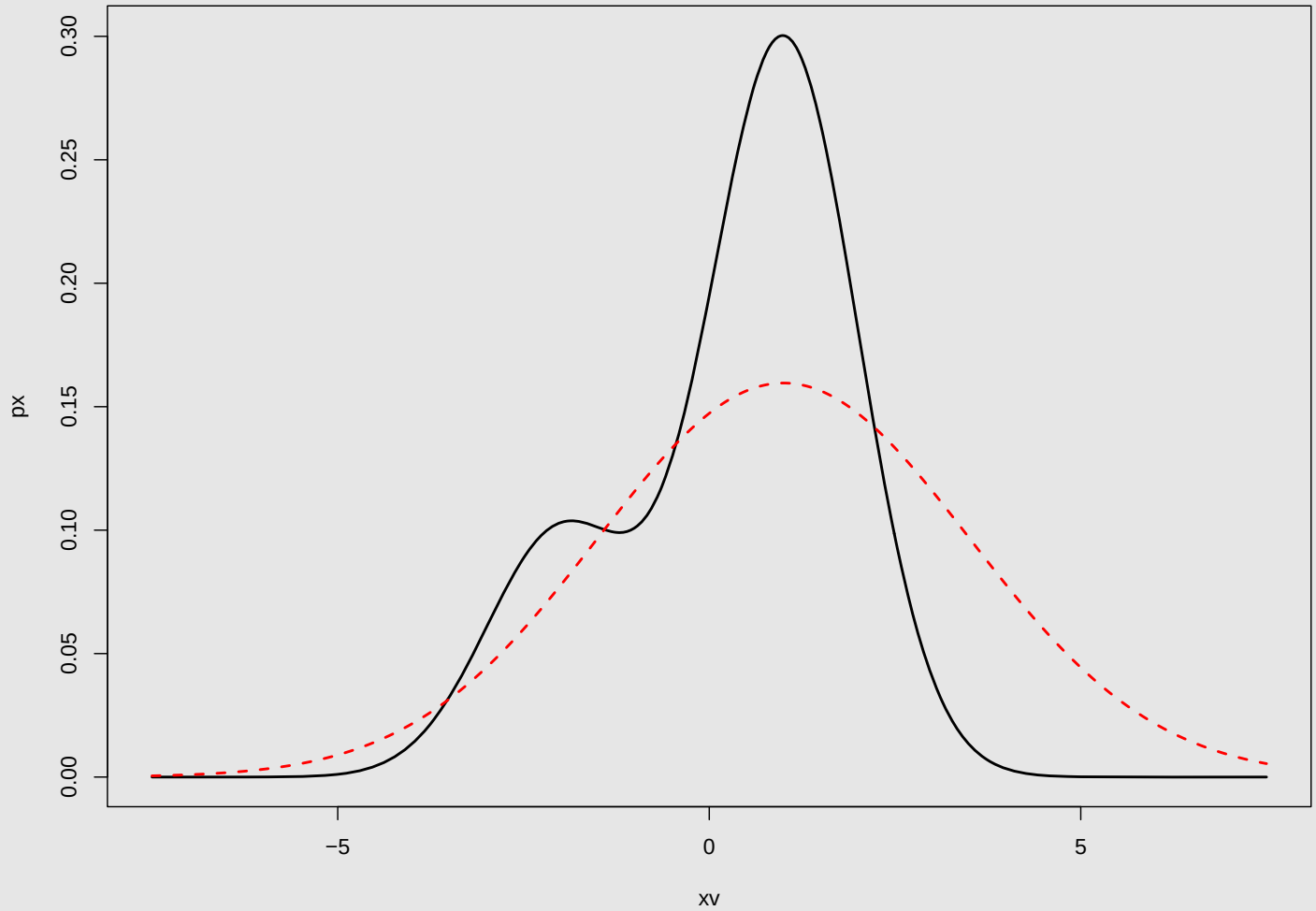
$$p_0(x) = \frac{1}{\sigma}\phi\left(\frac{x-1}{\sigma}\right)$$

for some variance σ^2.

Rejection sampling

Normal mixture

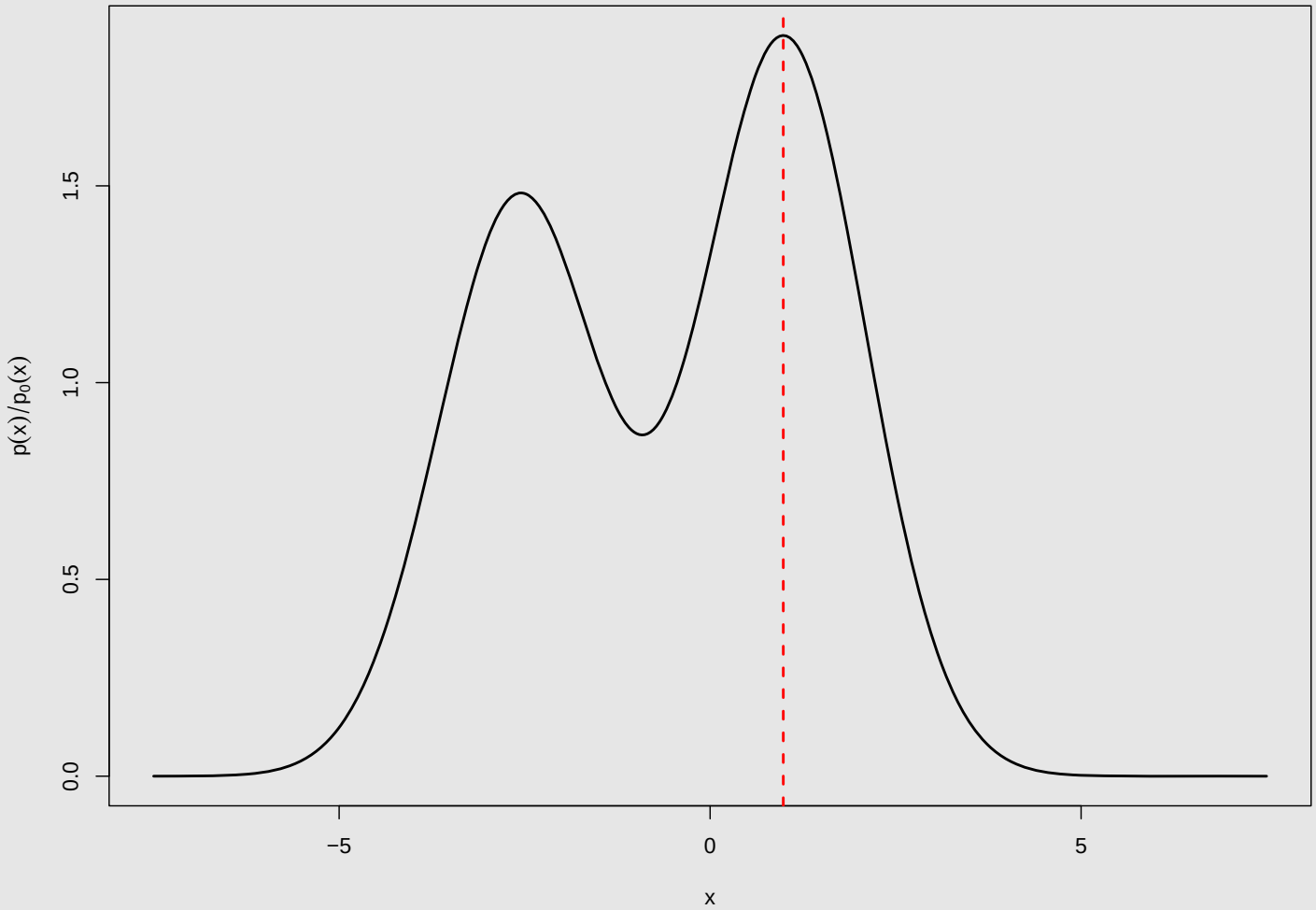
Target $p(x)$ (solid) and $p_0(x)$ (dashed) with $\sigma = 2.5$.



Rejection sampling

Normal mixture

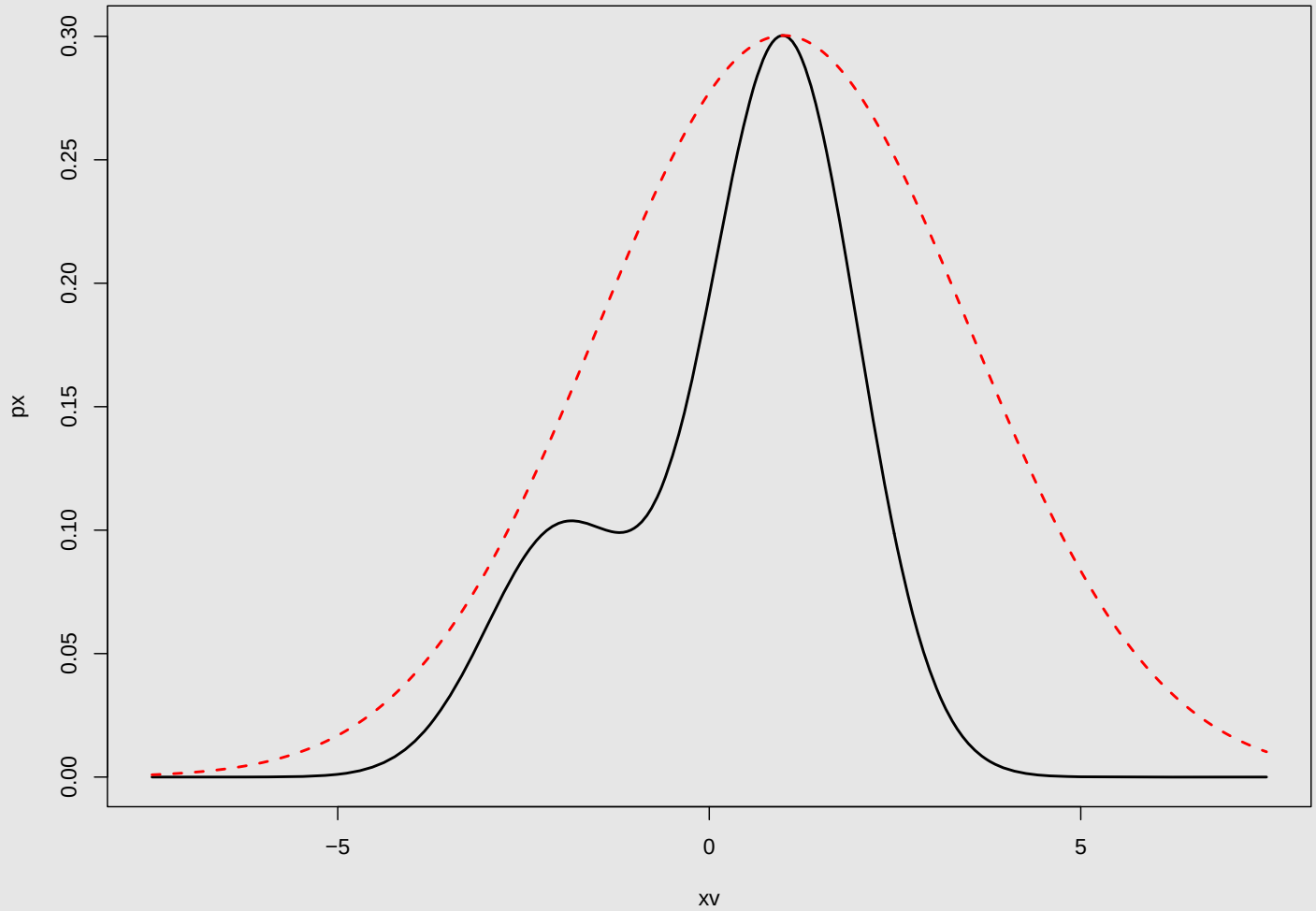
$p(x)/p_0(x)$ maximized at $x = 0.9863$, yielding $M = 1.8821$:



Rejection sampling

Normal mixture

$p(x)$ bounded by $Mp_0(x)$:



Rejection sampling

If x is a variate from p_0, then $p_0(x)$ is the value of the density at that variate, and $Mp_0(x)$ is the scaled version. Consider a vertical slice at x which is the line segment

$$(x, 0) \longrightarrow (0, Mp_0(x))$$

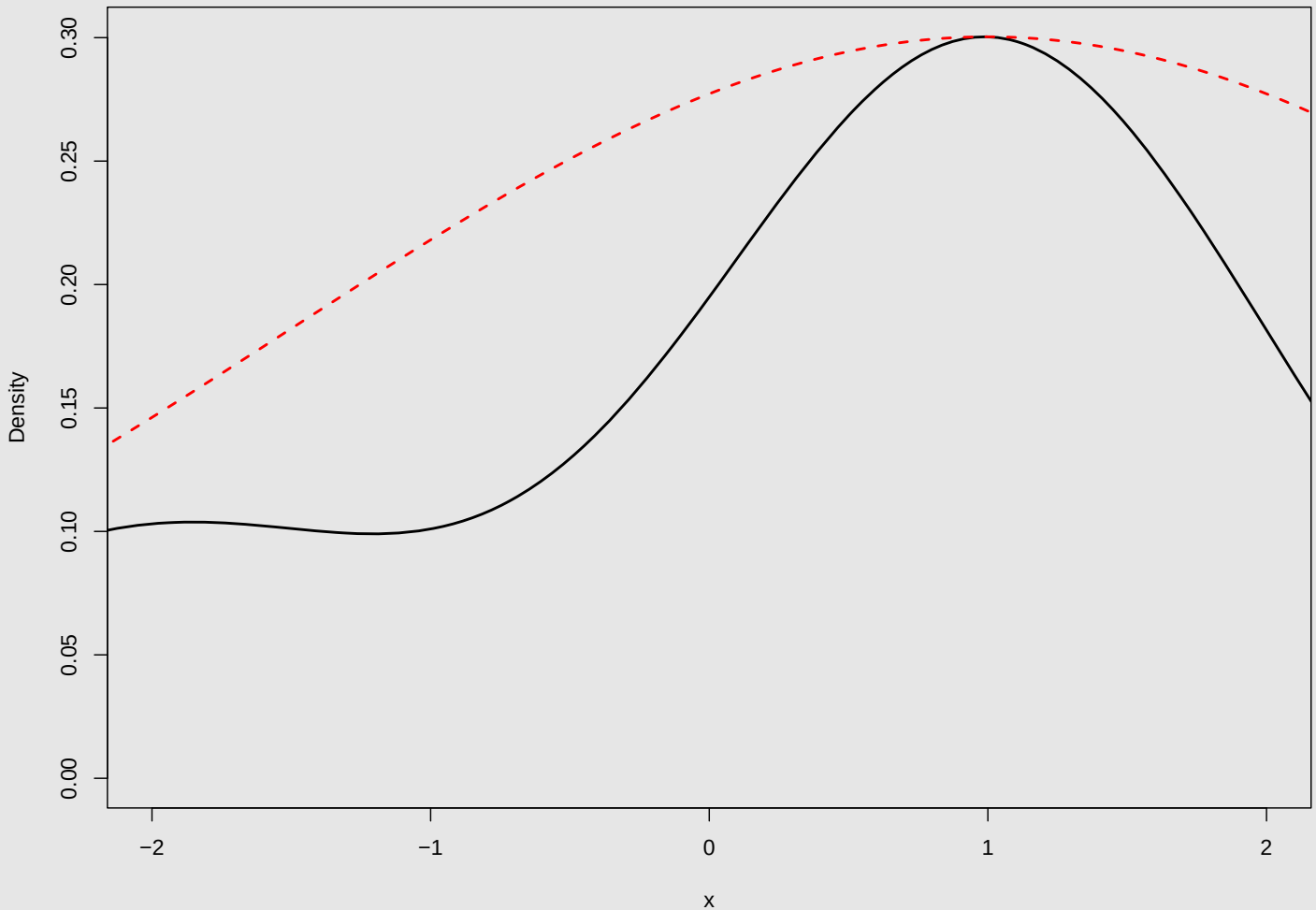
By assumption $p(x) \leqslant Mp_0(x)$. If $u \sim \text{Uniform}(0, 1)$, then $uMp_0(x)$ is the random portion of the vertical slice, and if

$$p(x) \leqslant uMp_0(x)$$

then $p(x)$ is **below** the random point on the line segment, and hence is rejected.

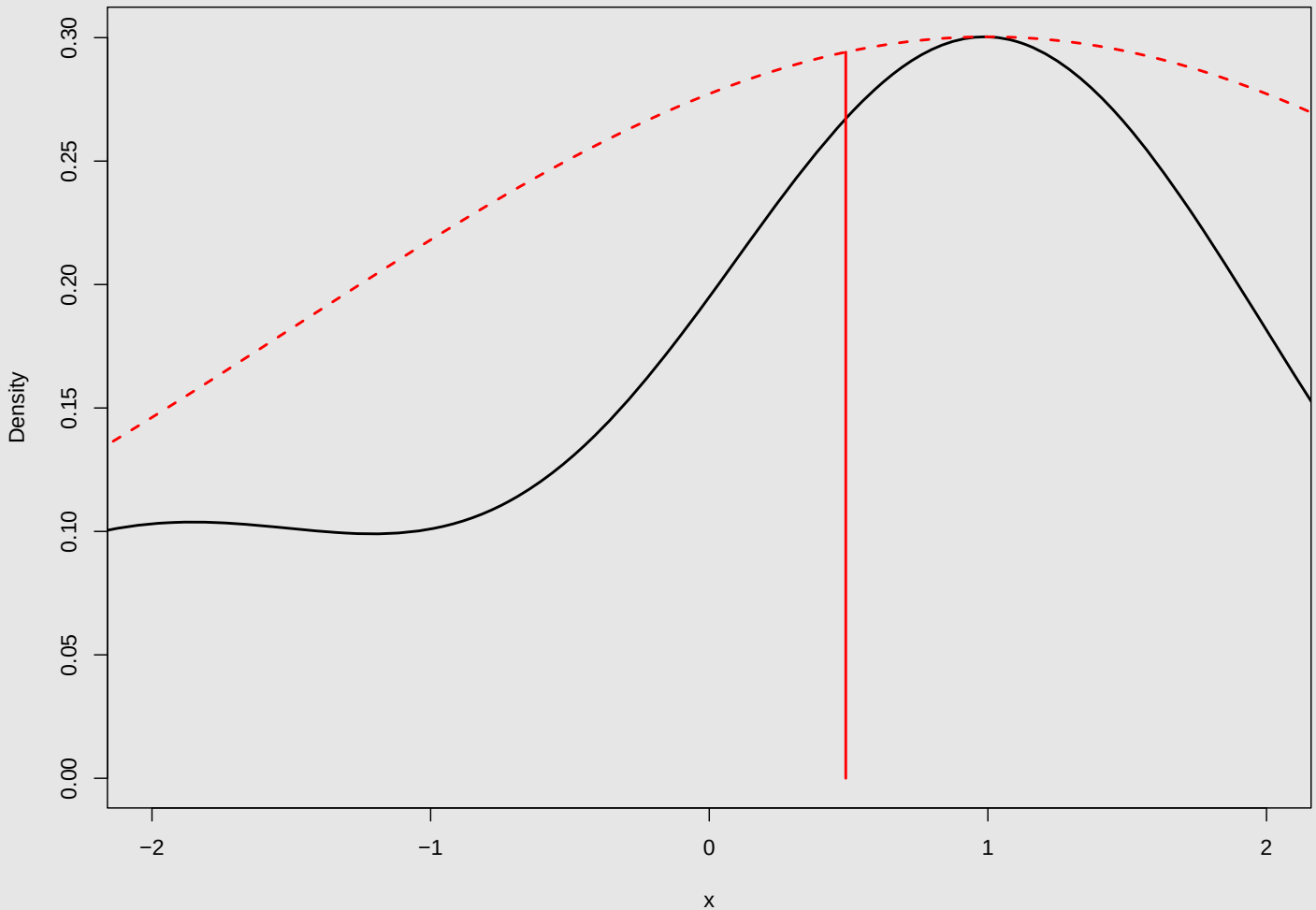
Rejection sampling

Normal mixture



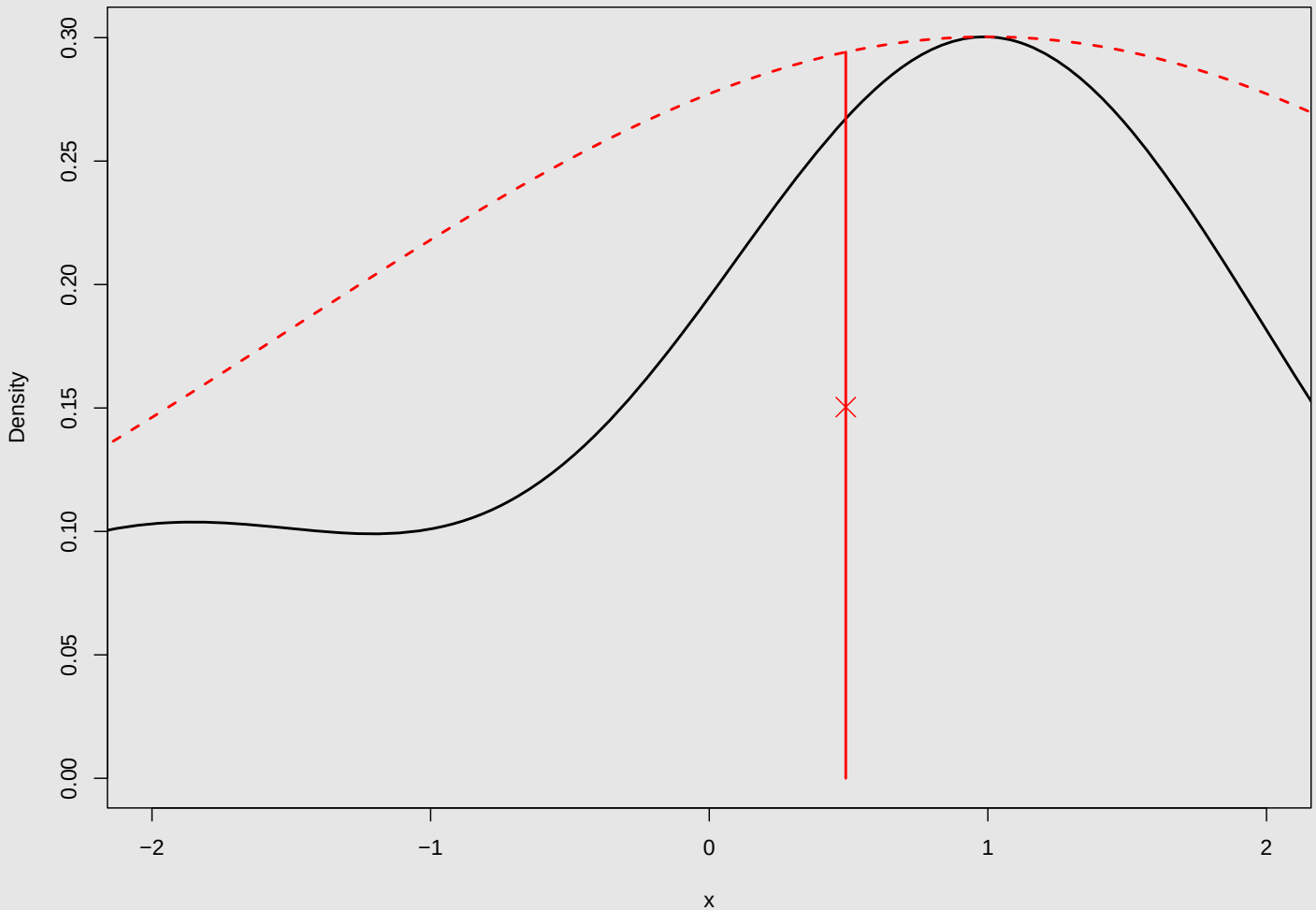
Rejection sampling

Normal mixture



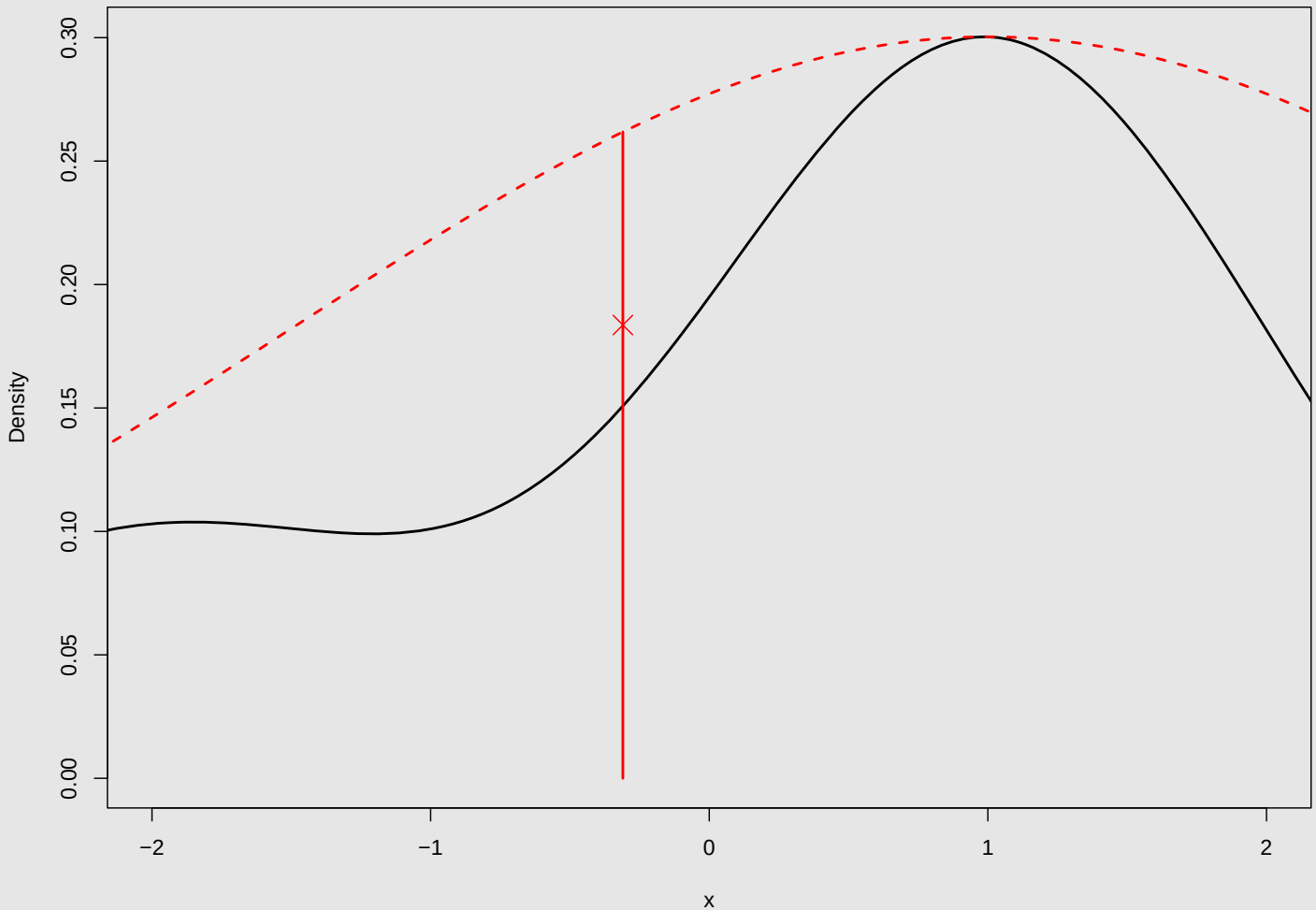
Rejection sampling

Normal mixture



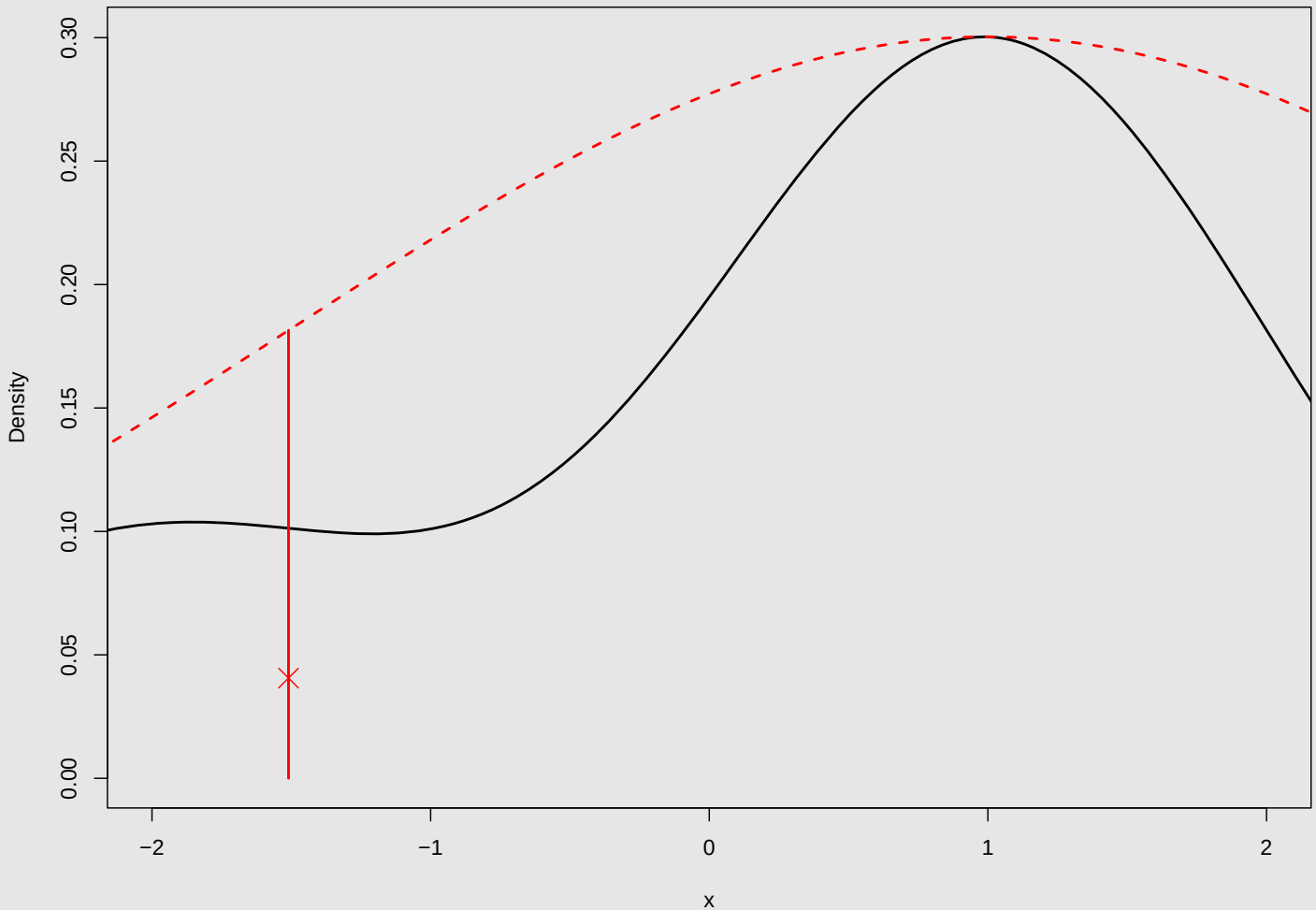
Rejection sampling

Normal mixture



Rejection sampling

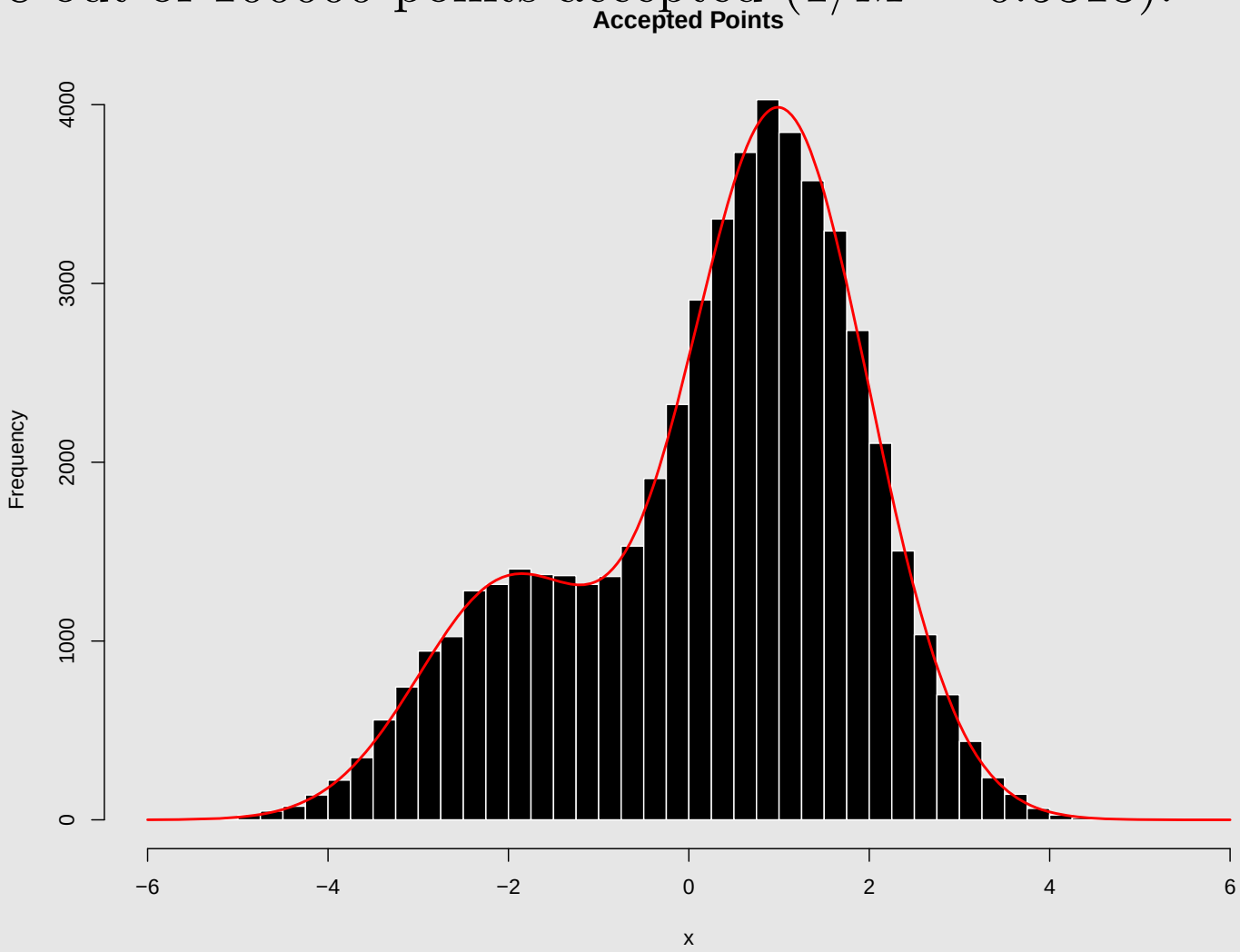
Normal mixture



Rejection sampling

Normal mixture

53078 out of 100000 points accepted ($1/M = 0.5313$).



Rejection sampling

Notes:

- The acceptance rate is 1 if $p_0(x) \equiv p(x)$.
- If $p_0(x)$ resembles $p(x)$, then the acceptance rate will be high.
- The maximization of $p(x)/p_0(x)$ can be done numerically.
- The extension to the multivariate case is straightforward
- The proposal density can be constructed *adaptively*; whenever a point is rejected, it is used to update the proposal function.

Rejection sampling and Importance sampling

There is obviously a clear connection between rejection sampling and importance sampling

- both rely on choosing a suitable $p_0(x)$
- the boundedness of the ratio $p(x)/p_0(x)$ is crucial in the construction of the procedure

The difference is that rejection sampling produces i.i.d. samples from $p(x)$, whereas importance sampling approximates numerical integration with respect to $p(x)$.

Adaptive Rejection sampling

If $p(x)$ is *log-concave* (that is $\log p(x)$ is concave), then *adaptive rejection sampling* constructs an adaptive proposal density using rejected points.

Even when a point x is rejected, $p(x)$ is computed, and this information is useful improving p_0.

For example, consider sampling from $Gamma(\alpha, 1)$ for noninteger $\alpha > 0$. We have

$$\log p(x) = \text{const.} + (\alpha - 1) \log x - x$$

which is concave, as the second derivative is negative.

Adaptive Rejection sampling

The proposal function $p_0(x)$ can be taken as $Exponential(1/2)$; in this case

$$\frac{p(x)}{p_0(x)} = \frac{2}{\Gamma(\alpha)} x^{\alpha-1} e^{-x/2}$$

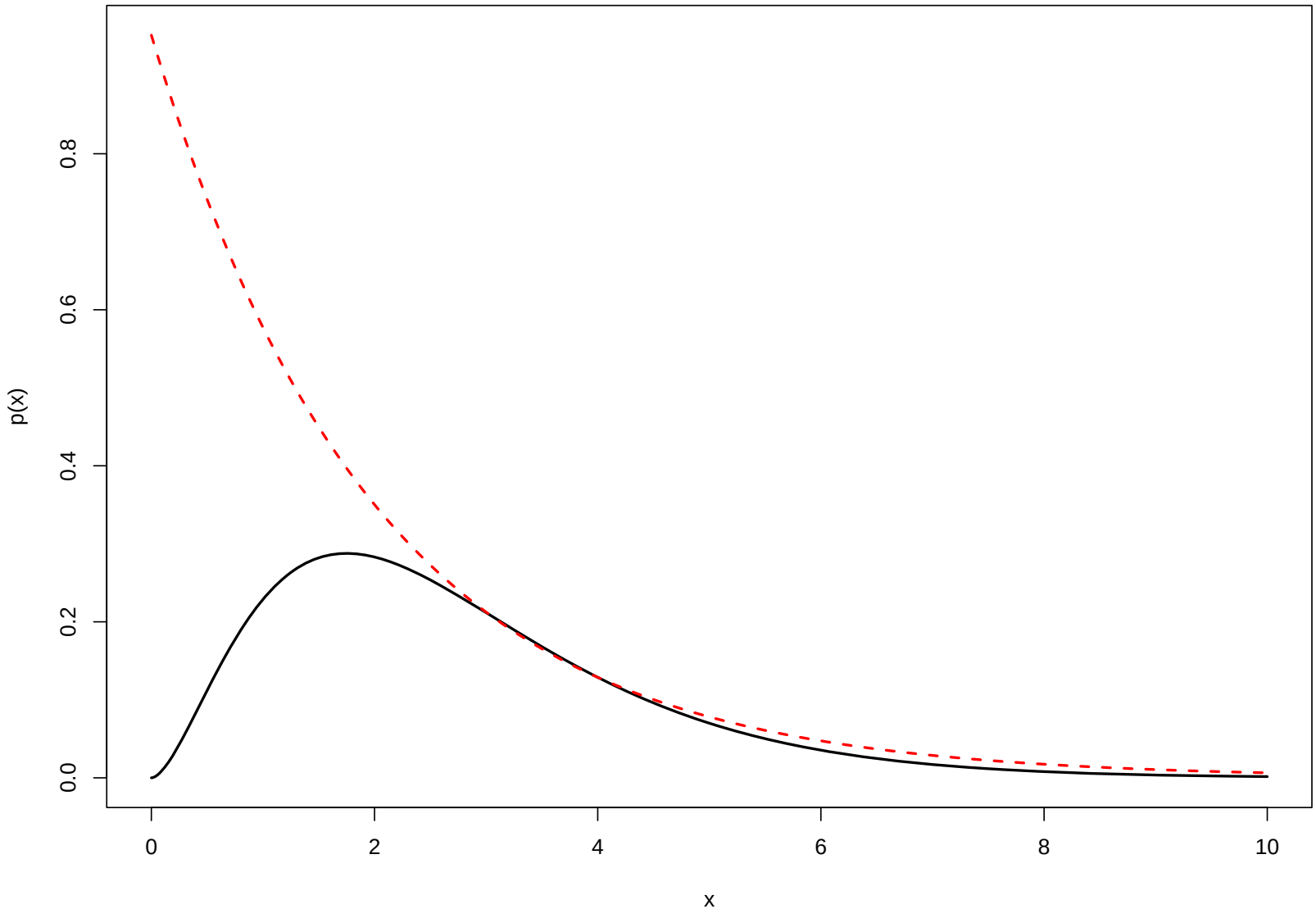
which achieves its unique maximum at

$$x_{\max} = 2(\alpha - 1)$$

Thus rejection sampling can be carried out using this p_0.

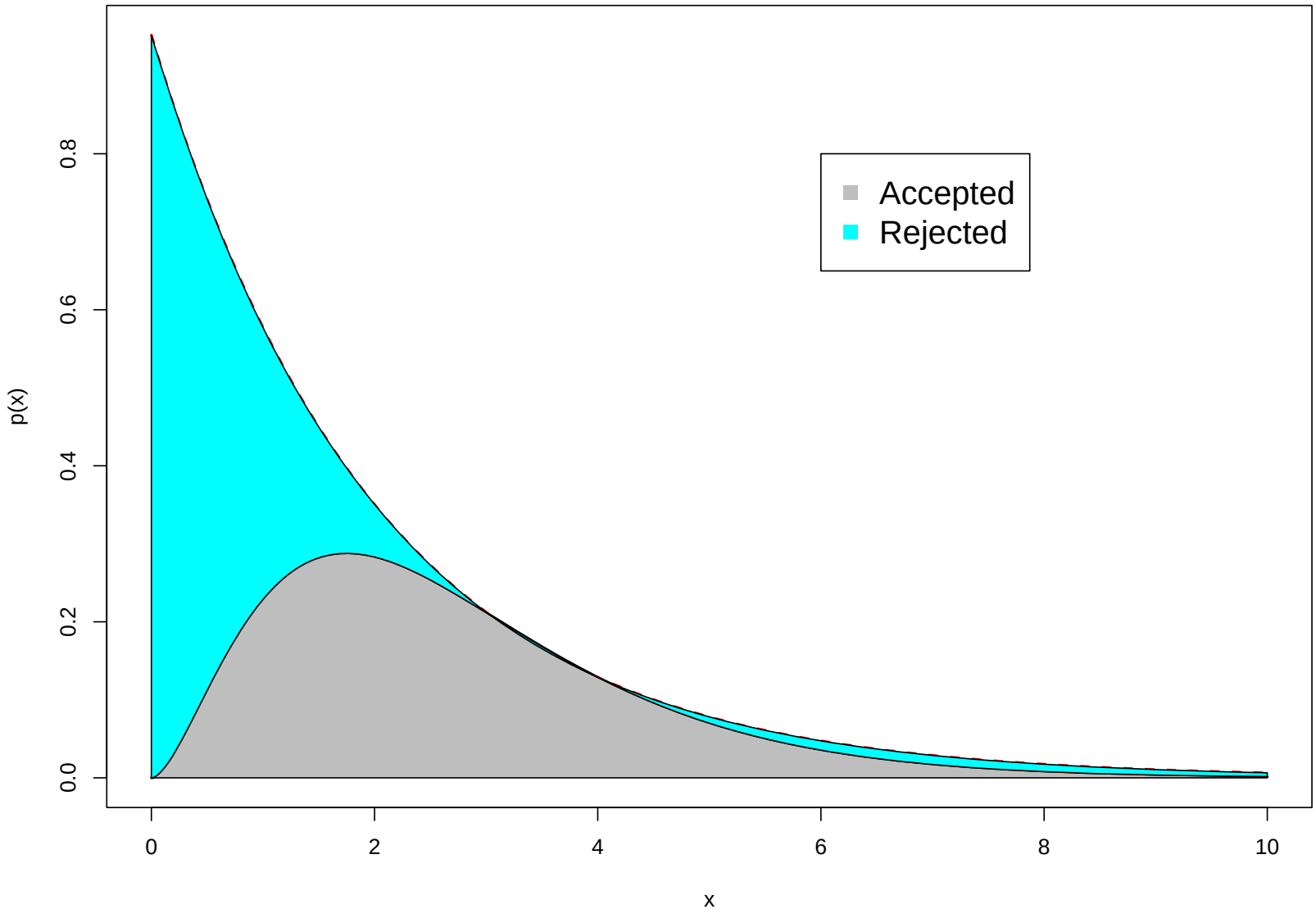
Adaptive Rejection sampling

Exponential(1/2) distribution is potentially inefficient



Adaptive Rejection sampling

Exponential(1/2) distribution is potentially inefficient



Adaptive Rejection sampling

In this problem

$$M = \frac{2}{\Gamma(\alpha)} x_{\max}^{\alpha-1} e^{-x_{\max}/2}$$

so acceptance rate is the reciprocal of this

α	1	2	3	4	5	6	7	8
$1/M$	0.5000	0.6796	0.4618	0.2790	0.1600	0.0890	0.0486	0.0262

For large α, the acceptance rate becomes very small.

Adaptive Rejection sampling

Adaptive rejection sampling proceeds by adapting the *squeezing* method. Suppose that there exists a function $l(x)$ such that

$$l(x) \leqslant p(x) \leqslant Mp_0(x).$$

Then the following algorithm generates a variate from $p(x)$:

- [1.] Generate x from p_0, and u from $Uniform(0, 1)$
- [2.] Accept x if $u \leqslant l(x)/(Mp_0(x))$ and stop.
- [3.] Accept x if $u \leqslant p(x)/(Mp_0(x))$ and stop.
- [4.] Return to 1.

The function $l(.)$ provides a screening mechanism.

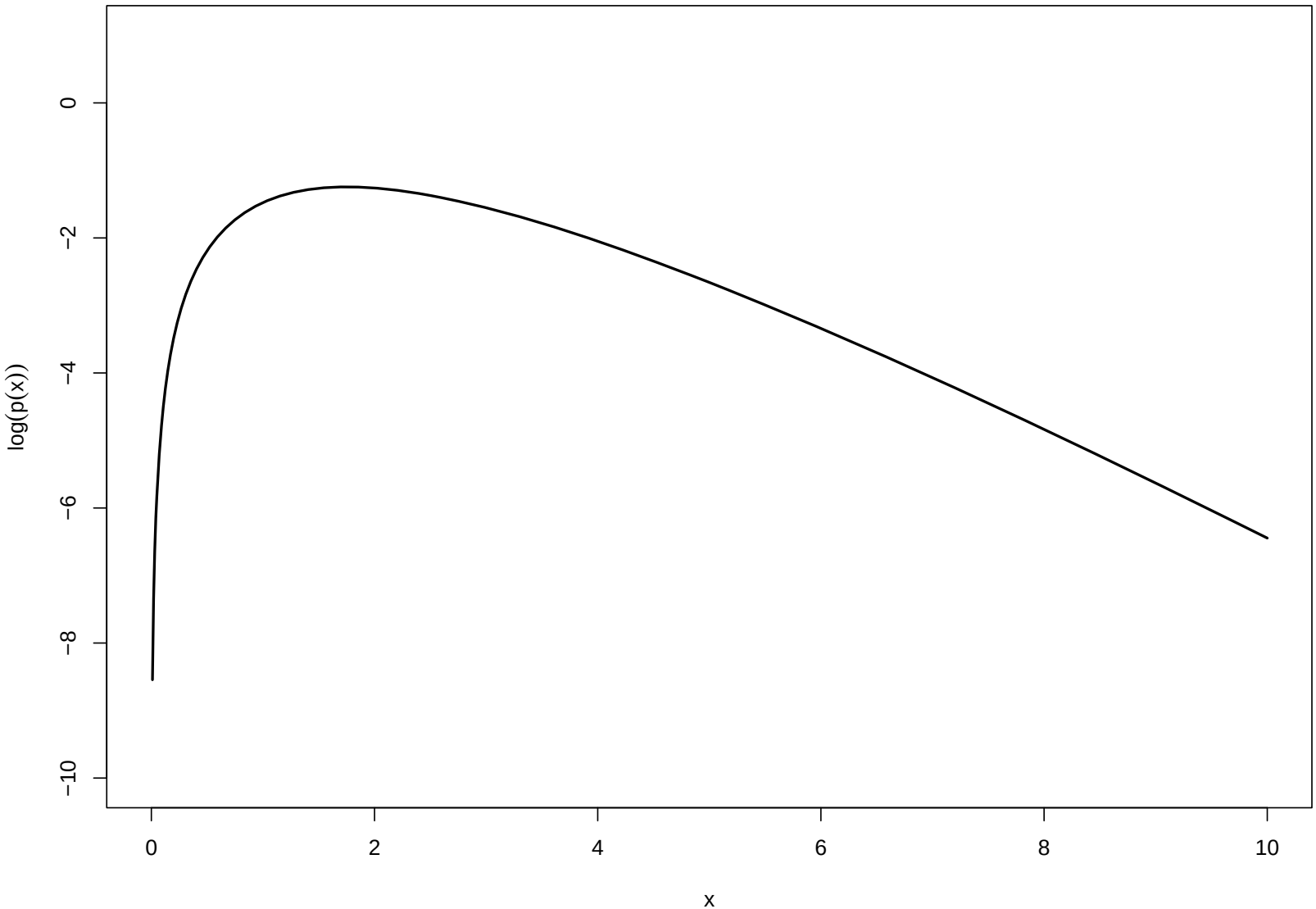
Adaptive Rejection sampling

For an adaptive version for log-concave densities, suppose that $(x_0, \ldots, x_{n+1})$ is an ordered set of points at which $y_i = \log p(x_i)$ has been evaluated for each i. For a log-concave density, the line segment

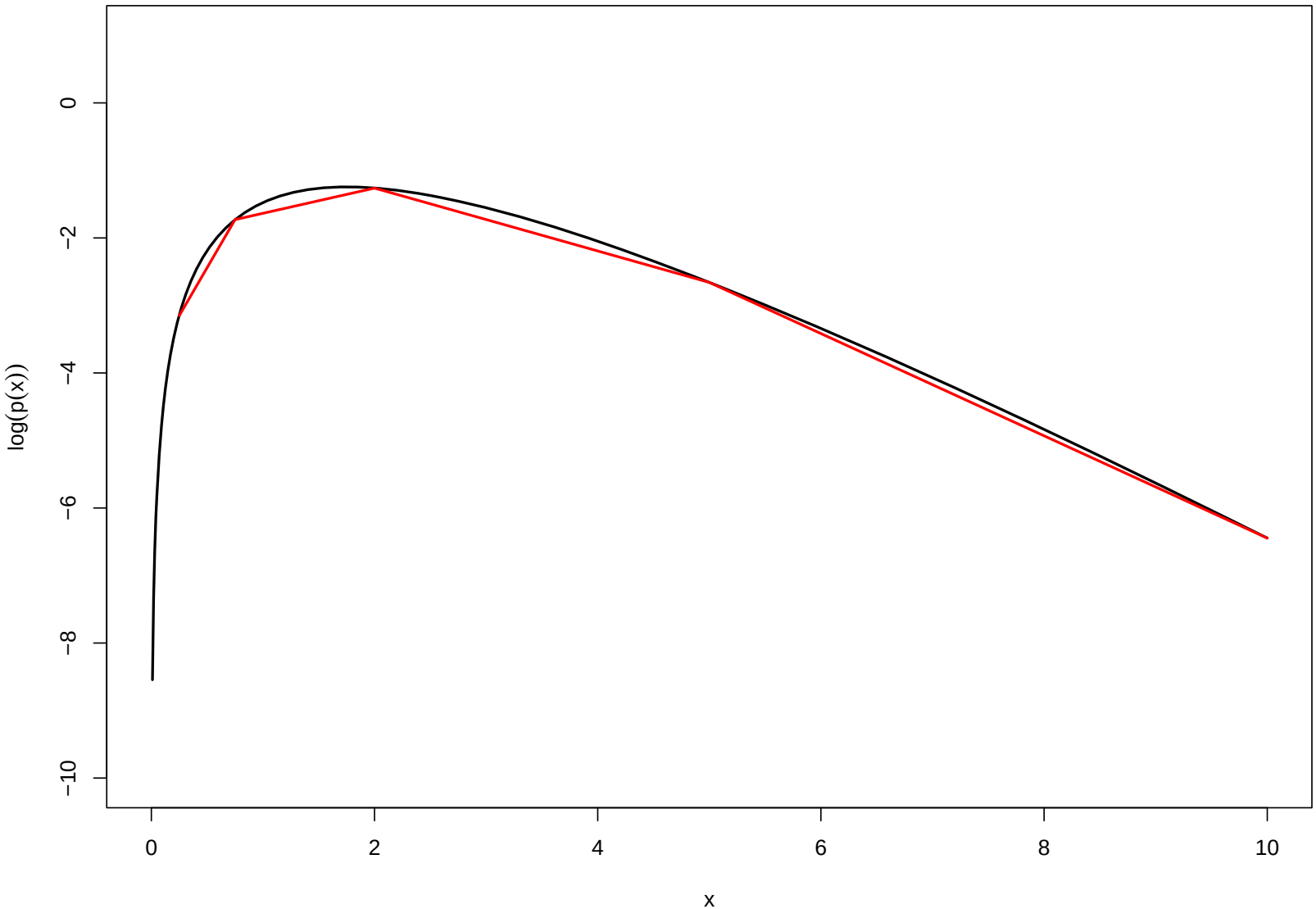
$$L_{i,i+1}(x) : (x_i, y_i) \longrightarrow (x_{i+1}, y_{i+1})$$

for $x_i \leqslant x \leqslant x_{i+1}$ is **below** $\log p(x)$ on (x_i, x_{i+1}), and above $\log p(x)$ on the rest of the range.

Adaptive Rejection sampling



Adaptive Rejection sampling



Adaptive Rejection sampling

Therefore, consider the squeezing envelope defined piecewise on the support of $p(x)$. For $x_i \leqslant x \leqslant x_{i+1}$

$$\begin{aligned} \text{Lower} \quad &: \quad l_n(x) = L_{i,i+1}(x) \\ \text{Upper} \quad &: \quad u_n(x) = \min\{L_{i-1,i}(x), L_{i+1,i+2}(x)\} \end{aligned}$$

where, for each i

$$L_{i,i+1}(x) = y_i + \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)(x - x_i)$$

Take

$$l_n(x) = -\infty \qquad\qquad u_n(x) = \min\{L_{0,1}(x), L_{n,n+1}(x)\}$$

for x outside of (x_0, x_{n+1}).

Adaptive Rejection sampling

The scheme needs initialization; usually two function evaluations at x values either side of the mode are sufficient.

The lower bound $l_n(x)$ and upper bound $u_n(x)$ yield the following bounds on exponentiation

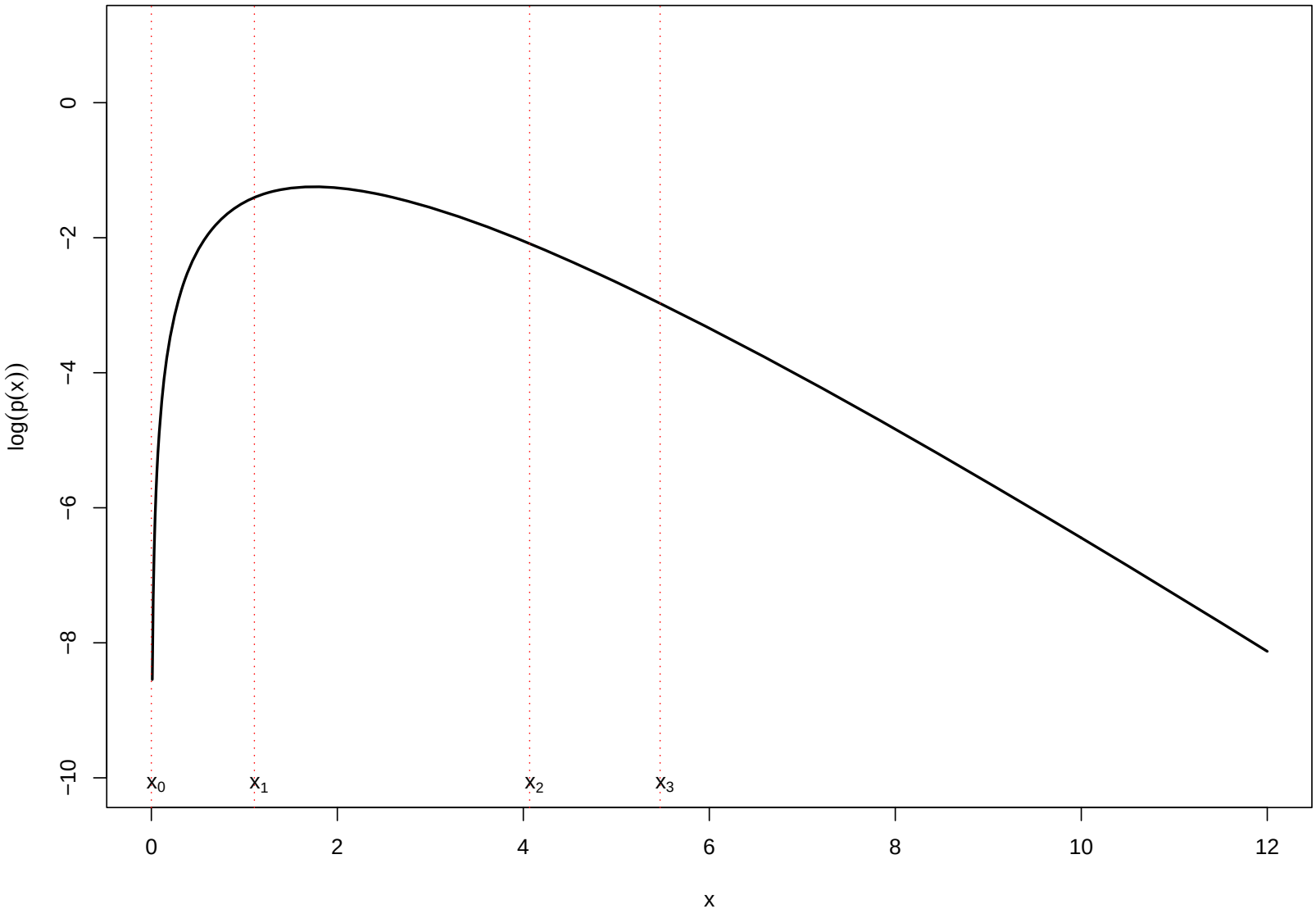
$$\exp\{l_n(x)\} \leqslant p(x) \leqslant \exp\{u_n(x)\} = M_n p_{0n}(x)$$

say, where M_n is a scaling constant such that $p_{0n}(x)$ is a density function

$$p_{0n}(x) = \frac{\sum_{i=0}^{n} \exp\{m_i x + c_i\} \mathbb{1}_{[x_i, x_{i+1}]}(x)}{\sum_{i=0}^{n} \int_{-\infty}^{\infty} \exp\{m_i t + c_i\} \mathbb{1}_{(x_i, x_{i+1})}(t)\, dt}$$

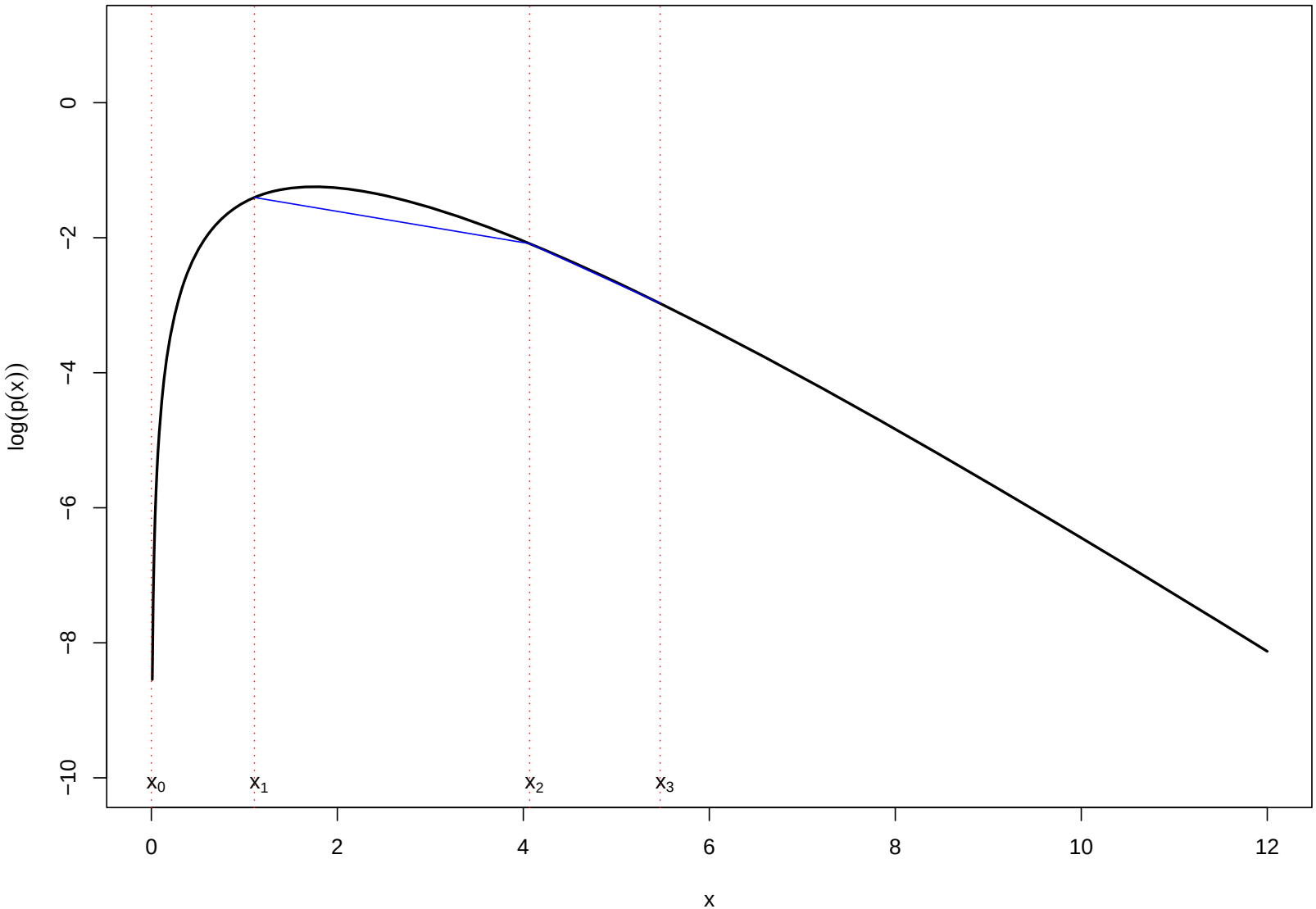
Adaptive Rejection sampling

Log-density:



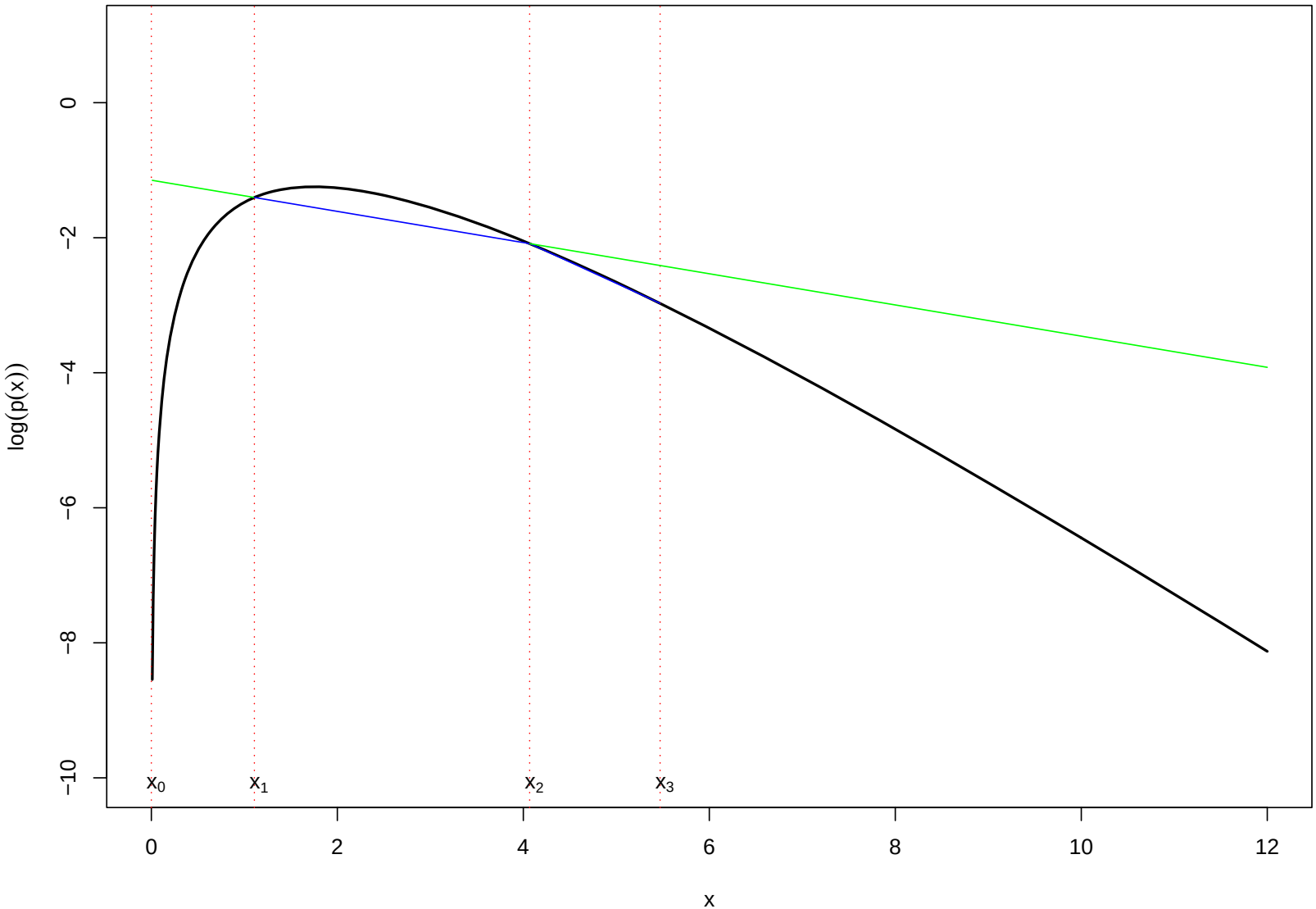
Adaptive Rejection sampling

Lower Envelope



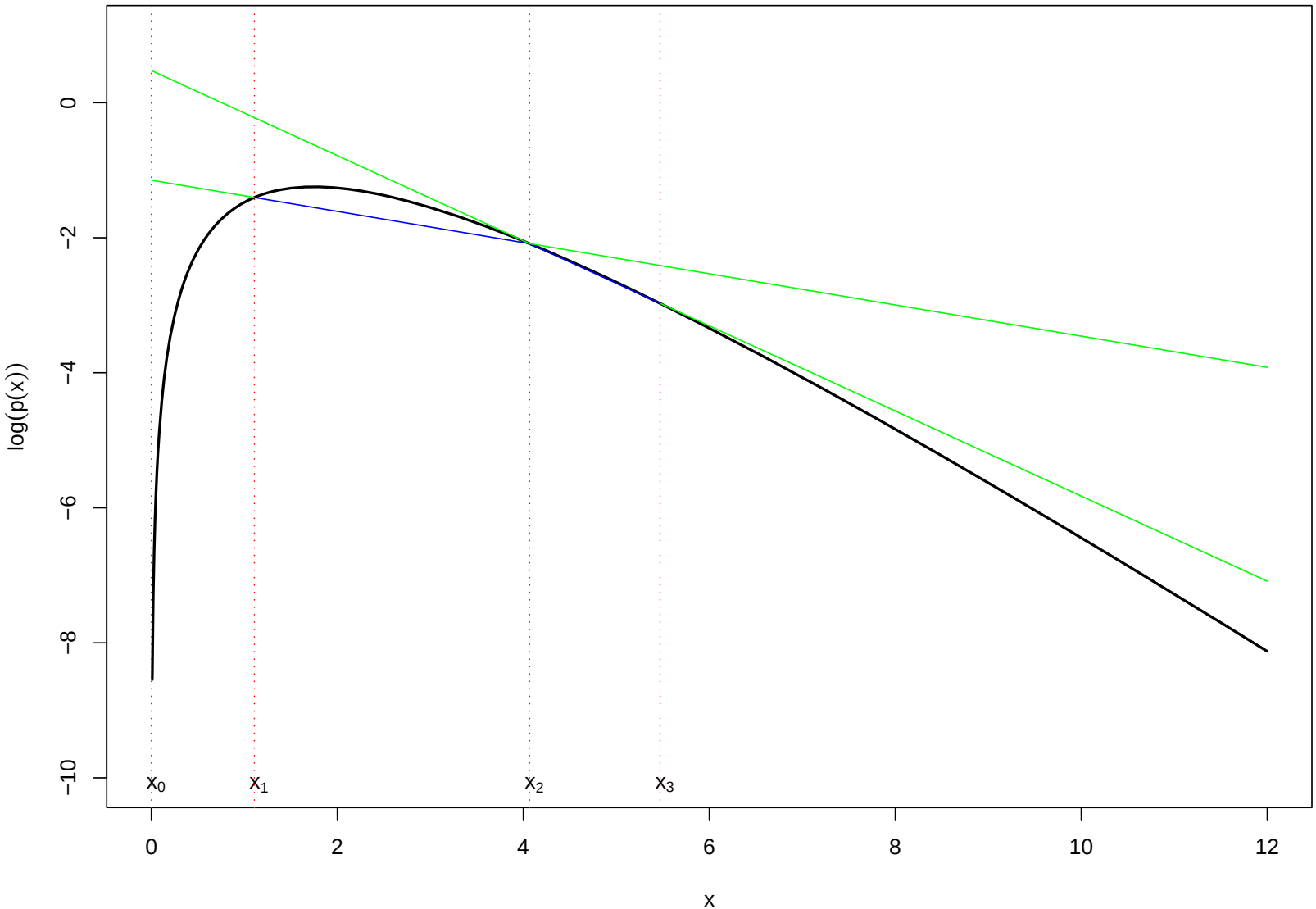
Adaptive Rejection sampling

Upper envelope (part 1):



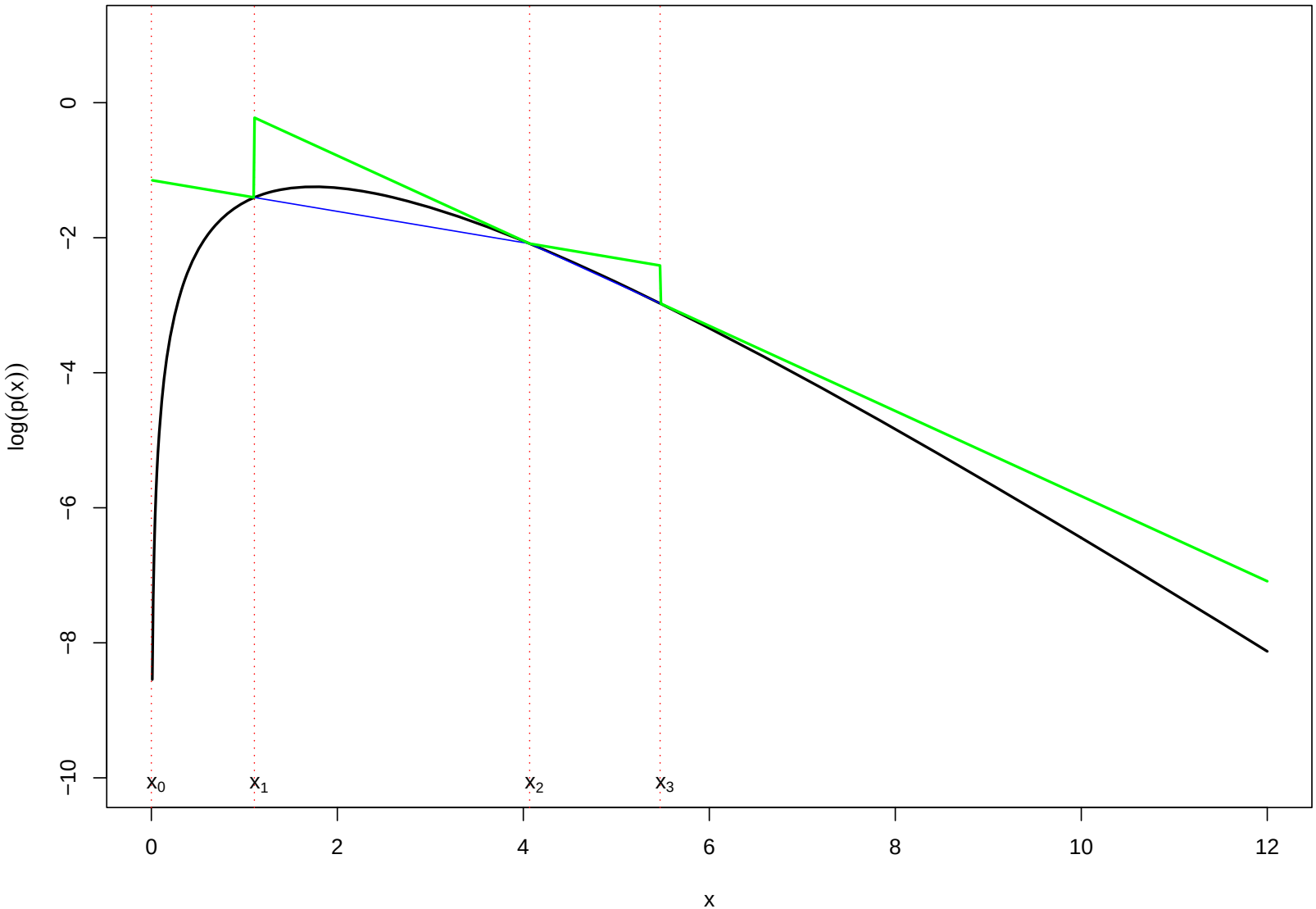
Adaptive Rejection sampling

Upper envelope (part 2):



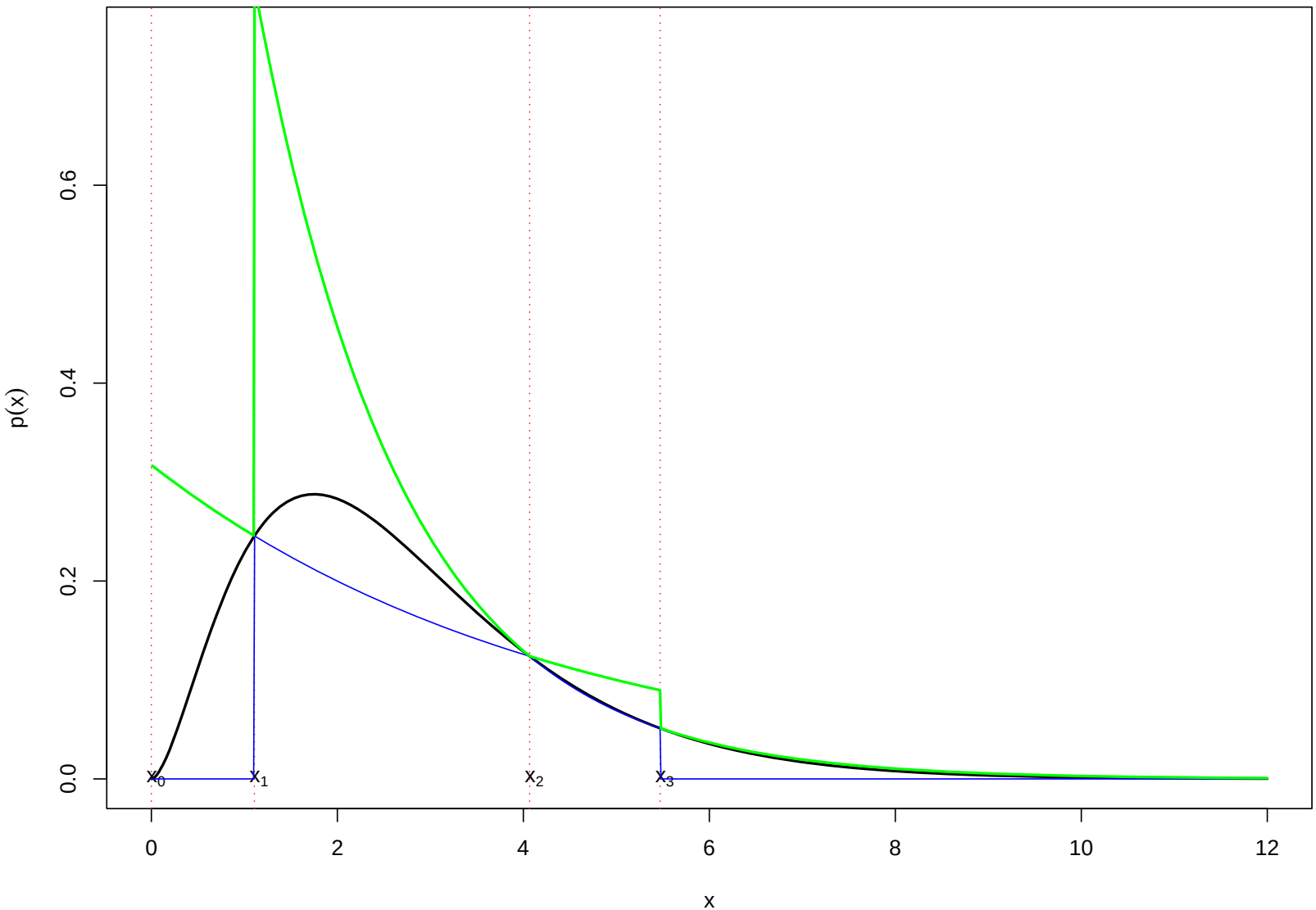
Adaptive Rejection sampling

Lower and Upper envelopes:



Adaptive Rejection sampling

Lower and Upper envelopes: original scale:



Adaptive Rejection sampling

This $p_{0n}(x)$ can be sampled directly by cdf inversion. The algorithm is therefore

1. Generate x from p_{n0}, and u from $Uniform(0, 1)$
2. Accept x if $u \leqslant \exp\{l_n(x)\}/(M_n p_{0n}(x))$, and go to 4.
3. Accept x if $u \leqslant p(x)/(M_n p_{0n}(x))$ and go to 4.
4. Update p_{n0} by updating the lower and upper envelopes taking into account the x and the function value $p(x)$.
5. Return to 1.

Every time a point is accepted or rejected, the envelopes are potentially updated; the updating only changes the envelopes locally, the intervals surrounding the location of the rejected points.

Gibbs Sampler: example

Weibull posterior distribution

Suppose that $Y_1, \ldots, Y_n$ are i.i.d. from the Weibull distribution with density

$$f_Y(y; \gamma_0, \lambda_0) = \gamma_0 \lambda_0 y^{\gamma_0 - 1} \exp\{-\lambda_0 y^{\gamma_0}\} \qquad y > 0$$

and zero otherwise, for parameters $\gamma_0, \lambda_0 > 0$. We seek to perform Bayesian inference for the two unknown parameters.

The likelihood function for observed data $y_1, \ldots, y_n$ takes the form

$$\mathcal{L}(\gamma_0, \lambda_0; \mathbf{y}) = \gamma_0^n \lambda_0^n \left(\prod_{i=1}^n y_i \right)^{\gamma_0 - 1} \exp\left\{ -\lambda_0 \sum_{i=1}^n y_i^{\gamma_0} \right\}$$

Gibbs Sampler: example

Weibull posterior distribution

We assume independent $Exponential(0.1)$ priors for γ_0 and λ_0

$$p(\gamma_0, \lambda_0) = 0.01e^{-0.1(\gamma_0+\lambda_0)} \qquad \gamma_0, \lambda_0 > 0.$$

This yields the posterior distribution up to proportionality as

$$\pi(\gamma_0, \lambda_0|\mathbf{y}) \propto \mathcal{L}(\gamma_0, \lambda_0; \mathbf{y}) p(\gamma_0, \lambda_0) \qquad \gamma_0, \lambda_0 > 0$$

which is a non-standard distribution.

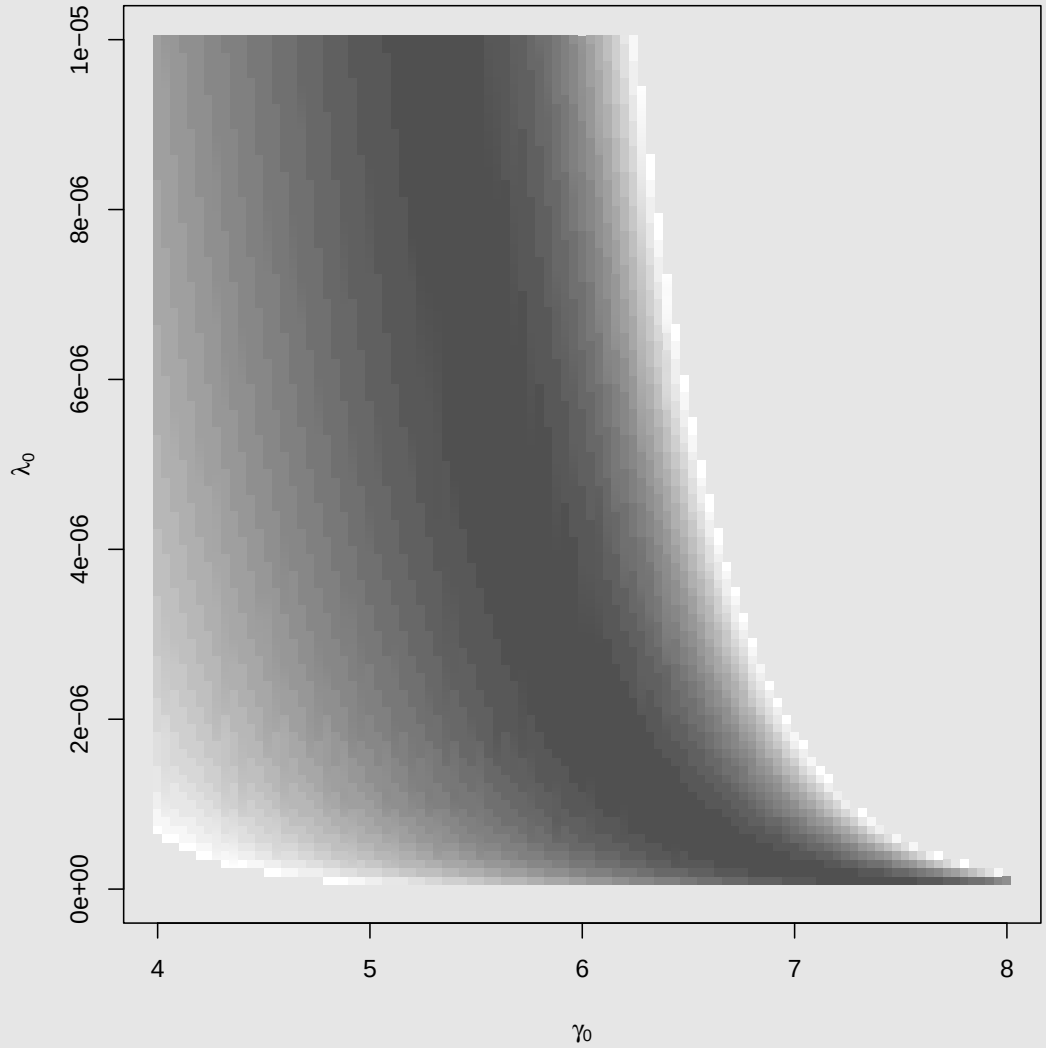
We seek to produce a sample from this joint posterior distribution for data ($n = 15$).

10.3959 6.2281 6.5331 10.7086 7.6138 8.9423 8.8254 6.1461
7.2988 8.8081 7.5316 8.2238 8.9831 6.4174 9.7648

Gibbs Sampler: example

Weibull example

Joint posterior (up to proportionality)



Gibbs Sampler: example

Weibull example

In this case, neither full conditional posterior

$$\pi(\gamma_0|\lambda_0, \mathbf{y}) \qquad\qquad \pi(\lambda_0|\gamma_0, \mathbf{y})$$

is a standard distribution, and cannot be sampled from easily.

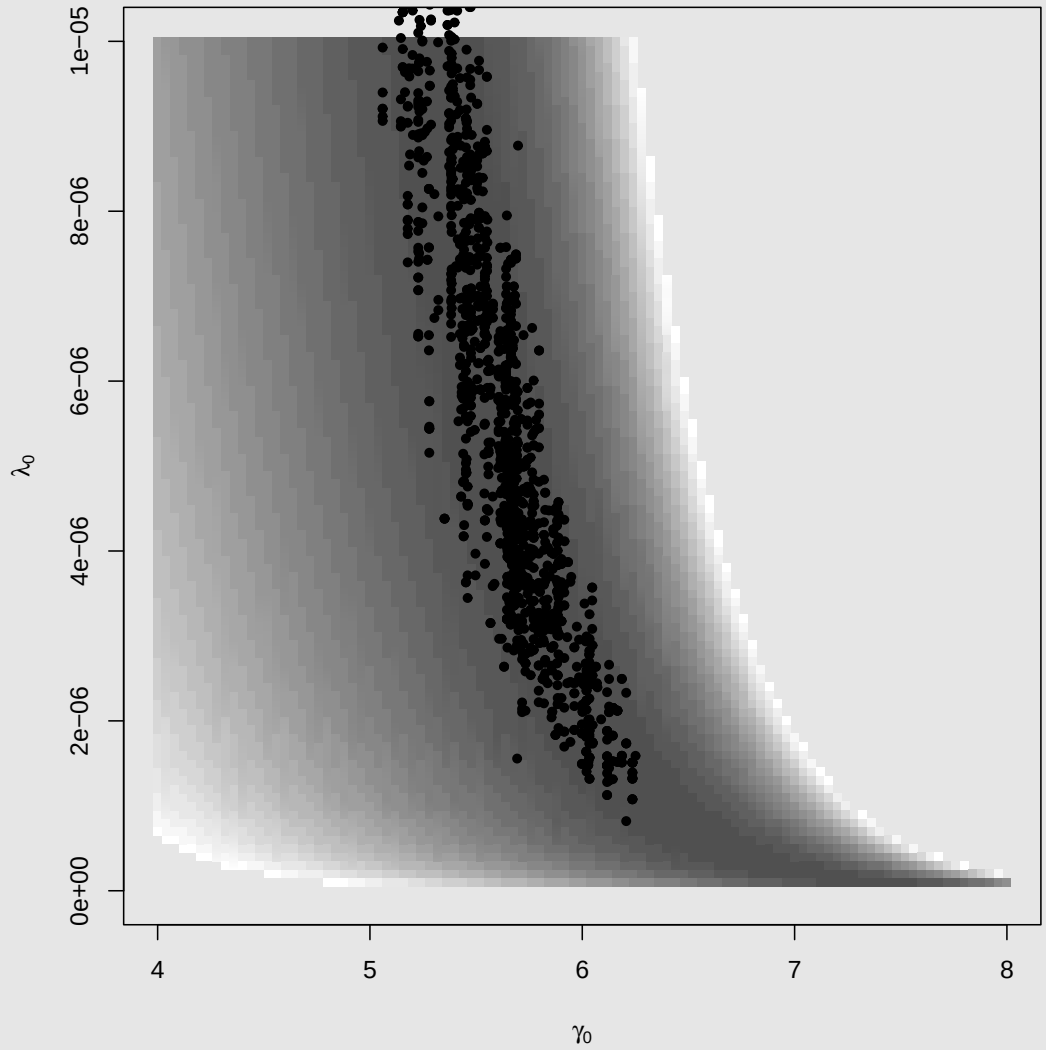
We adopt a *Metropolis-within-Gibbs* strategy, using Metropolis-Hastings accept-reject steps for each parameter and its full conditional.

Specifically, as both parameters are positive, we use the reflected normal proposal distribution from the previous example with $\sigma_q = 1$ for γ_0 and $\sigma_q = 10^{-3}$ for λ_0 proposals.

Gibbs Sampler: example

Weibull example

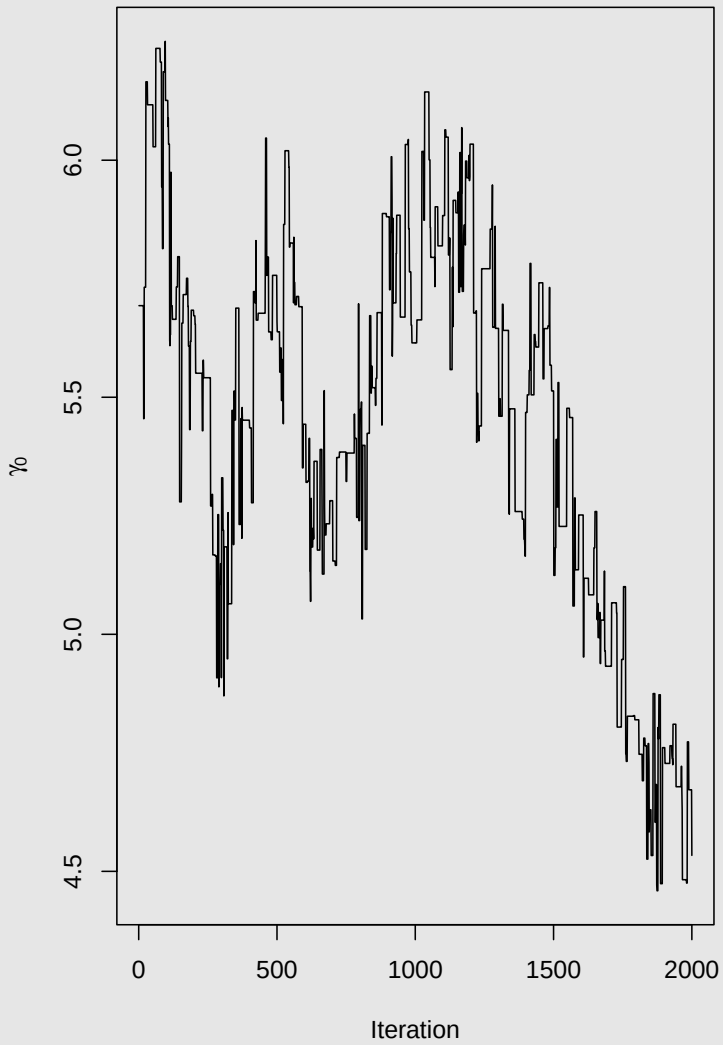
Sample from joint posterior from first 2000 Gibbs sampler steps.

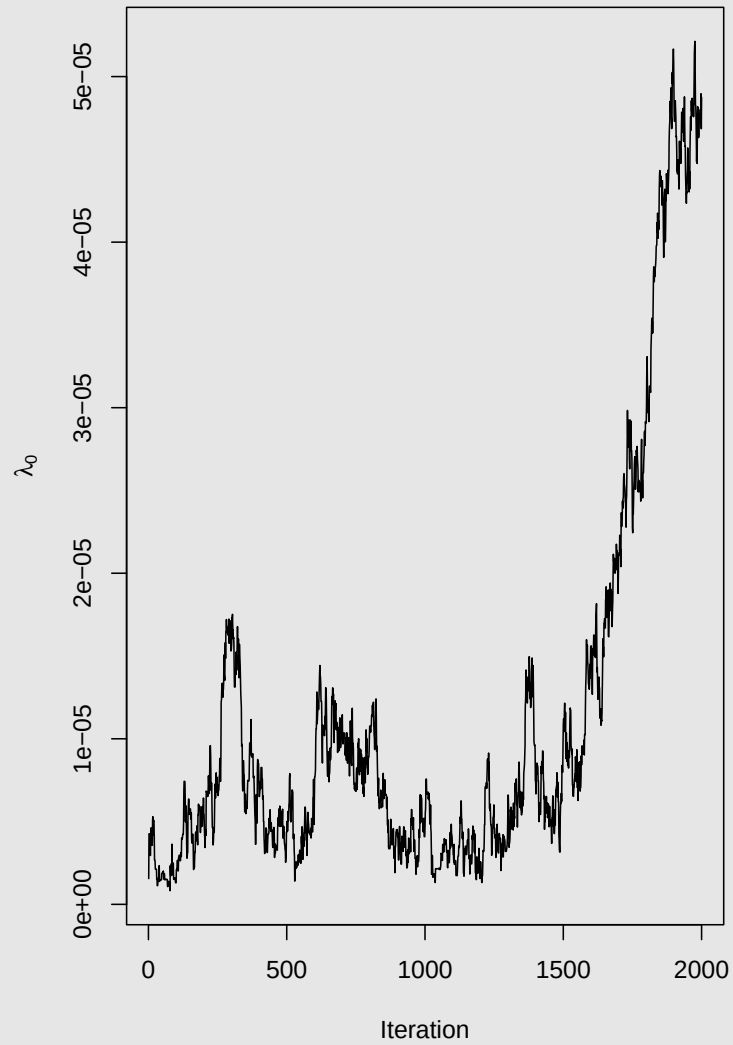


Gibbs Sampler: example

Weibull example

Trace plots:

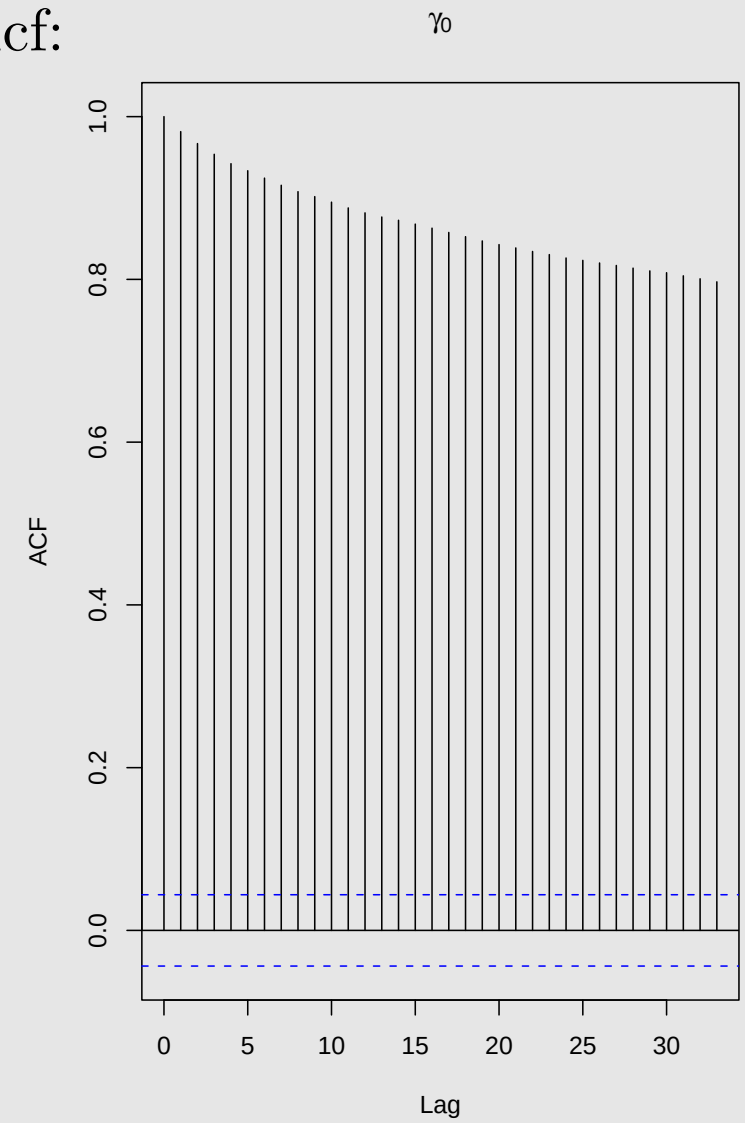


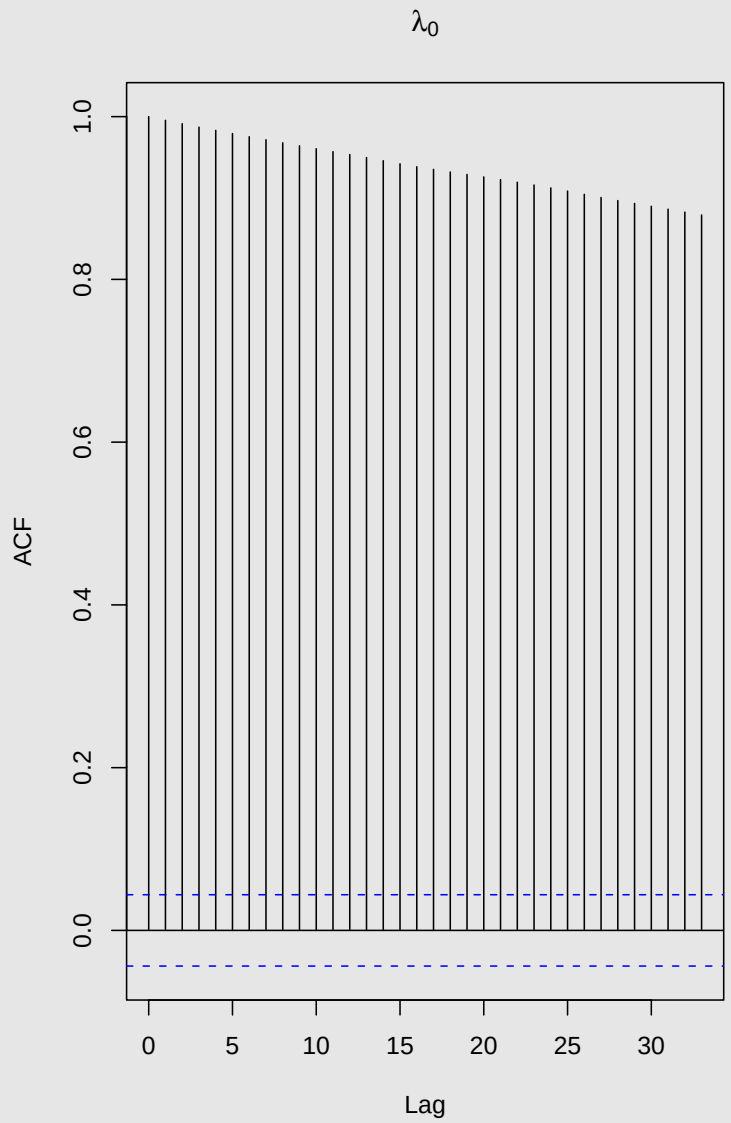


Gibbs Sampler: example

Weibull example

Acf:





Gibbs Sampler: example

Weibull example

The posterior correlation here is approximately 0.86; this severely affects the performance of the Gibbs sampler.

An alternative parameterization partially solves this problem. Let $\gamma = \gamma_0$, but

$$\lambda = \left(\frac{1}{\lambda_0}\right)^{1/\gamma} \qquad \lambda_0 = \left(\frac{1}{\lambda}\right)^{\gamma}$$

For this new parameterization, we must remember to include the Jacobian in the prior for the new parameters

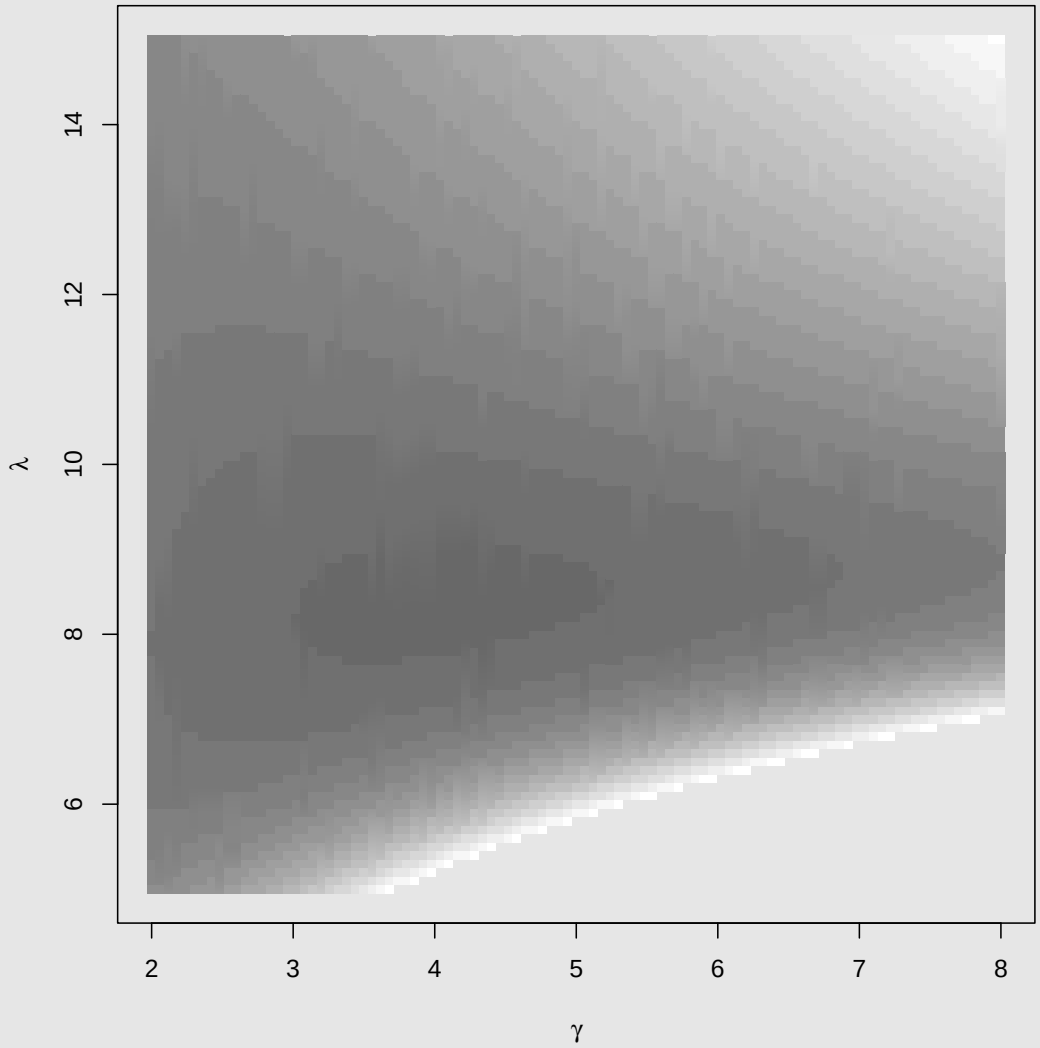
$$p(\gamma, \lambda) = p(\gamma, \lambda_0(\gamma, \lambda))|J(\gamma, \lambda)|$$

We again must use Metropolis-within-Gibbs.

Gibbs Sampler: example

Weibull example

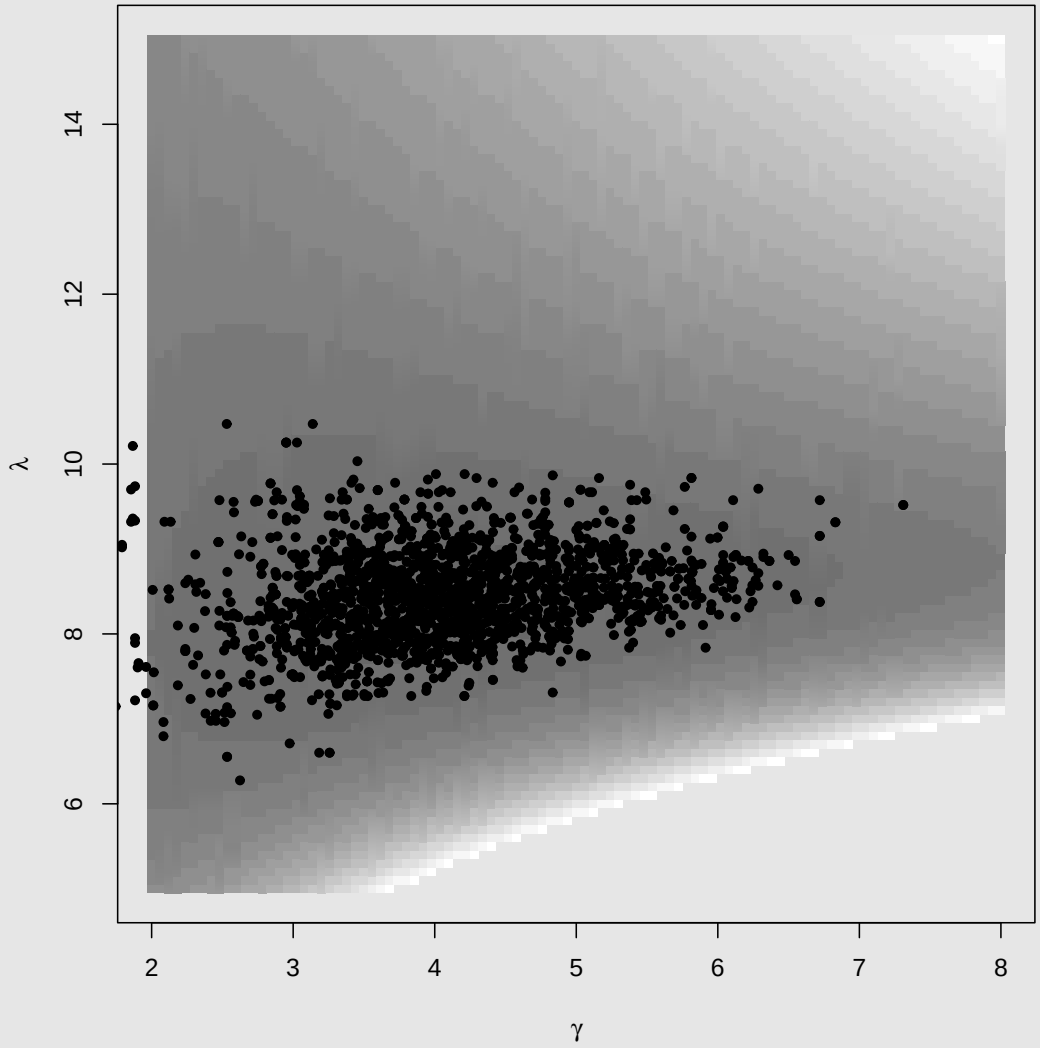
Joint posterior (up to proportionality) for new parameterization



Gibbs Sampler: example

Weibull example

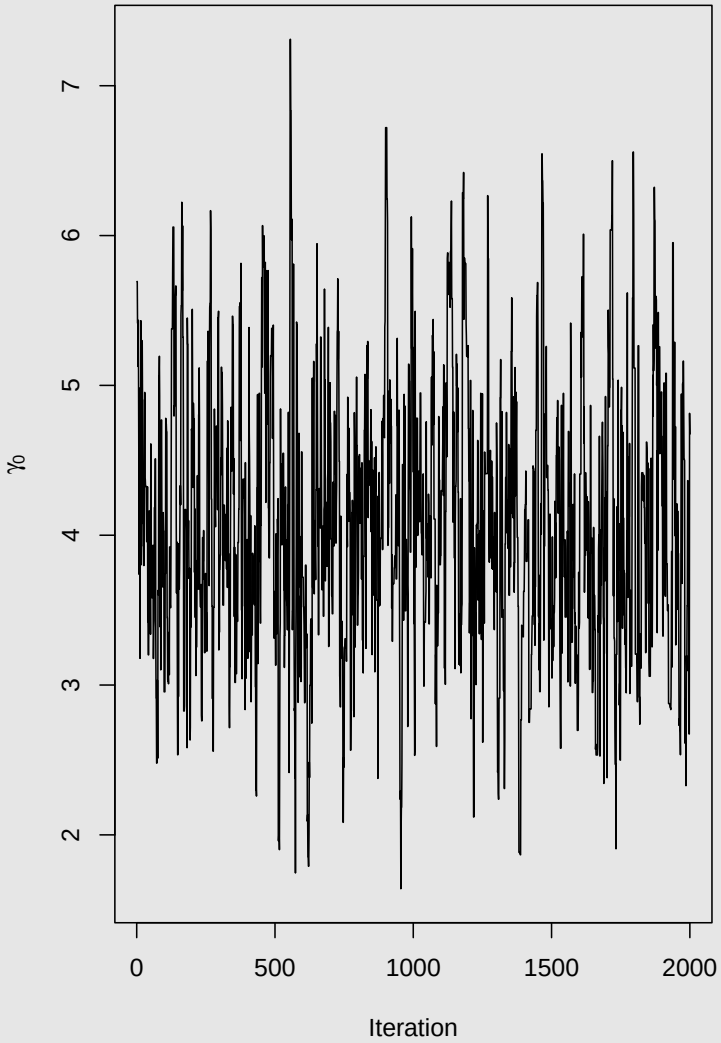
Sample from joint posterior for new parameterization

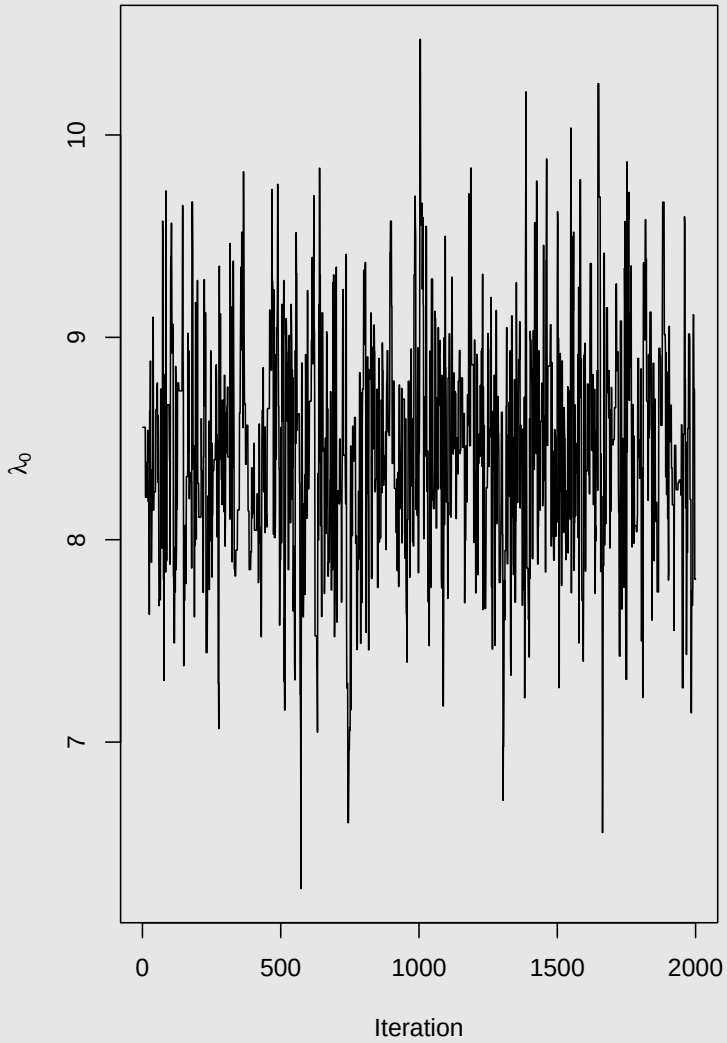


Gibbs Sampler: example

Weibull example

Trace plots: new parameterization

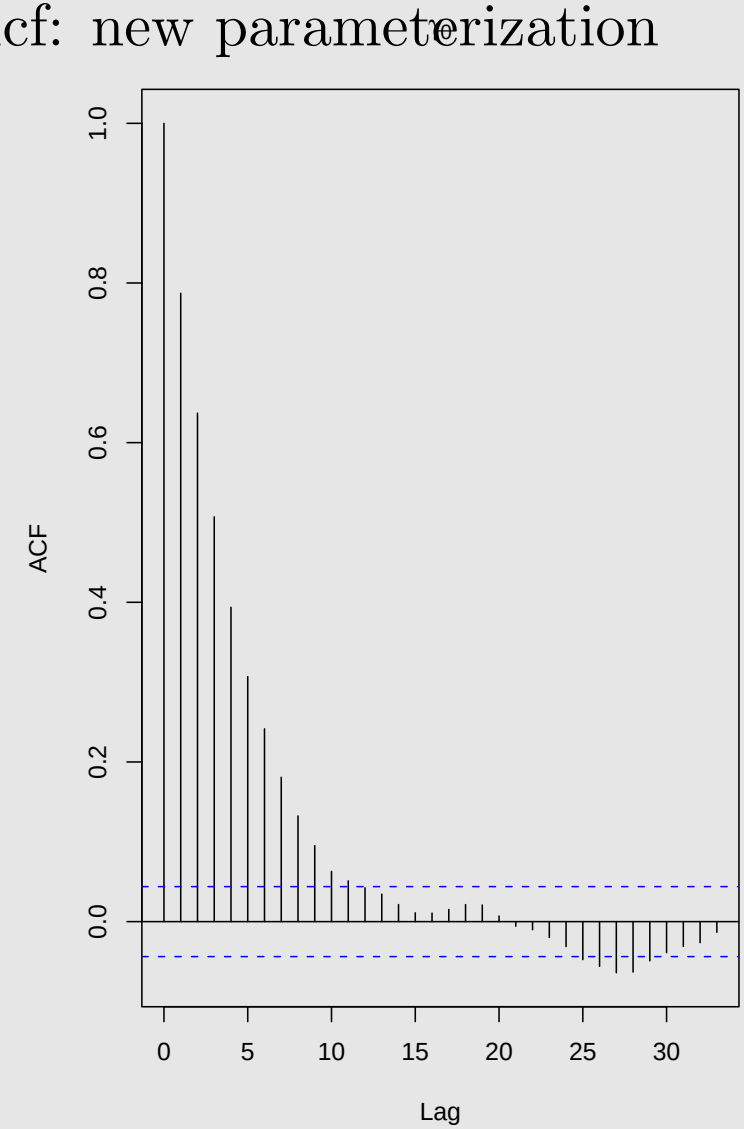


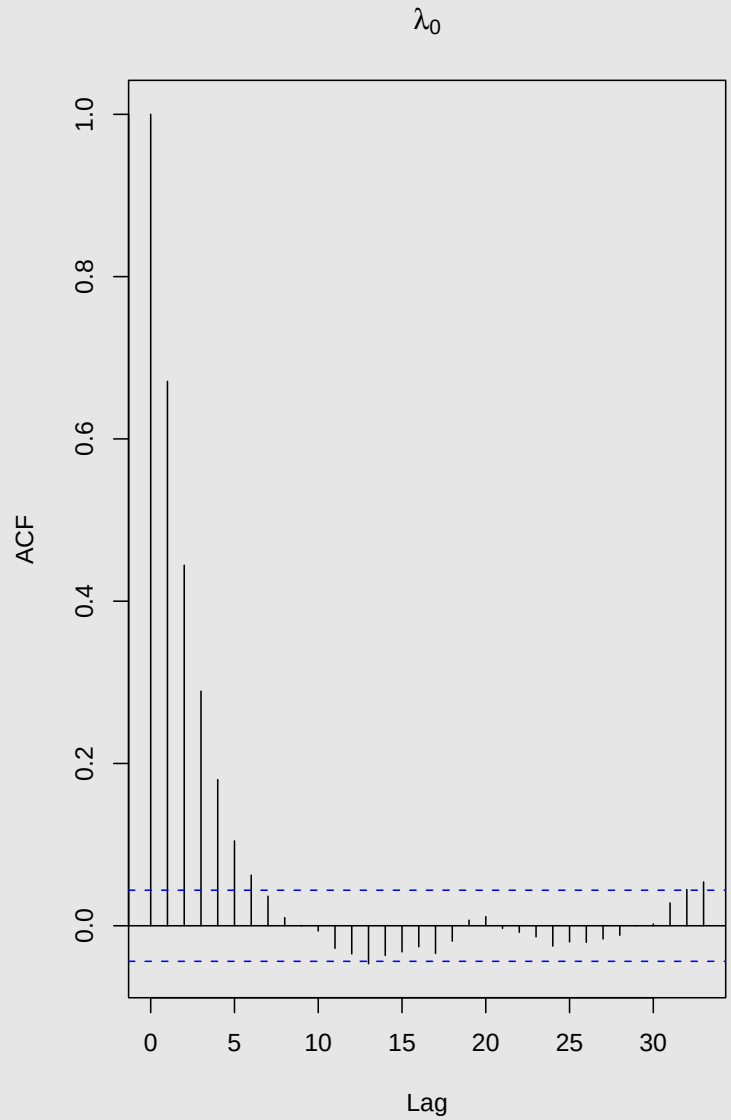


Gibbs Sampler: example

Weibull example

Acf: new parameterization





Gibbs Sampler: example

Weibull example

The posterior correlation here is approximately 0.25, and the Gibbs sampler can effectively traverse the parameter space.

Parameter estimates for the new parameters can be obtained from the posterior samples: the mean and 95% credible interval for each parameter is

$$\begin{aligned} \gamma &: 4.079\,(2.455, 6.038) \\ \lambda &: 8.446\,(7.310, 9.593) \end{aligned}$$

It is also possible to obtain posterior summaries for other functions of the posterior parameters.

Gibbs Sampler: example

Weibull example

For example, the survivor function, $S(y)$, is defined by

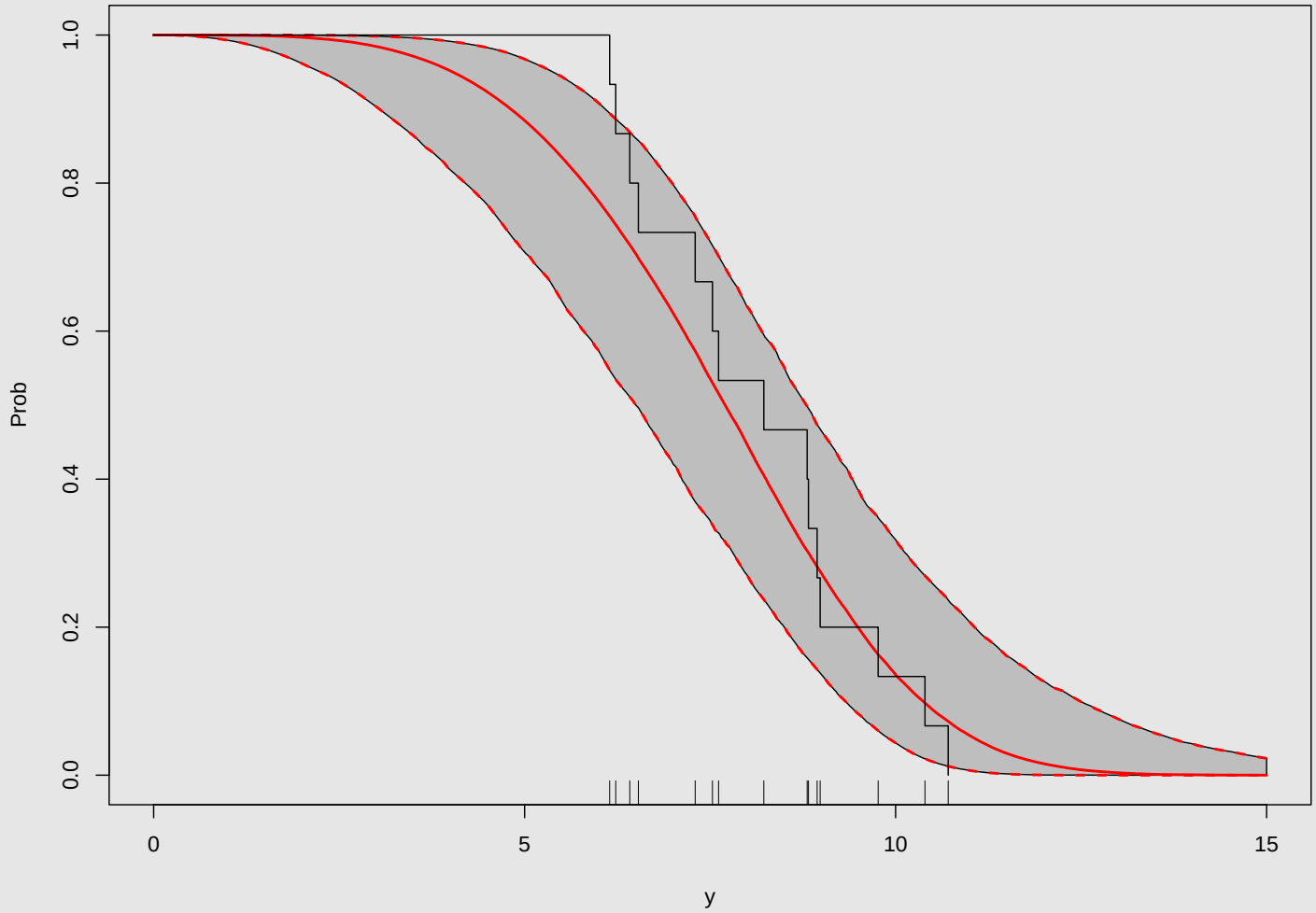
$$S(y) = \Pr[Y > y] = \exp\{-(y/\lambda)^{\gamma}\}.$$

For each $y \in \mathbb{R}^{+}$, we can compute this function for each sampled pair (γ, λ) obtained from the Gibbs sampler. We can then compute the pointwise credible intervals.

Gibbs Sampler: example

Weibull example

Posterior survivor function: Bayes estimate and 95 % credible interval (shaded). Solid line is empirical survivor function.



Effective sample size

We seek to measure the adequacy of the collected samples for estimating parameters, in particular, we wish to assess the variance of the estimators.

For an i.i.d. sample of size N, the variance of the Monte Carlo estimator $\widehat{I}_N(g)$ is

$$\frac{\mathrm{Var}[g(X)]}{N}.$$

However, for a **dependent** sample, the variance is

$$\frac{\mathrm{Var}[g(X)]}{N_{\mathrm{eff}}}$$

where N_{eff} is the *effective sample size.*

Effective sample size

We have that, for a series of N observations from a dependent stochastic process, the effective sample size is given by

$$N_{\text{eff}} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho(k)}$$

where $\rho(k)$ is the true lag-k autocorrelation for the Markov chain. The denominator is termed the *integrated autocorrelation time*.

The true autocorrelations are typically not known, so must be estimated from the data. Most typically this is achieved using spectral methods. The calculation is available in `R` from the library `coda`, in the function

`effectiveSize`.

Effective sample size

For the Weibull example, with $N = 2000$ in runs shown above

	N_{eff}	
	γ	λ
(γ_0, λ_0) parameterization	8.46	0.00
(γ, λ) parameterization	217.20	393.69

From this we can tell that the second MCMC run, in the (γ, λ) parameterization, has produced much larger effective sample sizes.

The value of zero (highlighted) is the result of apparent lack of convergence.

Rejection sampling for the Weibull example

Note that we could attempt to address the problem of sampling from the posterior distribution in the Weibull problem above by *rejection sampling*. In the (γ, λ) parameterization, we have

$$\pi(\gamma, \lambda|\mathbf{y}) \propto \mathcal{L}(\gamma, \lambda; \mathbf{y}) p(\gamma, \lambda_0(\gamma, \lambda))|J(\gamma, \lambda)|$$

where $p(.,.)$ is the product of independent $Exponential(0.01)$ priors.

We can use proposal function p_0 which is the product of $Gamma$ densities; we choose

$$Gamma(2, 1/2) \qquad Gamma(4, 2)$$

for proposing (γ, λ), and then use numerical maximization to find M.

Rejection sampling for the Weibull example

Weibull example

Rejection sampling: acceptance rate is approximately 0.116.

