Part IV

Monte Carlo Methods

Monte Carlo: basic principles

The basic principles of Monte Carlo are based on simple ideas from frequentist statistics:

- **Laws of large numbers:** Suppose $X_1, \ldots, X_n, \ldots$ are independent, identically distributed random variables. Then, usually, as $n \longrightarrow \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \xrightarrow[\text{p}]{\text{a.s.}} \mathbb{E}[g(X)]$$

- **Ergodic Theorems:** Under mild conditions on the joint distribution of random variables $X_1, \ldots, X_n, \ldots$, as $n \longrightarrow \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \xrightarrow[\text{p}]{\text{a.s.}} \mathbb{E}[g(X)]$$

Monte Carlo: basic principles

- **Central Limit Theorems:** Under mild conditions on the joint distribution of random variables $X_1, \ldots, X_n, \ldots$, as $n \longrightarrow \infty$,

$$a_n \left(\frac{1}{n} \sum_{i=1}^{n} g(X_i) - b_n \right) \xrightarrow{d} \mathcal{N}(\mu, \sigma^2)$$

  for suitable choices of the sequences $\{a_n\}$ and $\{b_n\}$.

Essentially, standardized sums of random variables have tractable long-run behaviour. For example, for a distribution with finite second moment

$$\overline{X} \xrightarrow{p} \mathbb{E}[X] \qquad \frac{1}{n} \sum_{i=1}^{n} X_i^2 \xrightarrow{p} \mathbb{E}[X^2]$$

as $n \longrightarrow \infty$, and so on.

Monte Carlo: basic principles

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

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

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

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

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.

We need to establish

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

Monte Carlo: Validity

The strong law of large numbers ensures that

$$\frac{1}{N}\sum_{i=1}^{N} g(X_i) \xrightarrow{a.s.} \mathbb{E}[g(X)] = \int g(x)p(x)\,dx$$

provided the expectation $\mathbb{E}[g(X)]$ exists. Does this ever go wrong ?

Consider computing

$$\int_0^1 \frac{1}{x}\sin(2\pi/x)\,dx$$

by sampling $X_i \sim \mathit{Uniform}(0,1)$, and then computing

$$\frac{1}{N}\sum_{i=1}^{N} \frac{1}{X_i}\sin(2\pi/X_i)$$

Monte Carlo: Validity

The integral can be computed as

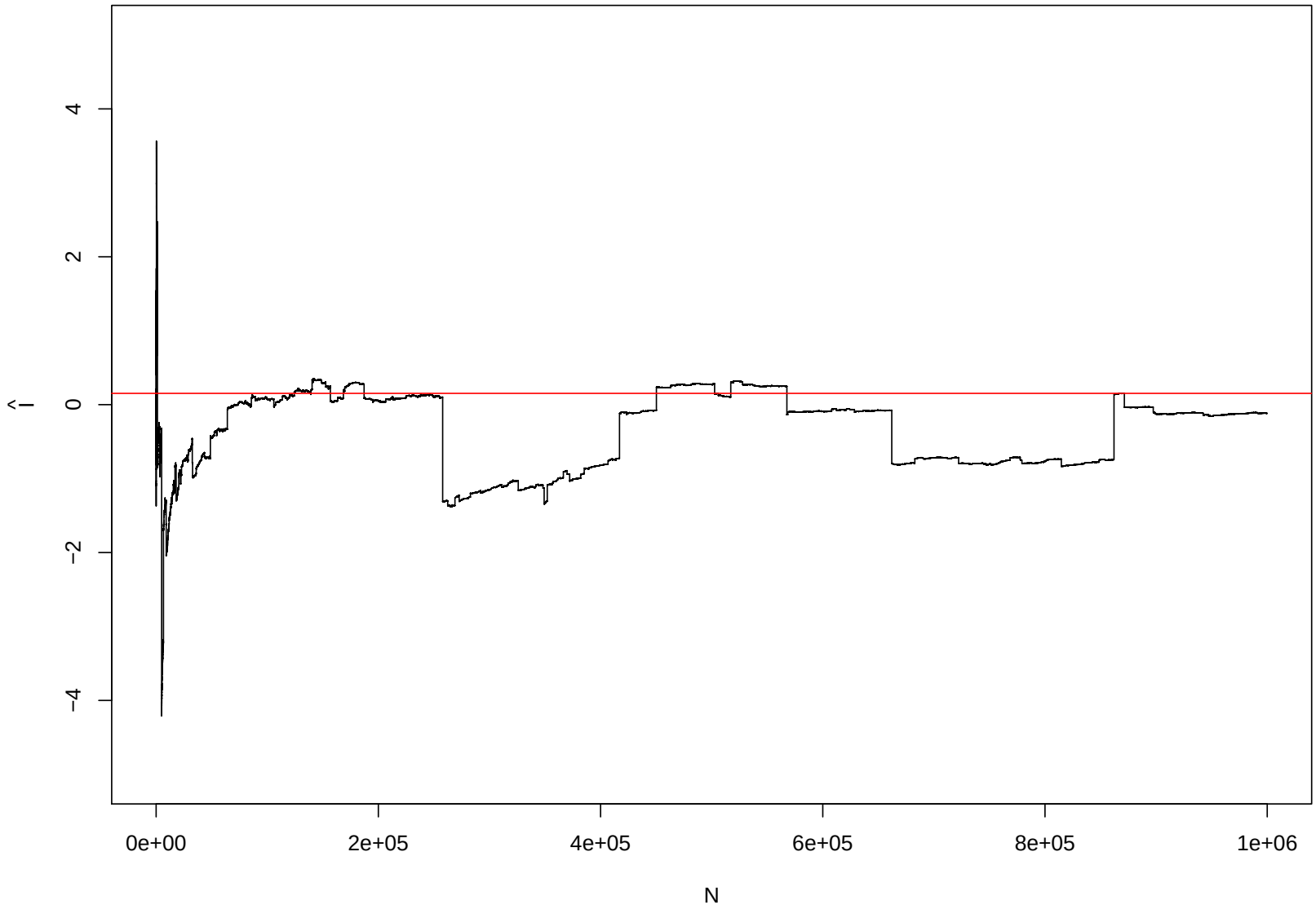
$$\begin{aligned}\int_0^1 \frac{1}{x}\sin(2\pi/x)\,dx &= \int_1^\infty \frac{\sin(2\pi t)}{t}\,dt \\ &= \int_0^\infty \frac{\sin(t)}{t}\,dt - \int_0^{2\pi} \frac{\sin(t)}{t}\,dt \\ &= \mathrm{Si}(\infty) - \mathrm{Si}(2\pi)\end{aligned}$$

where Si(.) is a special function (the *sine integral*) with $\mathrm{Si}(\infty) = \pi/2$.

The numerical value of the integral is 0.1526.

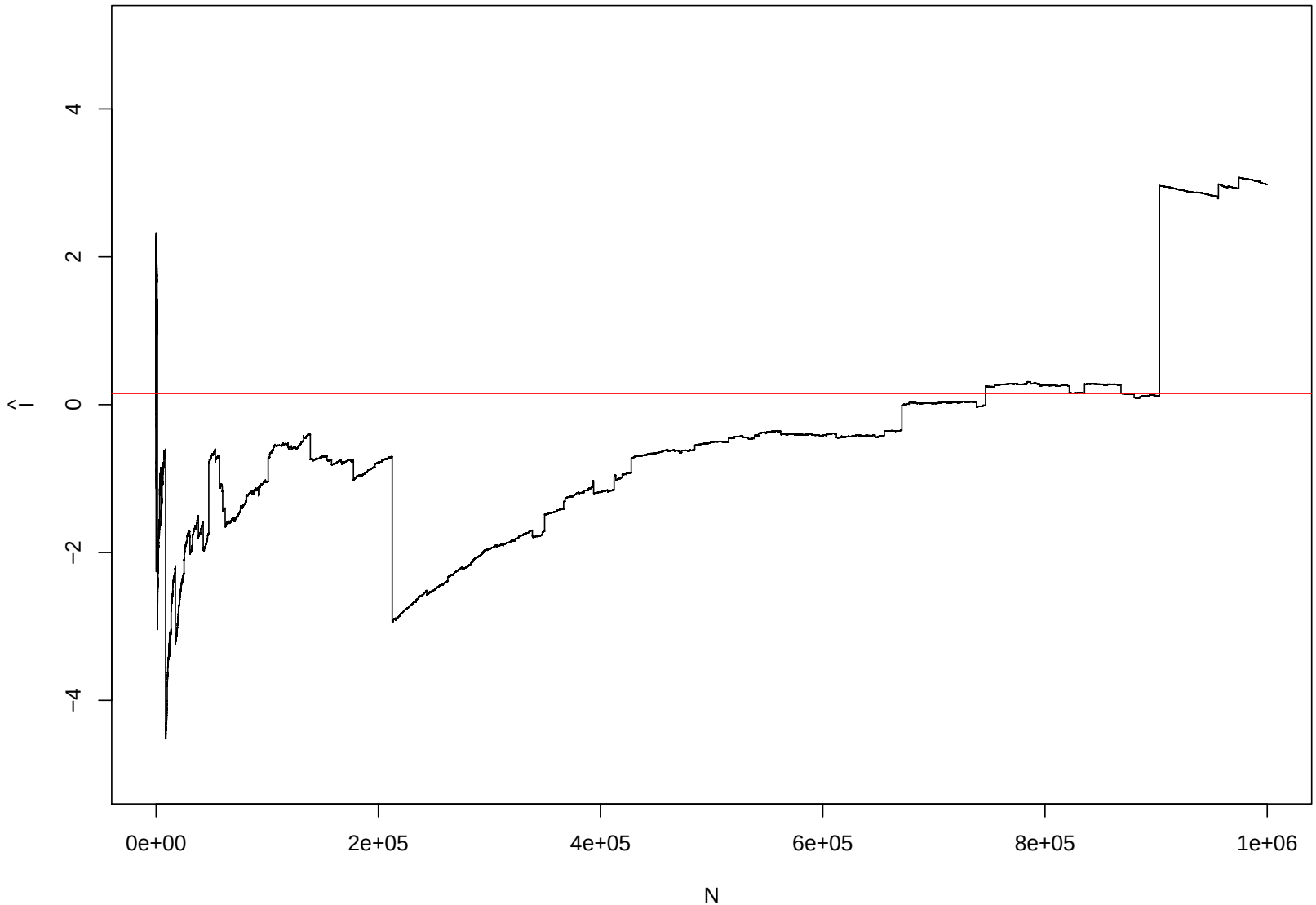
Monte Carlo: Validity

Run 1:



Monte Carlo: Validity

Run 2:



Monte Carlo: Validity

Occasional large values of

$$Y_i = \frac{1}{X_i} \sin(2\pi/X_i)$$

cause the Monte Carlo average to not converge as N gets large.

A sufficient condition for strong convergence is

$$\int |g(x)|p(x)\, dx < \infty$$

which does **not** hold here.

Monte Carlo Estimation: Statistical Properties

In situations where the Monte Carlo estimator

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

of $\mu(g) = \mathbb{E}[g(X)]$ does converge satisfactorily, **and** a central limit theorem applies, we have that

$$\sqrt{N}(\widehat{I}_N(g) - \mu(g)) \xrightarrow{d} \mathcal{N}(0, V(g))$$

where

$$V(g) = \mathbb{V}ar[g(X)] = \int (g(x) - \mu(g))^2 \, p(x) \, dx$$

That is, the Monte Carlo estimator exhibits $o_P(\sqrt{N})$ convergence.

Monte Carlo Estimation: Example

Example (Normal Interval Probabilities)

Consider $p(x) = \phi(x)$ (standard Normal pdf), the function

$$g(x) = \begin{cases} 0 & x \leqslant x_0 \\ 1 & x > x_0 \end{cases}$$

for some fixed $x_0 > 0$, and the integral

$$\int_{-\infty}^{\infty} g(x)p(x)\,dx = 1 - \Phi(x_0).$$

The integral is the tail probability from ordinate x_0.

Monte Carlo Estimation: Example

Example (Normal Interval Probabilities)

The Monte Carlo estimator is this case is

$$\widehat{I}_N(g) = \frac{1}{N}\sum_{i=1}^{N} g(X_i) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(x_0,\infty)}(X_i)$$

where $X_1, \ldots, X_N \sim \mathcal{N}(0,1)$, which has $\mathbb{E}[\widehat{I}_N(g)] = 1 - \Phi(x_0)$, and variance

$$\frac{\Phi(x_0)(1-\Phi(x_0))}{N}$$

which **decreases** with **increasing** x_0. The variance estimate is

$$\frac{\widehat{I}_N(g)(1-\widehat{I}_N(g))}{N}$$

Monte Carlo Estimation: Example

Example (Normal Interval Probabilities)

A problem that occurs when x_0 is large is finite sample bias; the probability of obtaining no $X_i > x_0$, and thus an estimate $\widehat{I}_N(g) = 0$, is

$$p_0 = \{\Phi(x_0)\}^N$$

which can remain large even for large N if x_0 is large enough.

In the following table, N_0 is N such that $p_0 = 10^{-6}$:

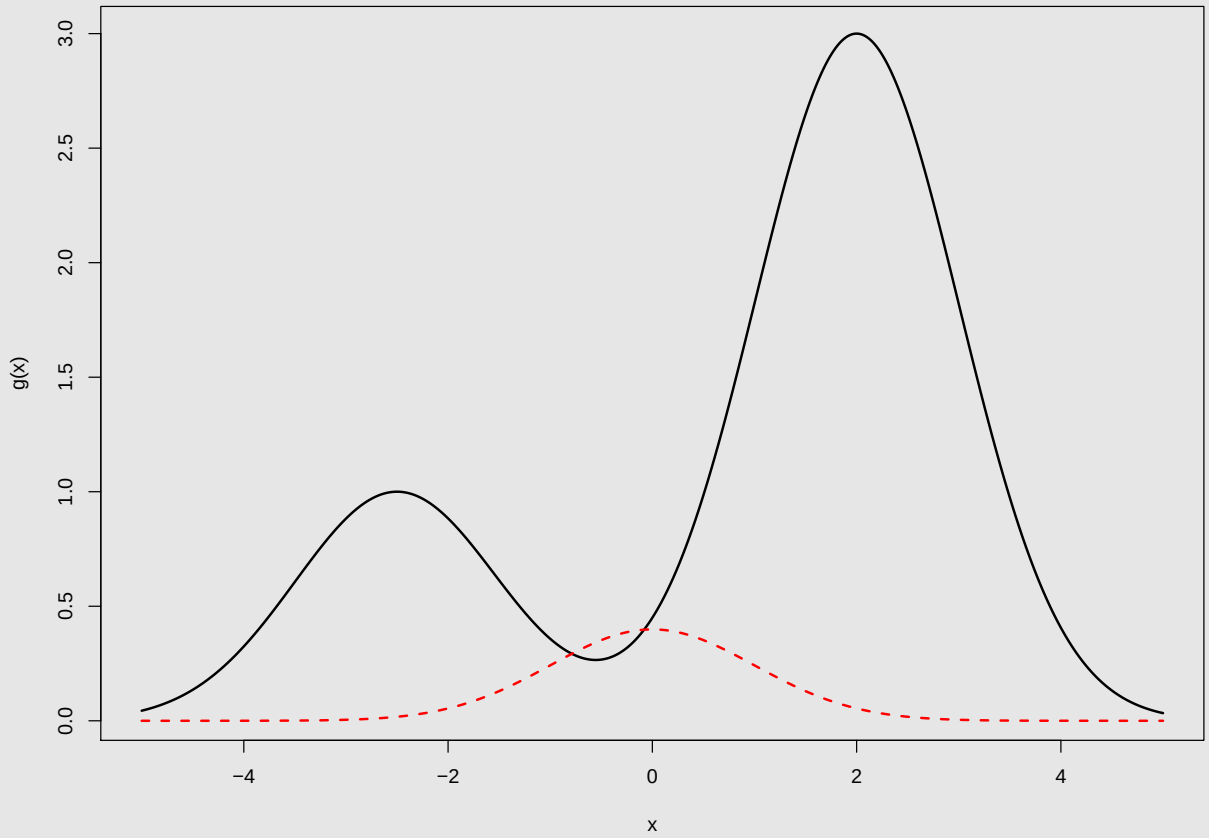
x_0	2	4	6	8	10	12
N_0	6.00×10^2	4.36×10^5	1.40×10^{10}	2.21×10^{16}	1.81×10^{24}	7.77×10^{33}

Monte Carlo Estimation: Example

Example (Mixture Integrand)

Consider

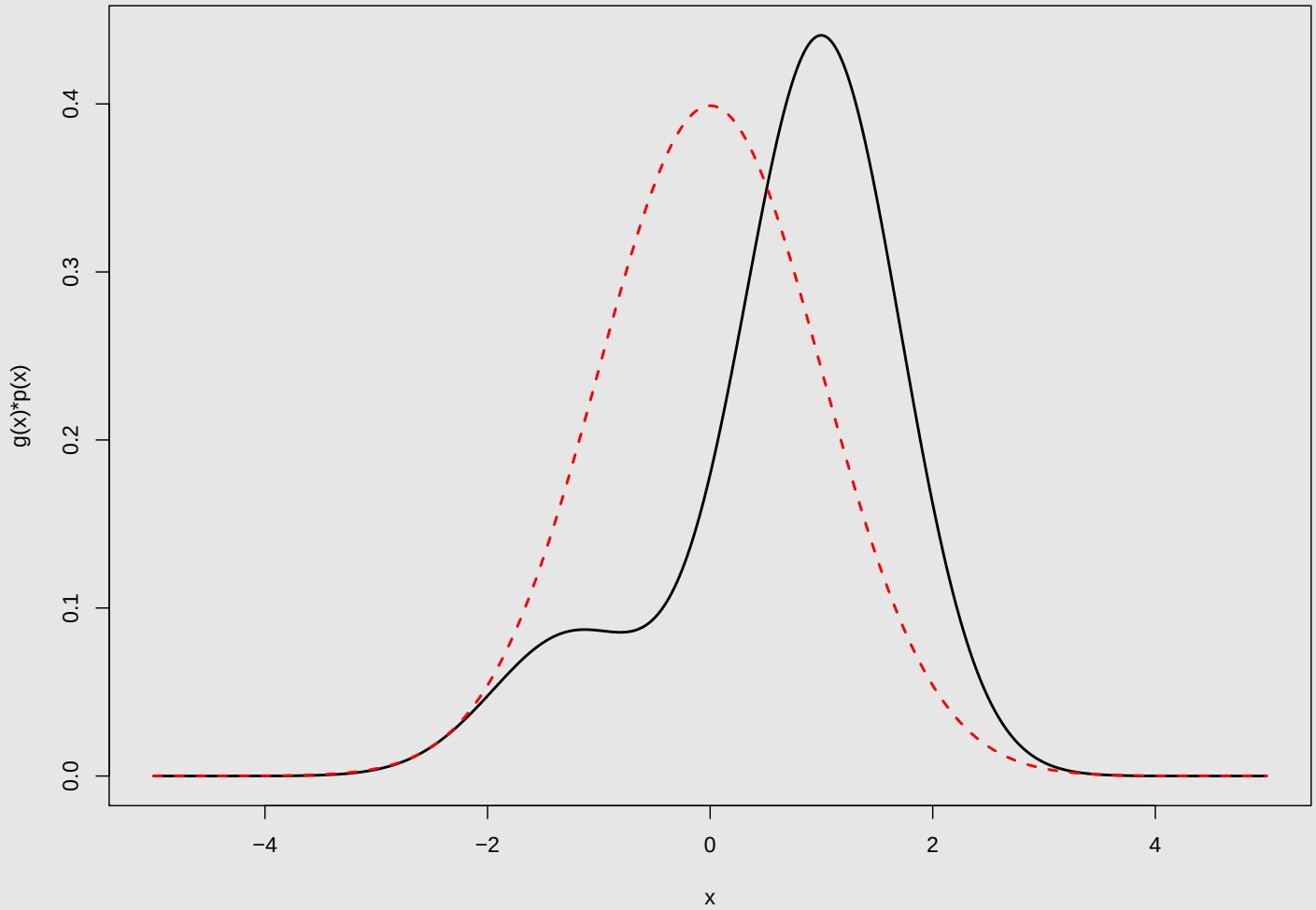
$$g(x) = \exp\left\{-\frac{1}{2}(x+2.5)^2\right\} + 3\exp\left\{-\frac{1}{2}(x-2)^2\right\}$$



Monte Carlo Estimation: Example

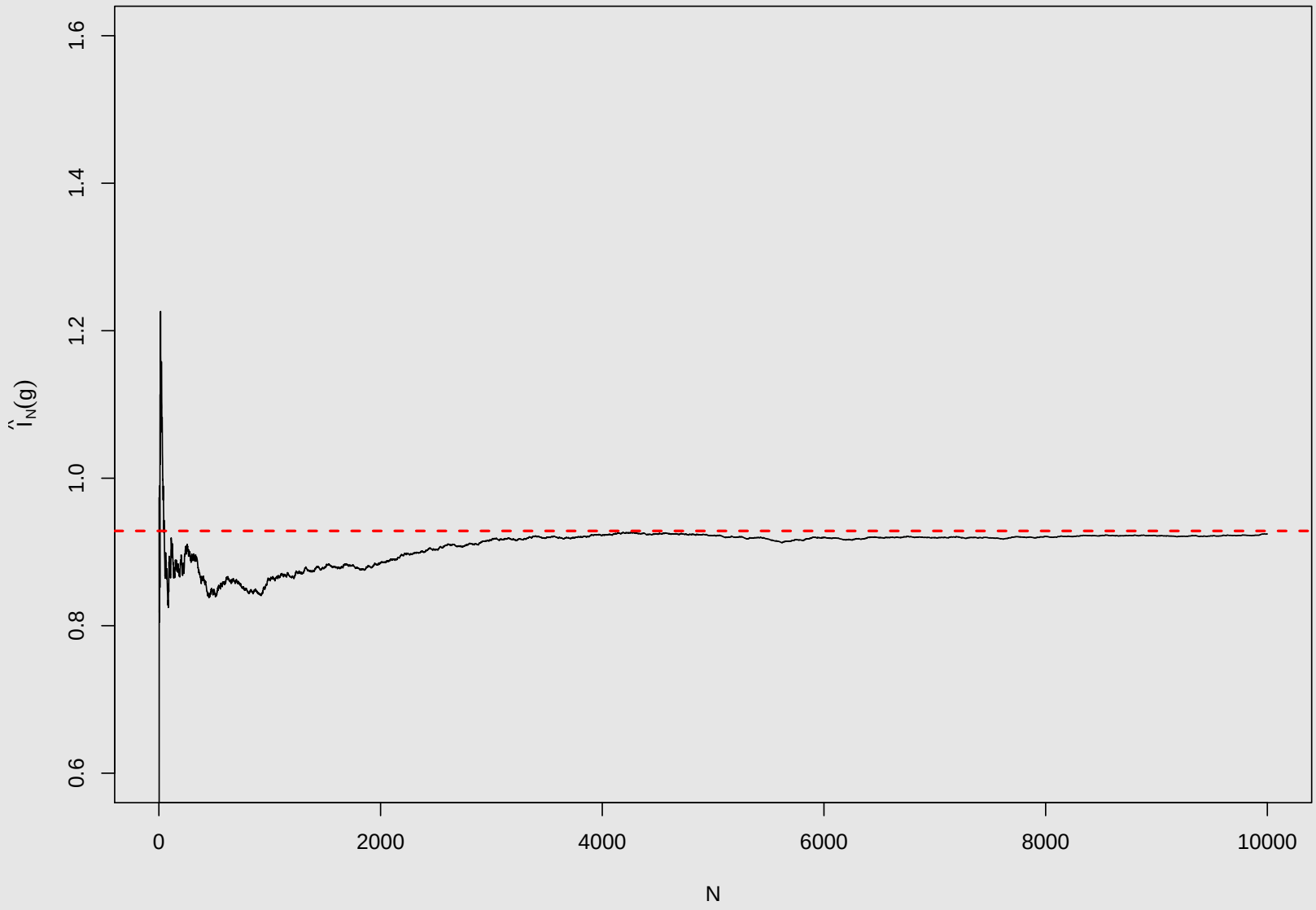
Example (Mixture Integrand)

Plot of integrand $g(x)p(x)$ (solid) and $p(x)$ (dashed):



Monte Carlo Estimation: Example

Example (Mixture Integrand)



Monte Carlo Estimation

Motivated by the deterministic approximation

$$\int g(x)p(x)\,dx \simeq \sum_{j=0}^{k} w_j g(x_j)$$

where weights are determined by $p(x)$, to minimize the error in the approximation, it should be advantageous to choose design points $x_0, \ldots, x_k$ where g is largest.

In a Monte Carlo setting, it seems clear that the estimator will converge more quickly and have lower variance for finite N when $p(x)$ generates points in regions where $g(x)$ is large in magnitude.

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.

Importance sampling: Example

Example (Normal tail probability)

Consider the Monte Carlo estimation of

$$\int_{x_0}^{\infty} \phi(x)\, dx = 1 - \Phi(x_0)$$

For x_0 large, the Monte Carlo estimator is prone to finite sample bias.

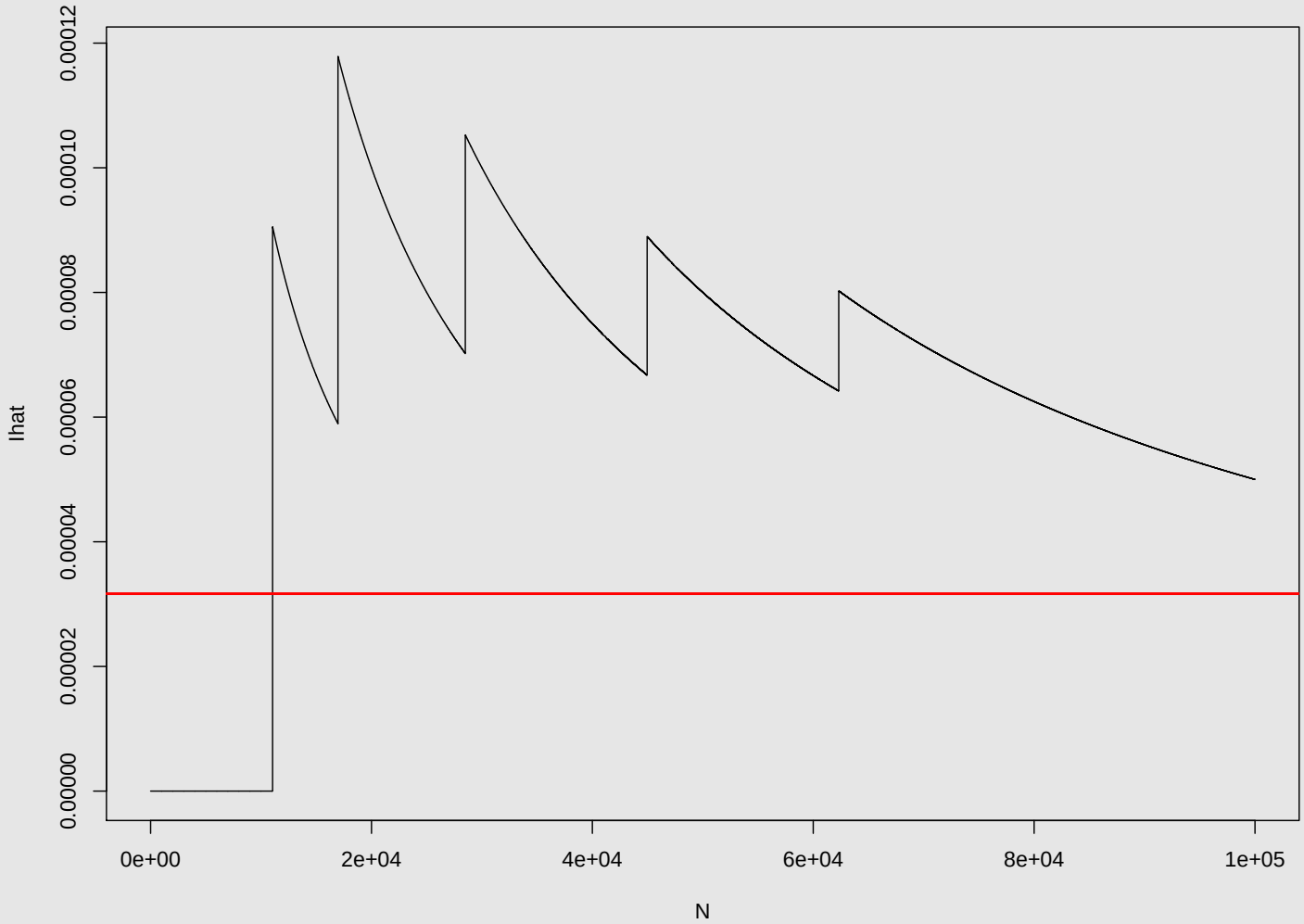
For $x_0 = 4$, $1 - \Phi(x_0) = 3.167 \times 10^{-5}$, and the probability of getting $\widehat{I}_N = 0$ for $N = 100000$ is

$$\{\Phi(x_0)\}^{100000} = 0.0421$$

Importance sampling: Example

Example (Normal tail probability)

Monte Carlo estimate $\widehat{I}_N(g)$ of $1 - \Phi(x_0)$, with $x_0 = 4$:



Importance sampling: Example

Example (Normal tail probability)

Consider using the importance sampling density

$$p_0(x) = \lambda \exp\{-\lambda(x - x_0)\} \qquad x > x_0$$

This density has support identical to the tail region of interest, so in the importance sampling estimator, we have

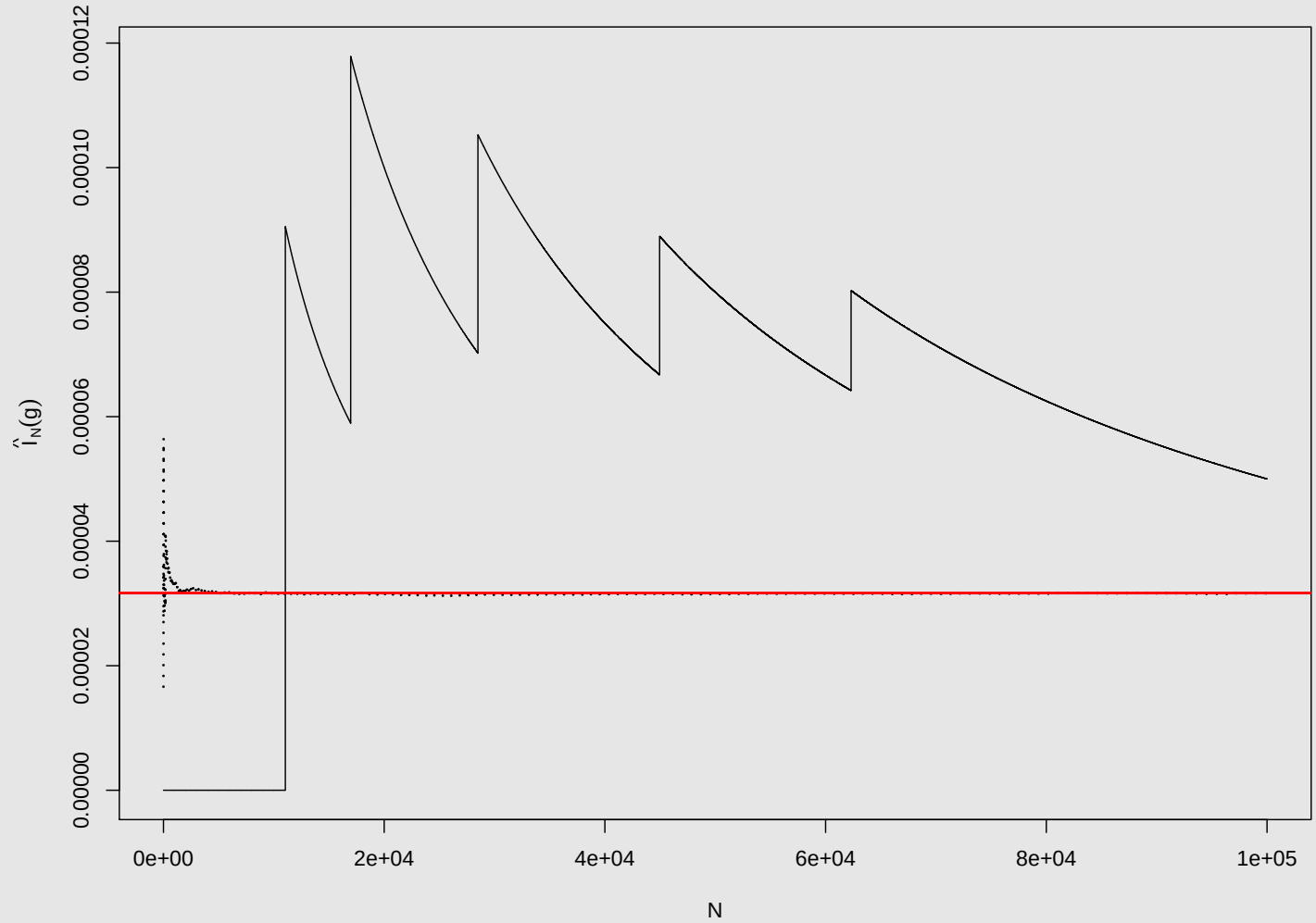
$$\widehat{I}_N^{(p_0)} = \frac{1}{N} \sum_{i=1}^{N} \frac{g(X_i)p(X_i)}{p_0(X_i)} = \frac{1}{N\lambda} \sum_{i=1}^{N} \phi(X_i) \exp\{\lambda(X_i - x_0)\}$$

as $g(X_i) = 1$ for all variates sampled from p_0.

Importance sampling: Example

Example (Normal tail probability)

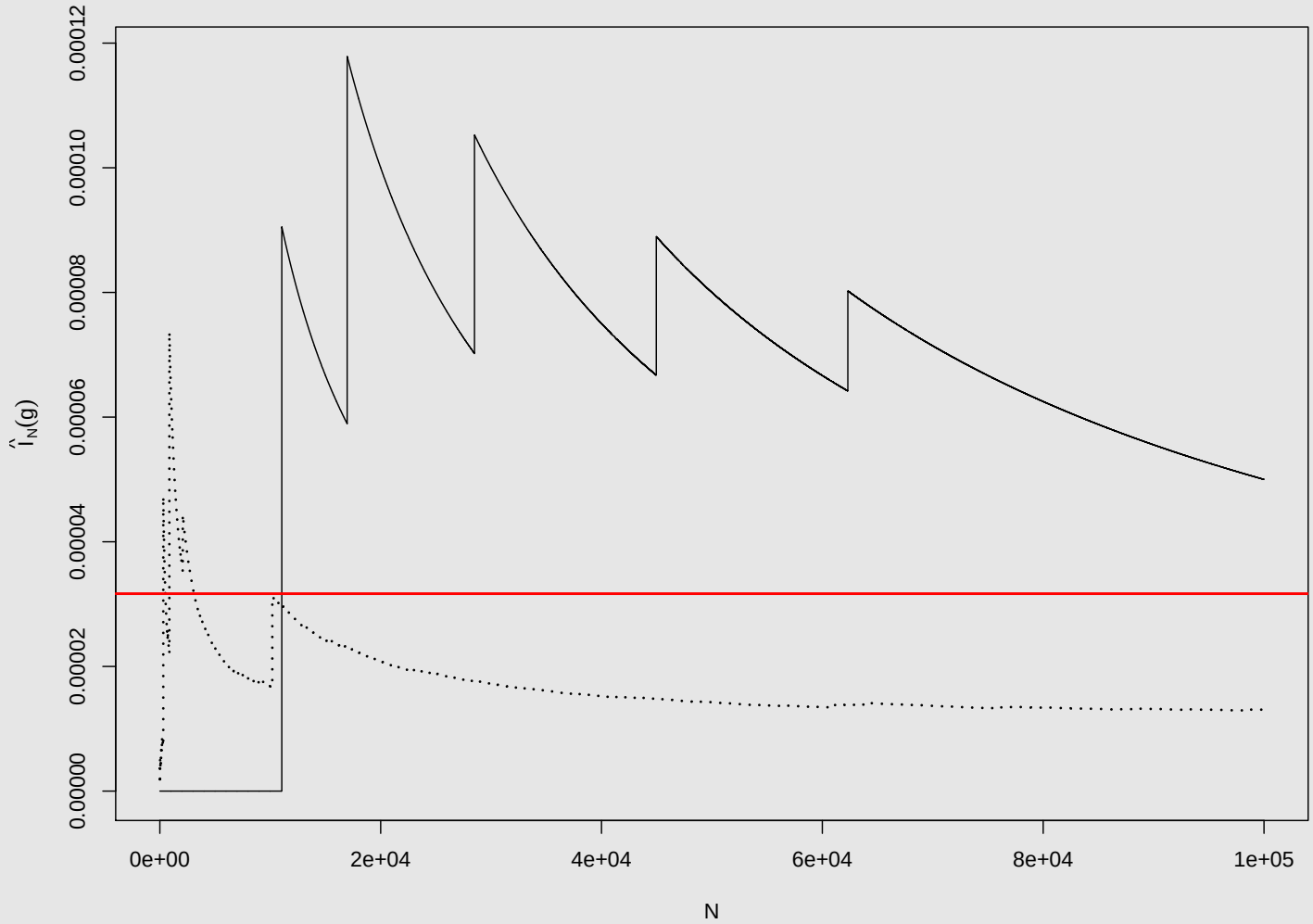
IS estimate $\widehat{I}_N^{(p_0)}(g)$ (dotted) for $\lambda = 10$



Importance sampling: Example

Example (Normal tail probability)

IS estimate $\widehat{I}_N^{(p_0)}(g)$ (dotted) for $\lambda = 100$



Importance sampling: Example

Example (Normal tail probability)

Consider using the importance sampling density $p_0(x) \equiv t(5)$, the Student-t density with five degrees of freedom.

A similar importance sampling estimator can be defined for this density; in this case, the tail-region indicator function

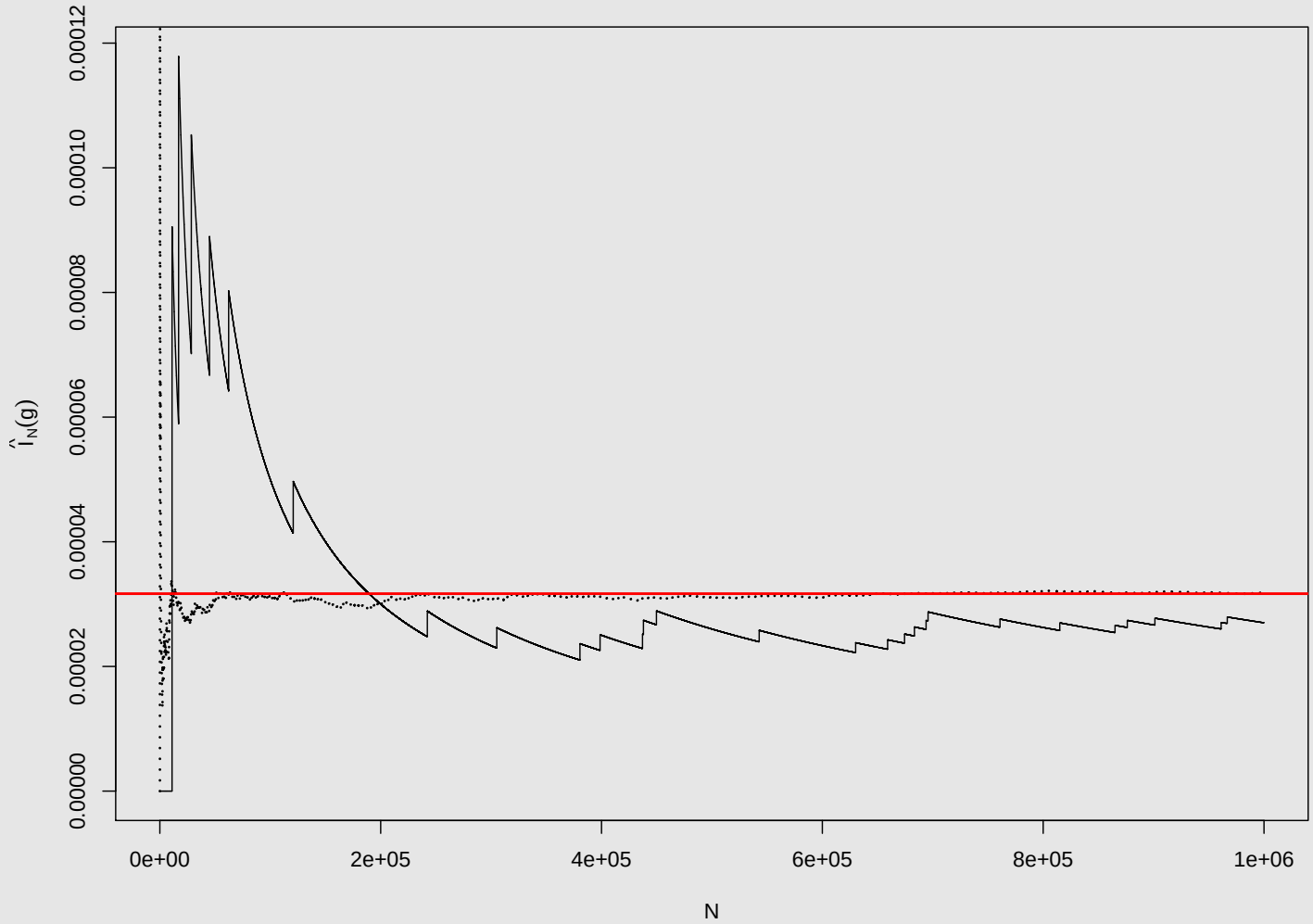
$$g(x) = \begin{cases} 0 & x \leqslant x_0 \\ 1 & x > x_0 \end{cases}$$

must be included in the computation of $\widehat{I}_N^{(p_0)}(g)$.

Importance sampling: Example

Example (Normal tail probability)

IS estimate $\widehat{I}_N^{(p_0)}(g)$ (dotted) for $p_0 \equiv t(5)$



Optimal Importance Sampling

There are many possible choices for the importance sampling density p_0. We now seek guidelines for choosing p_0 optimally.

Note first that the variance of $\widehat{I}_N^{(p_0)}(g)$ is finite if and only if

$$\frac{g(X)p(X)}{p_0(X)}$$

has finite variance, that is, if and only if

$$\mathbb{E}_{p_0}\left[\left\{\frac{g(X)p(X)}{p_0(X)}\right\}^2\right] = \int_{-\infty}^{\infty}\left\{\frac{g(x)p(x)}{p_0(x)}\right\}^2 p_0(x)\,dx$$

is finite.

Optimal Importance Sampling

This equates to

$$\int_{-\infty}^{\infty} \frac{\{g(x)p(x)\}^2}{p_0(x)}\, dx$$

being finite. Note also that

$$\mathbb{E}_{p_0}\left[\left\{\frac{g(X)p(X)}{p_0(X)}\right\}^2\right] = \mathbb{E}_p\left[\{g(X)\}^2\,\frac{p(X)}{p_0(X)}\right]$$

Now, if $p(x)/p_0(x)$ is **unbounded** on the support of p, then these expectations, and the variance of $\widehat{I}_N^{(p_0)}(g)$, are **not finite**.

Therefore, for p with unbounded support, we must ensure that $p(x)/p_0(x)$ stays bounded on $\mathbb{R}$, particularly in the tails.

Optimal Importance Sampling

To optimize the importance sampling procedure, we inspect the variance of the IS estimator. We have that

$$\mathbb{V}ar_{p_0}\left[\frac{g(X)p(X)}{p_0(X)}\right] = \mathbb{E}_{p_0}\left[\left\{\frac{g(X)p(X)}{p_0(X)}\right\}^2\right] - \left\{\mathbb{E}_{p_0}\left[\frac{g(X)p(X)}{p_0(X)}\right]\right\}^2.$$

But

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

does not depend on p_0, so to minimize the variance, we must minimize the first term. Note that

$$\begin{aligned}\mathbb{E}_{p_0}\left[\left\{\frac{g(X)p(X)}{p_0(X)}\right\}^2\right] &\geqslant \left\{\mathbb{E}_{p_0}\left[\frac{|g(X)|p(X)}{p_0(X)}\right]\right\}^2 \\ &= \left\{\int |g(x)|p(x)\,dx\right\}^2\end{aligned}$$

by Jensen's inequality.

Optimal Importance Sampling

This is a lower bound on the variance of the estimator which is independent of p_0. By the logic of Jensen's inequality, the inequality becomes an equality when

$$\frac{|g(x)|p(x)}{p_0(x)}$$

is a constant, that is, when

$$p_0(x) \propto |g(x)|p(x)$$

that is,

$$p_0(x) = \frac{|g(x)|p(x)}{\int |g(u)|p(u)\, du}.$$

This is an infeasible choice, as we do not know the denominator.

Optimal Importance Sampling

Note that, in general, if $X_1, \ldots, X_N, \ldots \sim p_0$, then

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p(X_i)}{p_0(X_i)} \xrightarrow{a.s.} \int_{-\infty}^{\infty}\frac{p(x)}{p_0(x)}p_0(x)\,dx = 1$$

so therefore

$$\widetilde{I}_N^{(p_0)}(g) = \frac{\sum_{i=1}^{N}\frac{g(X_i)p(X_i)}{p_0(X_i)}}{\sum_{i=1}^{N}\frac{p(X_i)}{p_0(X_i)}} \xrightarrow{a.s.} \mathbb{E}[g(X)]$$

also, if the expectation exists. For finite N, this estimator $\widetilde{I}_N^{(p_0)}(g)$ is biased, but asymptotically unbiased. It may have smaller variance than $\widehat{I}_N^{(p_0)}(g)$

Optimal Importance Sampling

It seems appealing to combine this with the optimality result. The estimator

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

is feasible. When $g(.)$ is positive

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

that is, the *harmonic mean* estimator. Unfortunately, this estimator often has poor properties.

Optimal Importance Sampling

However, to get as close to the variance bound as possible, a sensible objective is to choose p_0 so that

$$\frac{|g(x)|p(x)}{p_0(x)}$$

is almost constant, such that the variance is **finite**.

This suggests designing p_0 to have high density whenever the original integrand $|g(x)|p(x)$ is large, subject to the constraint

$$\frac{p(x)}{p_0(x)} < M \qquad \text{or} \qquad \mathbb{E}_p\left[\frac{p(X)}{p_0(X)}\right] < M$$

for some finite bound M.

Monte Carlo and Importance Sampling in Higher Dimensions

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

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

Uniform random number generation

Generation of truly random uniform variates mechanically is reasonably challenging. All statistical packages rely on *pseudo-random* uniform generation:

- sequences $\{u_n\}$ that are generated by some **deterministic** mechanism, but that appear to be random even under severe statistical testing.

Pseudo-random number generators are typically built on deterministic equations of the form

$$u_{n+1} = h(u_1, \ldots, u_n)$$

constructed to that u_{n+1} is apparently not predictable from previous elements.

Congruential Generators

A (linear) *congruential generator* takes the form

$$u_{n+1} = au_n + b \qquad \text{mod } c$$

for integers a, b, c, where $0 < a < c$ and $0 \leqslant b < c$. The initialization of the sequence is achieved by setting the seed u_0, $0 \leqslant u_0 < c$. The recursion generates integers on $\{0, 1, \ldots, c-1\}$, which can be transformed into uniform random variates by rescaling.

Care is needed in the choice of the constants, but provided

- a and b are coprime,
- $a - 1$ is divisible by all prime factors of c
- $a - 1$ is a multiple of 4 if c is a multiple of 4.

the generator has full cycle length c. Typically $c = 2^{32}$ is chosen.

Congruential Generators: defects

Congruential generators can perform adequately, but also have known defects. They are principally favoured because of simplicity of computation.

Most obviously, the generated points can be shown to exhibit specific forms of serial dependence. For examples, certain subsequences of points can be demonstrated to lie on a hyperplanes (and hence be predictable).

Predictability is an undesirable property for randomness, but may not be too problematic in Monte Carlo for certain integrals.

Recursion/shift Generators

More recently, vector recursion generators have superseded congruential generators. These algorithms take linear (matrix) combinations of current vectors to produce new vectors in the sequence.

The most widely used shift generator is the *Mersenne Twister*, which is the default in R, and in its most typical implementation has a cycle length of

$$2^{19937} - 1$$

It produces very reliable uniform pseudo-random variates that are virtually indistinguishable from truly random numbers.

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: Efficiency

Note that

$$\begin{aligned}
\mathbb{P}[X \text{ is accepted}] &= \mathbb{P}\left[U \leqslant \frac{p(X)}{Mp_0(X)}\right] \\
&= \int_{-\infty}^{\infty} \left\{ \int_0^{p(x)/(Mp_0(x))} du \right\} p_0(x)\, dx \\
&= \int_{-\infty}^{\infty} \frac{p(x)}{Mp_0(x)} p_0(x)\, dx \\
&= \frac{1}{M} \int_{-\infty}^{\infty} p(x)\, dx = \frac{1}{M}
\end{aligned}$$

so ideally M should be as small as possible.

Rejection sampling: Efficiency

Note also neither p nor p_0 need to be normalized for this algorithm to be valid:

- If $p(x) = mg(x)$ and $p_0(x) = m_0 g_0(x)$, then

$$\frac{p(x)}{p_0(x)} = \frac{m}{m_0}\frac{g(x)}{g_0(x)} < M$$

  or

$$\frac{g(x)}{g_0(x)} < M' = \frac{mM}{m_0}$$

  is the rejection sampling bound.
- We proceed by bounding $g(x)/g_0(x)$.
- The acceptance probability is now indeterminate, however, by monitoring the empirical acceptance rate, an estimate of m/m_0 can be obtained.

Rejection sampling

Example (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

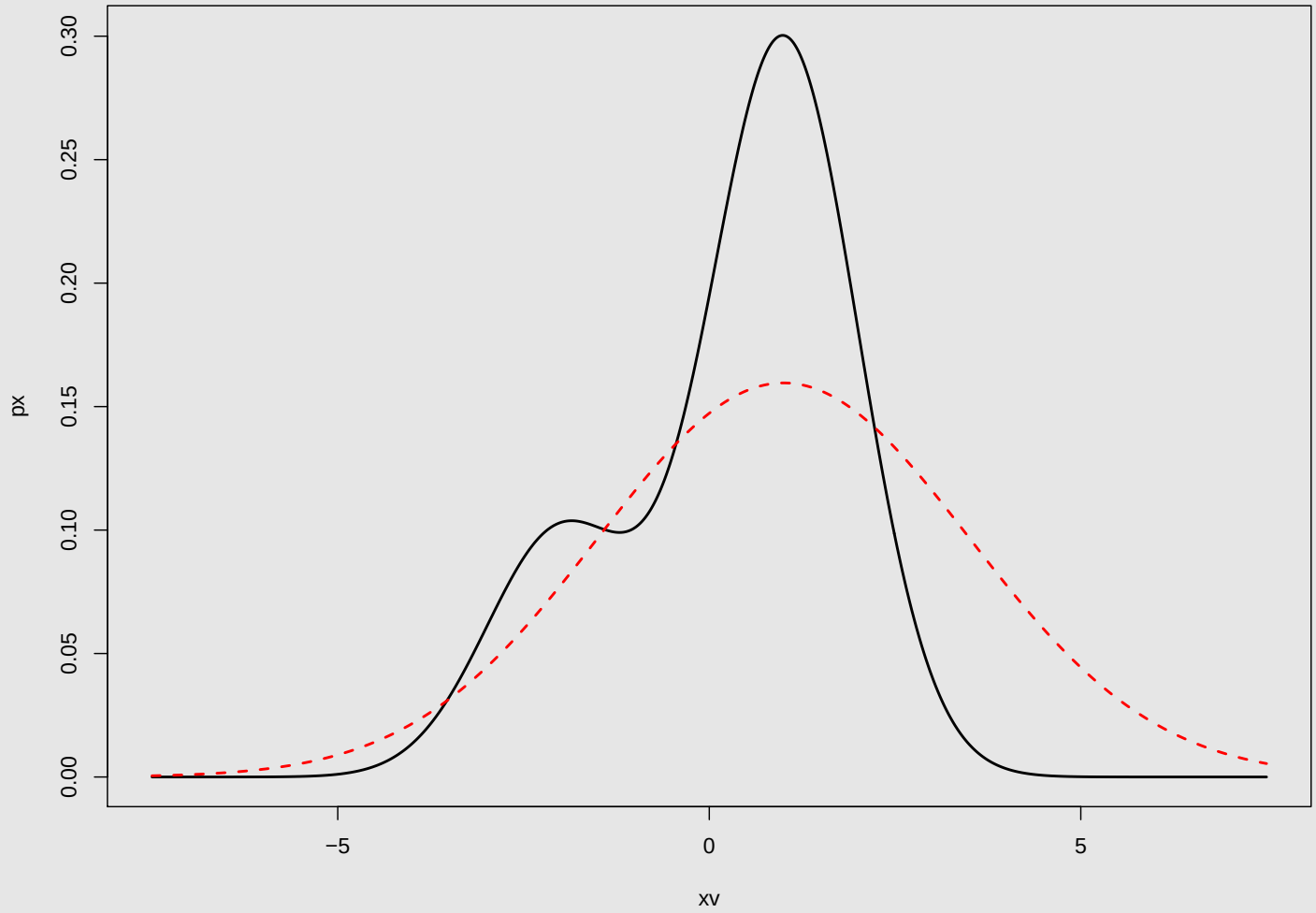
$$p_0(x) = \frac{1}{\sigma}\phi\left(\frac{x-1}{\sigma}\right)$$

for some variance σ^2.

Rejection sampling

Example (Normal mixture)

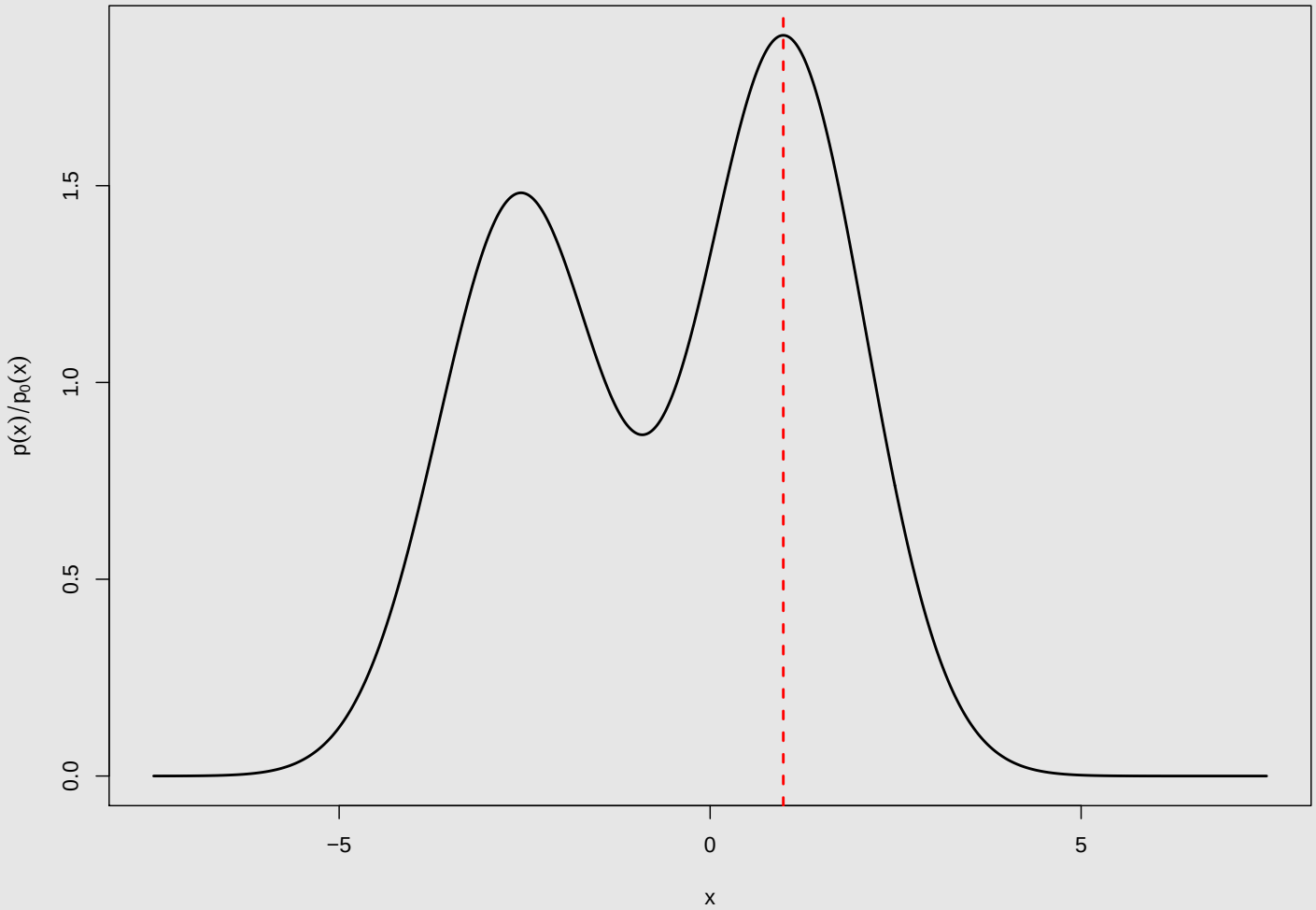
Target $p(x)$ (solid) and $p_0(x)$ (dashed) with $\sigma = 2.5$.



Rejection sampling

Example (Normal mixture)

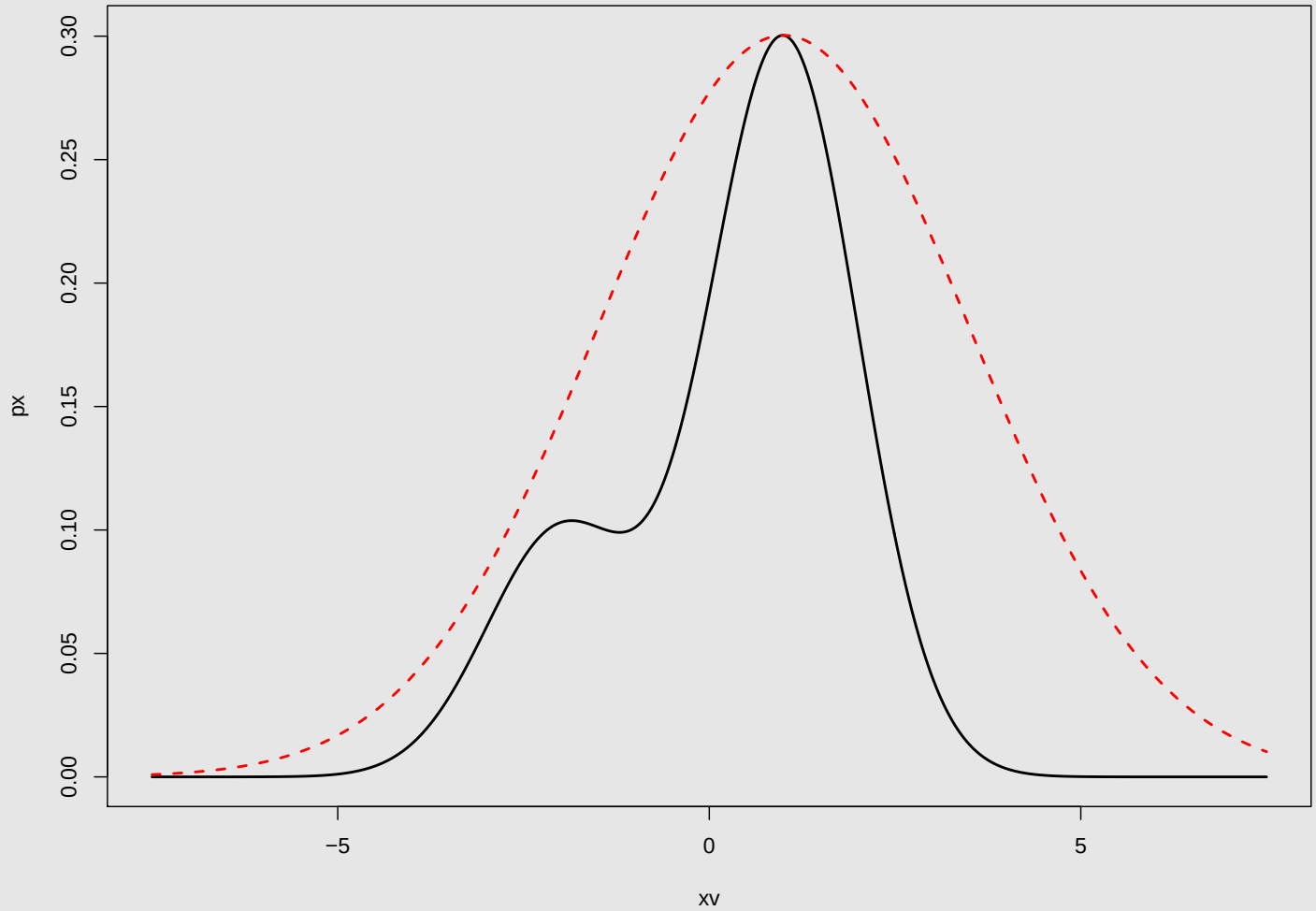
$p(x)/p_0(x)$ maximized at $x = 0.9863$, yielding $M = 1.8821$:



Rejection sampling

Example (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

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

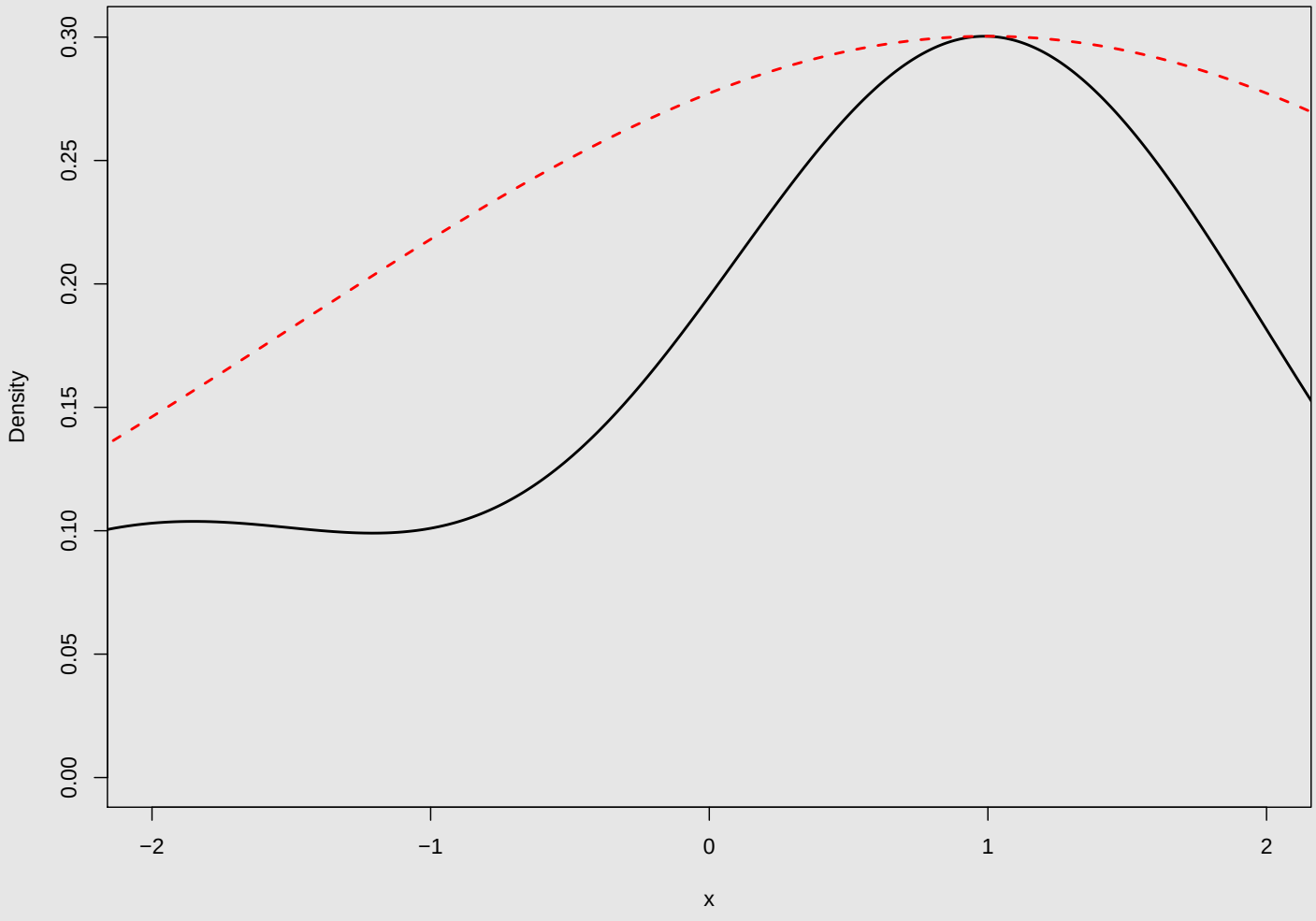
By assumption $p(x) \leqslant Mp_0(x)$. If u is simulated from $Uniform(0, 1)$, then $uMp_0(x)$ is the random portion of the vertical slice, and if

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

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

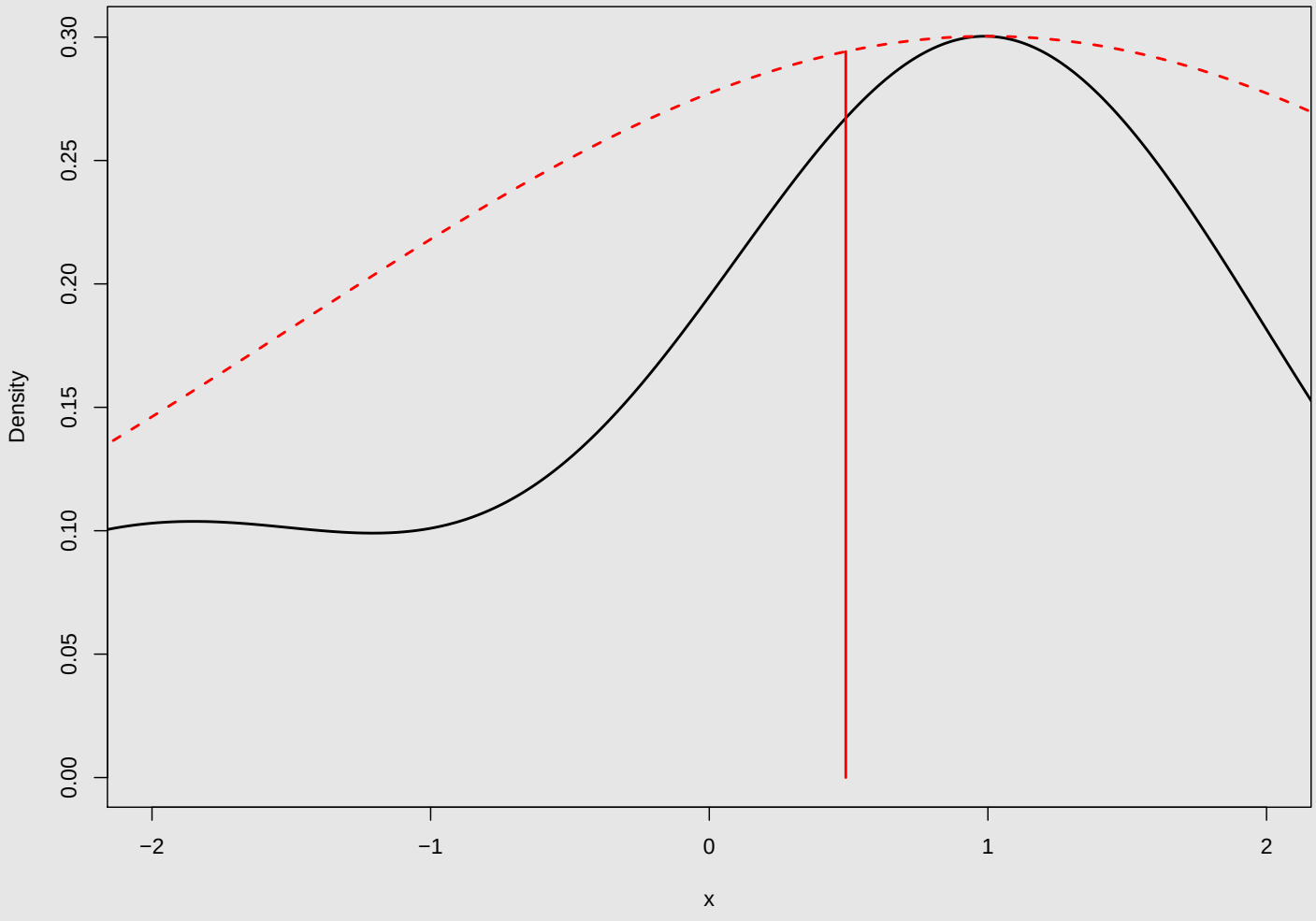
Rejection sampling

Example (Normal mixture)



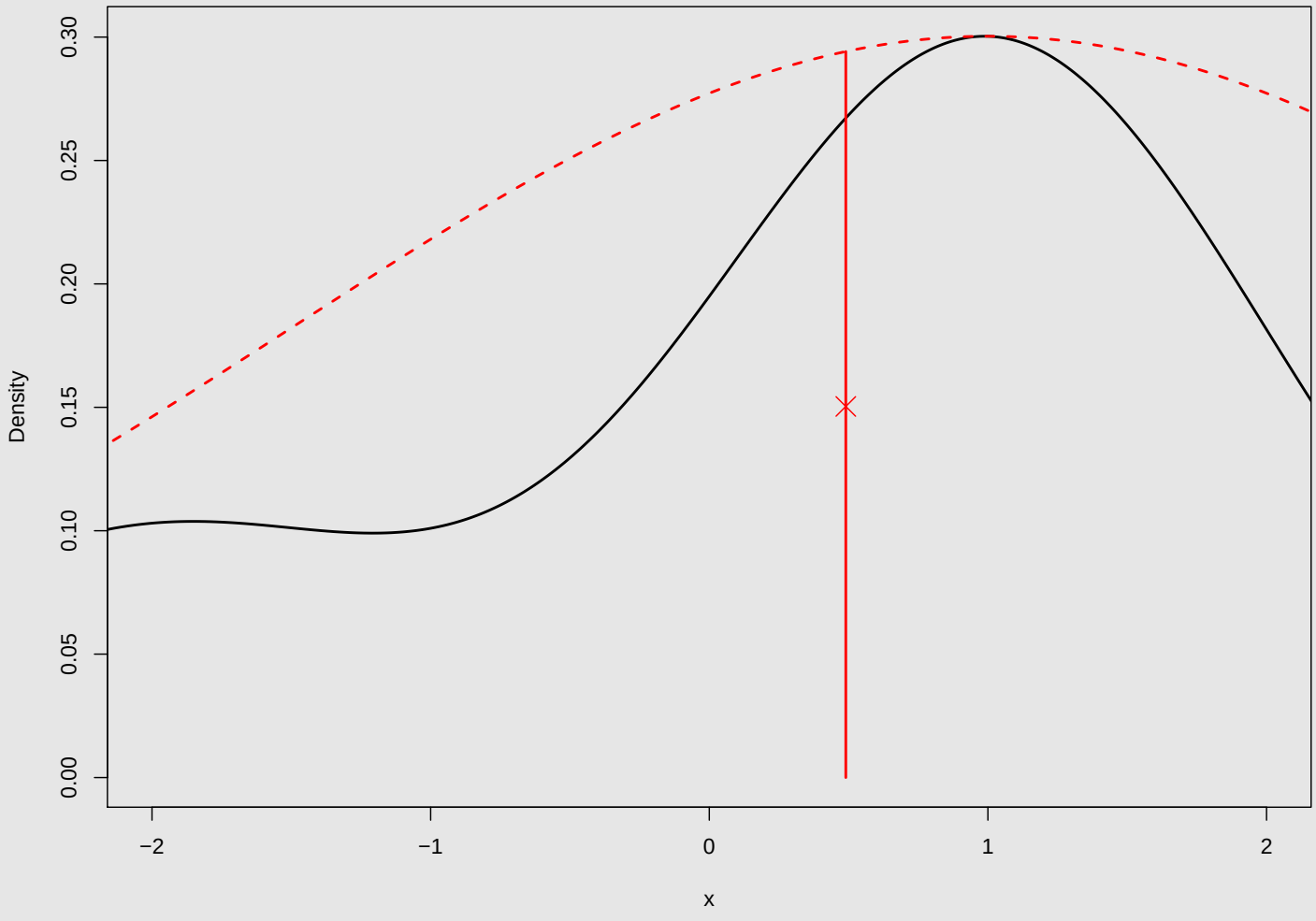
Rejection sampling

Example (Normal mixture)



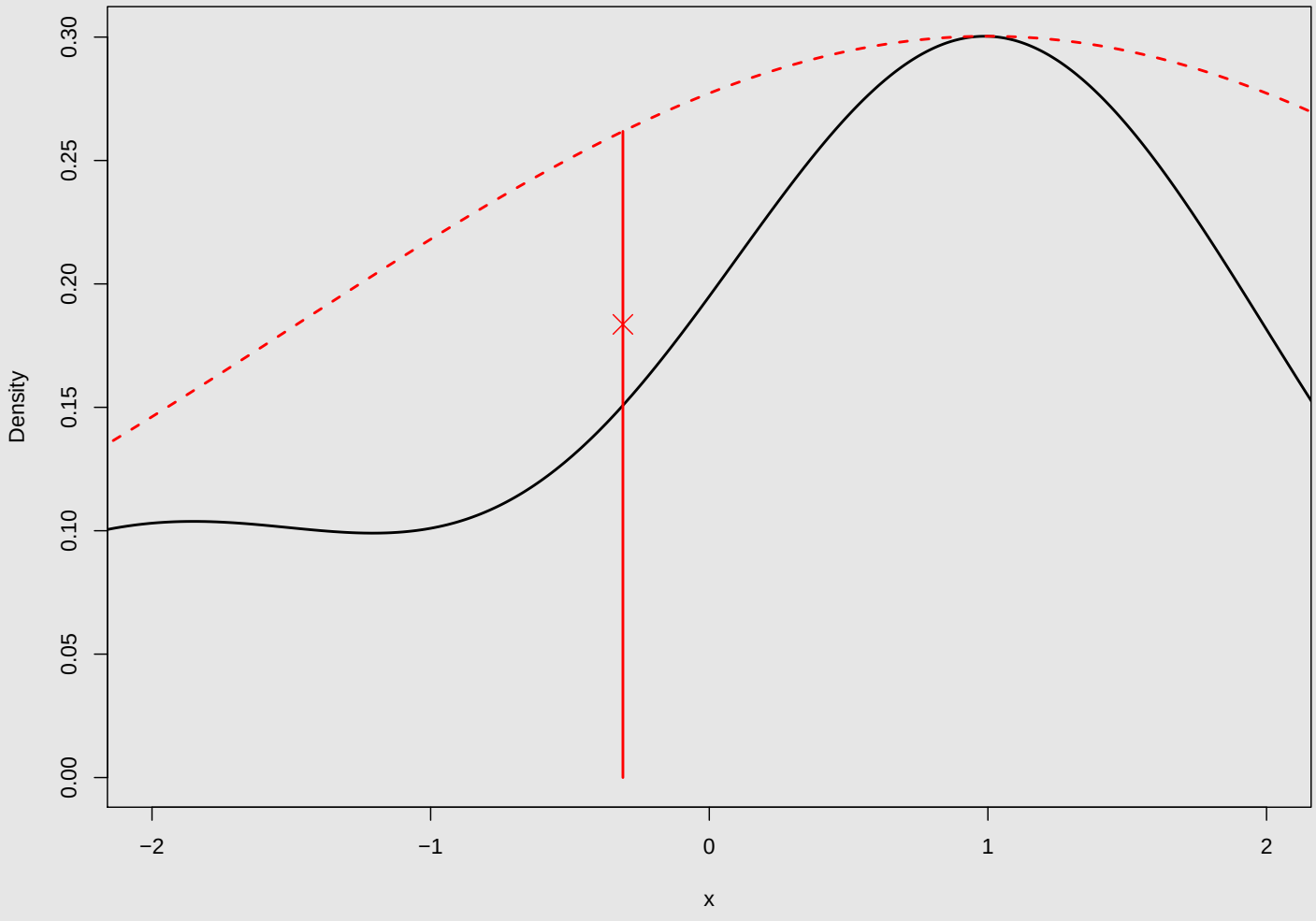
Rejection sampling

Example (Normal mixture)



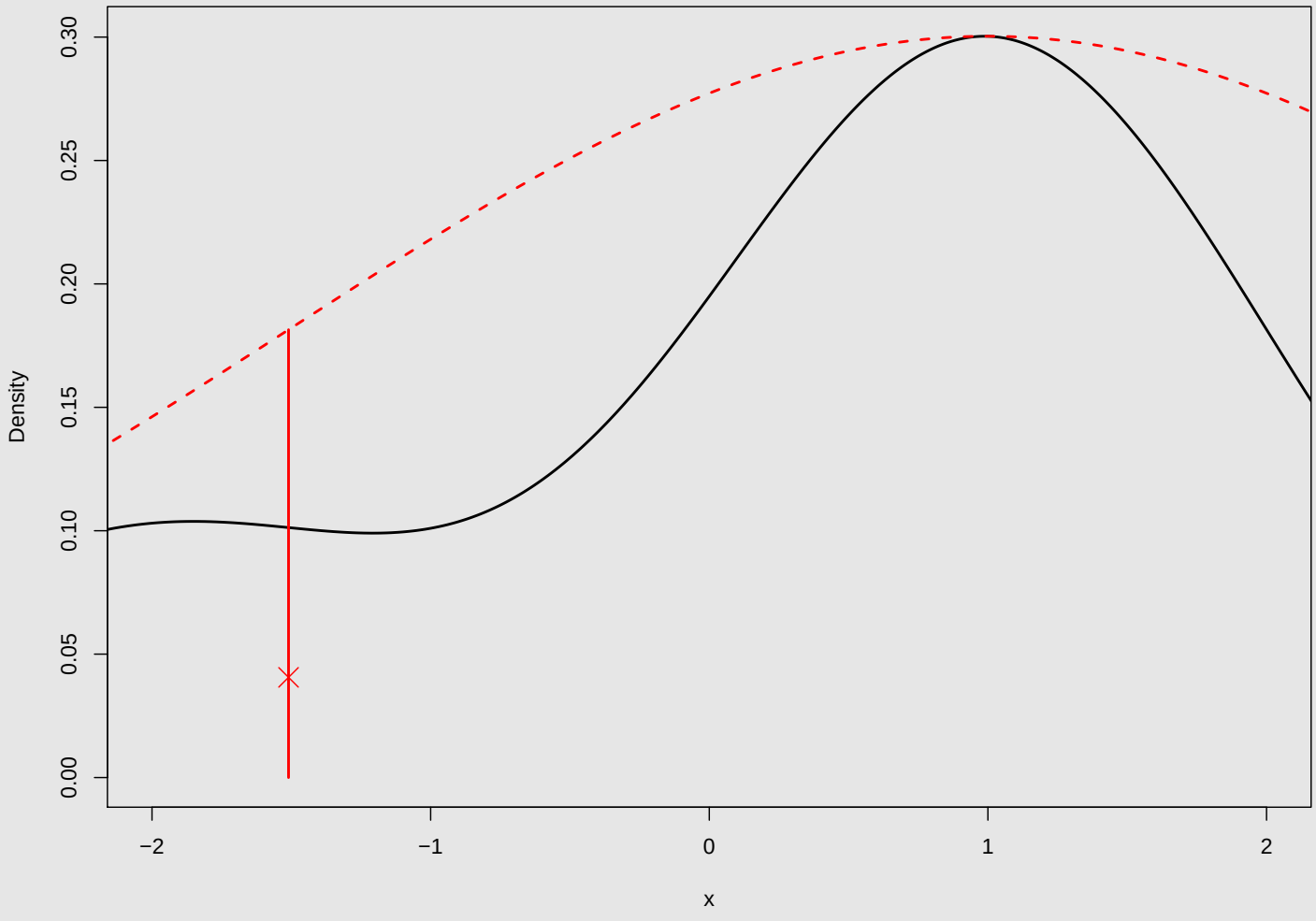
Rejection sampling

Example (Normal mixture)



Rejection sampling

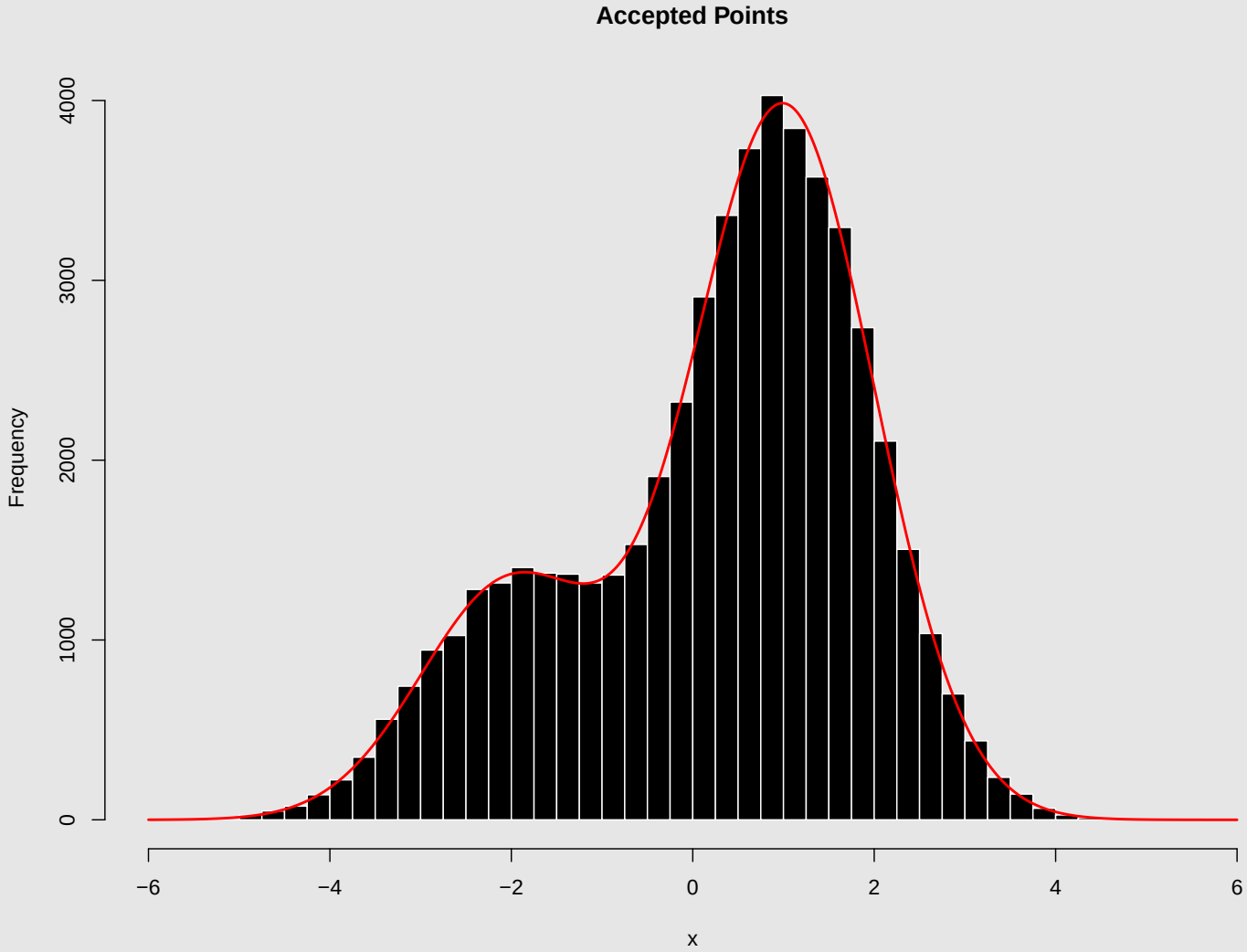
Example (Normal mixture)



Rejection sampling

Example (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.

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

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

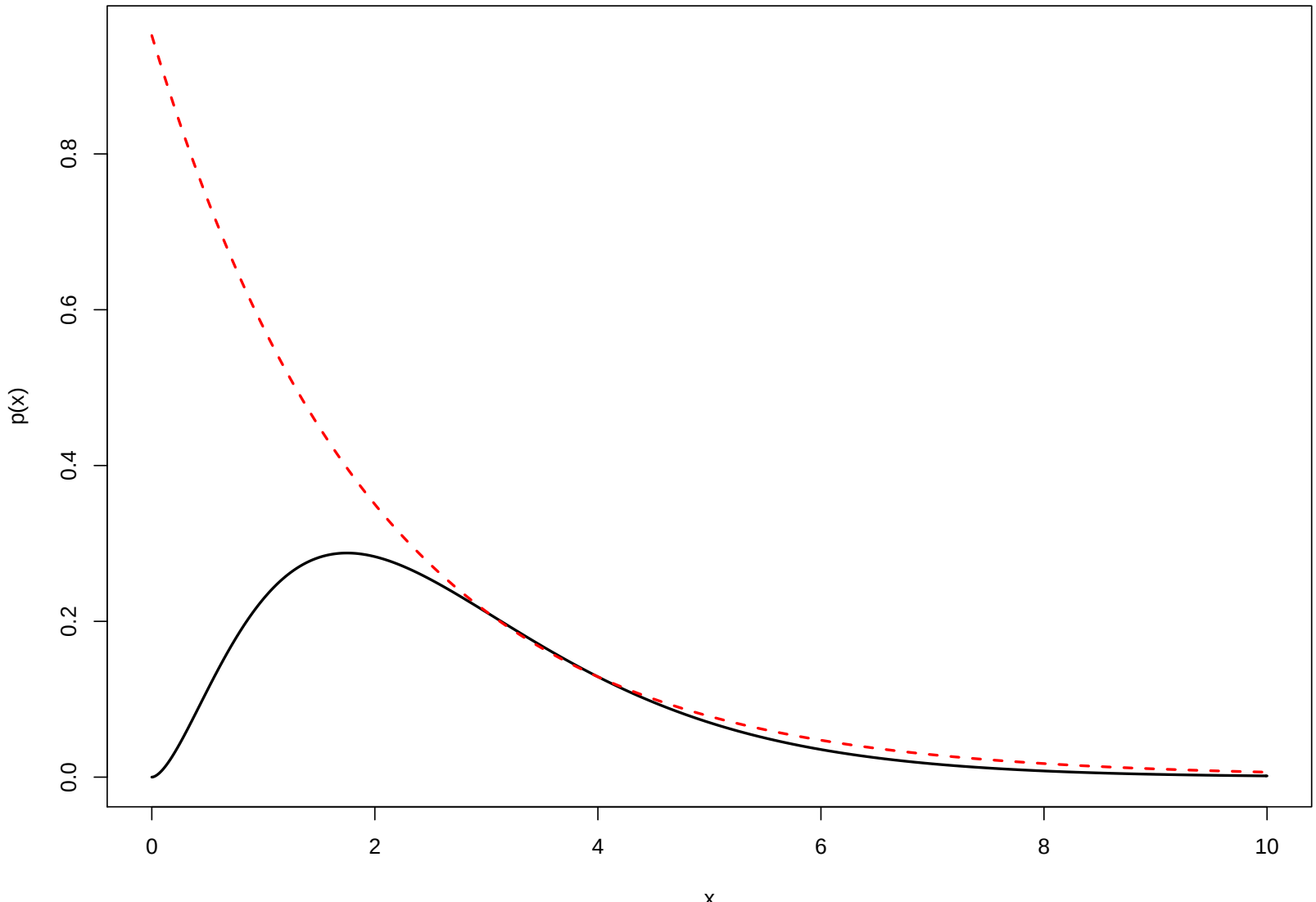
which achieves its unique maximum at

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

Thus rejection sampling can be carried out using this p_0.

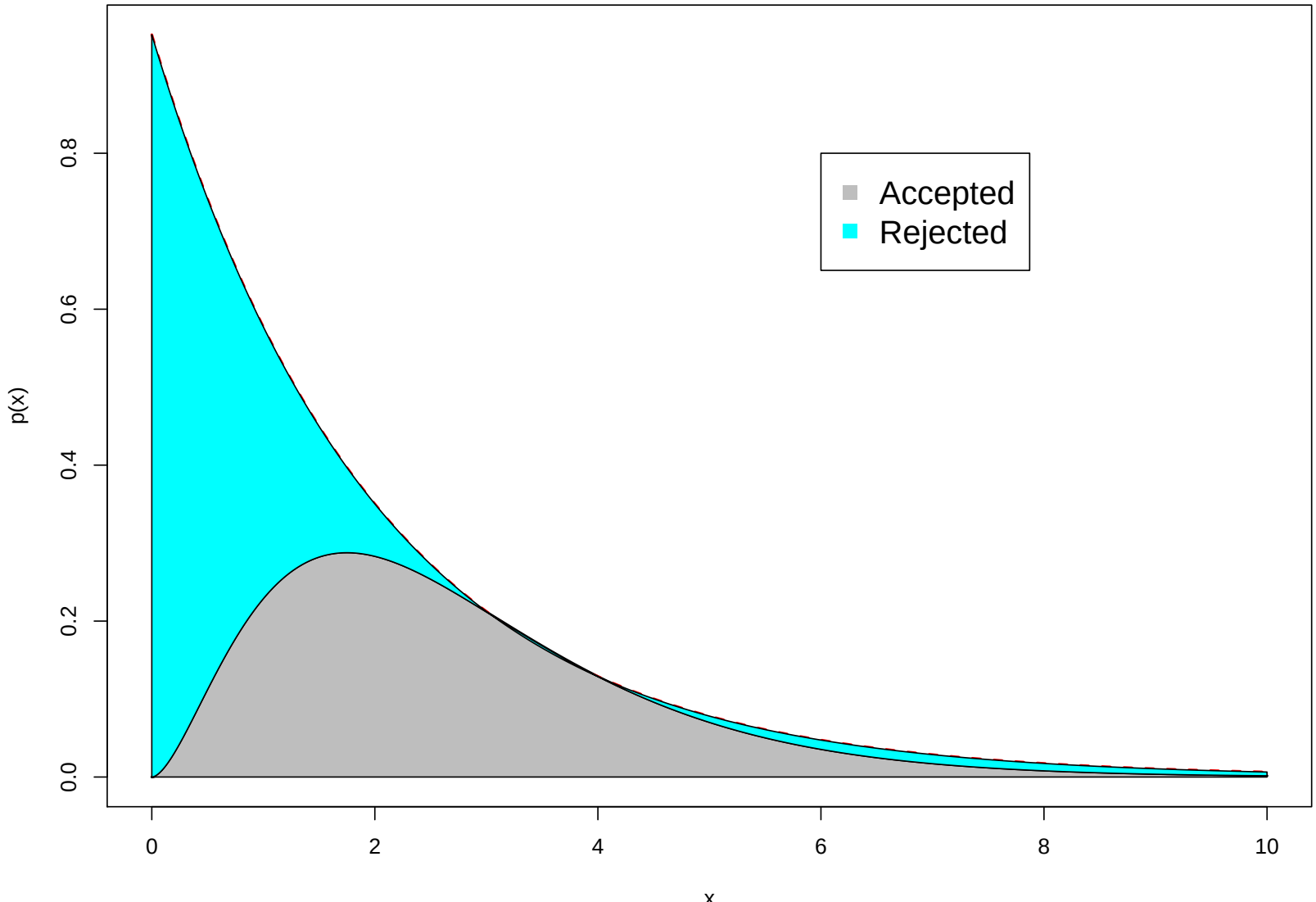
Adaptive Rejection sampling

However, the $Exponential(1/2)$ distribution is potentially inefficient



Adaptive Rejection sampling

However, the $Exponential(1/2)$ 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.

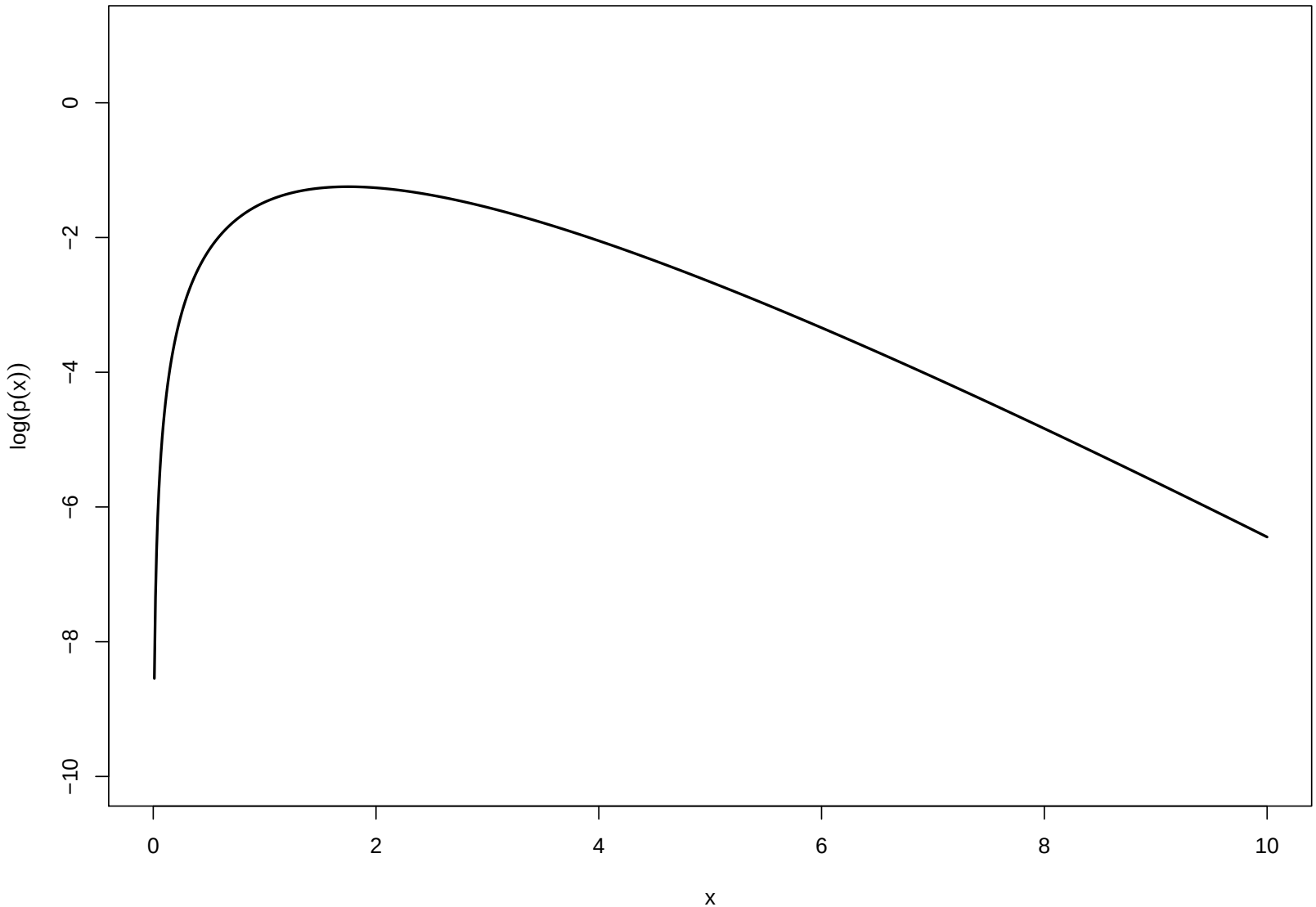
Adaptive Rejection sampling

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

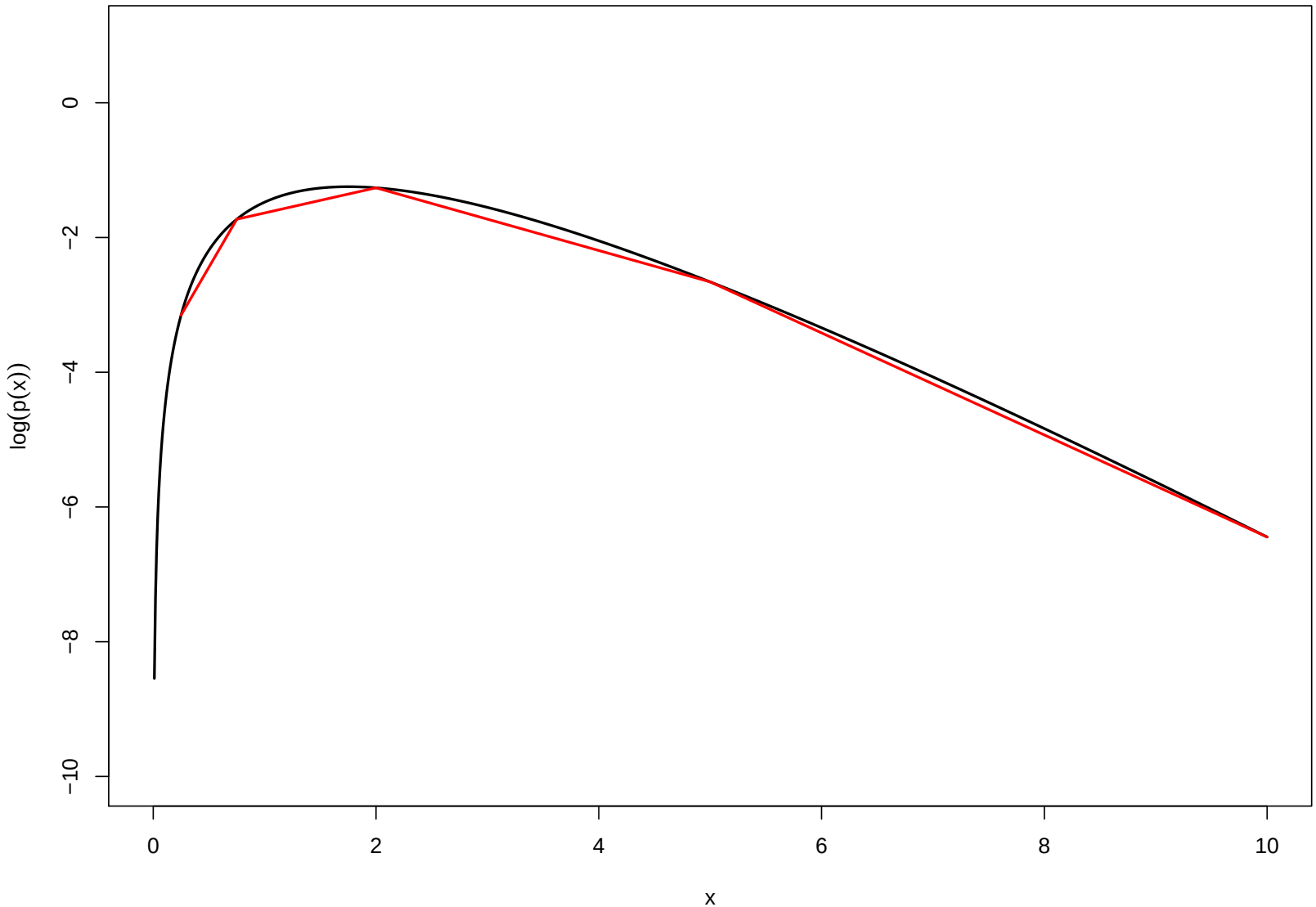
$$L_{i,i+1}(x) : (x_i, y_i) \longrightarrow (x_{i+1}, y_{i+1})$$

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

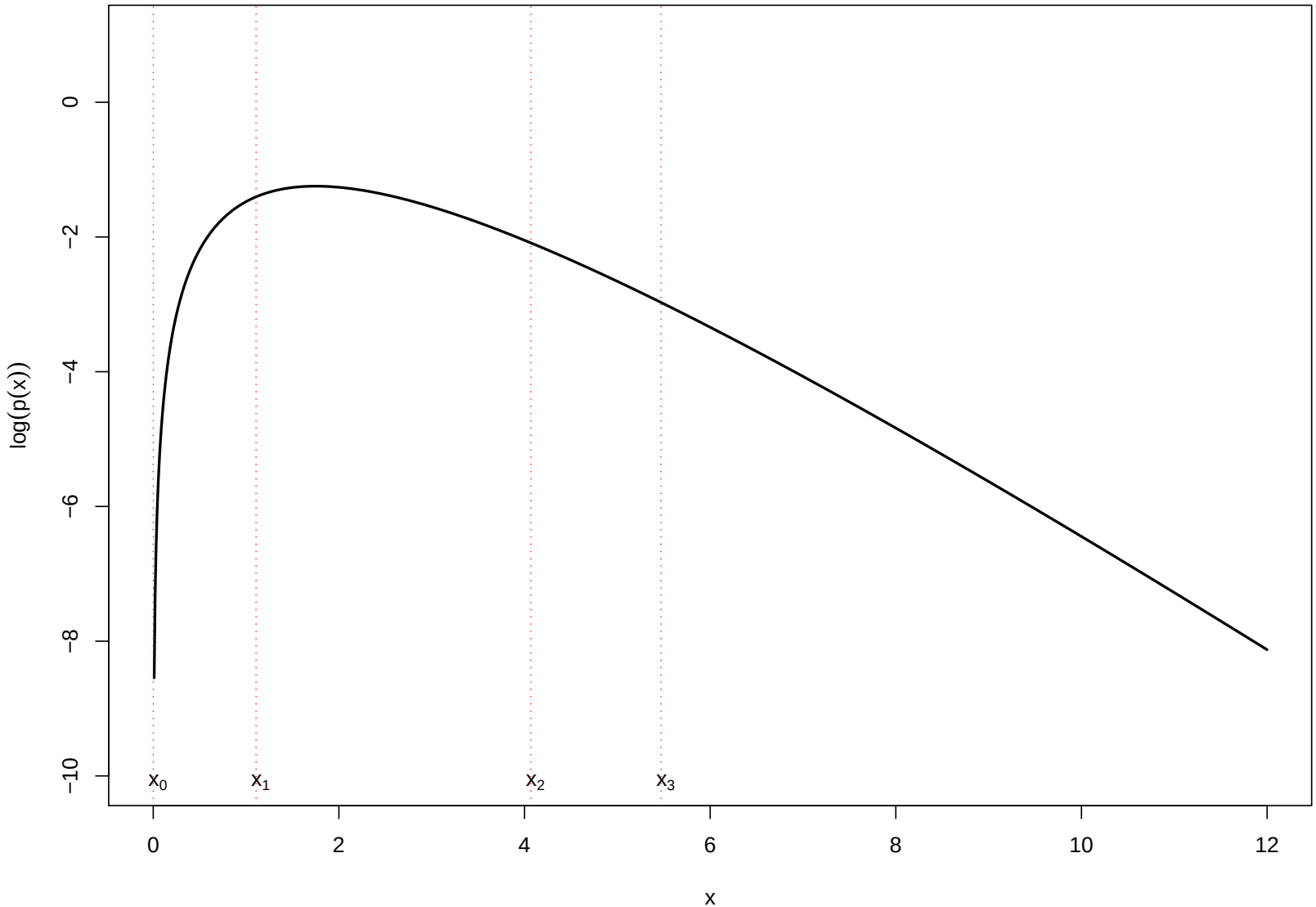
$$\exp\{l_n(x)\} \leqslant p(x) \leqslant \exp\{u_n(x)\} = M_n p_{0n}(x)$$

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

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

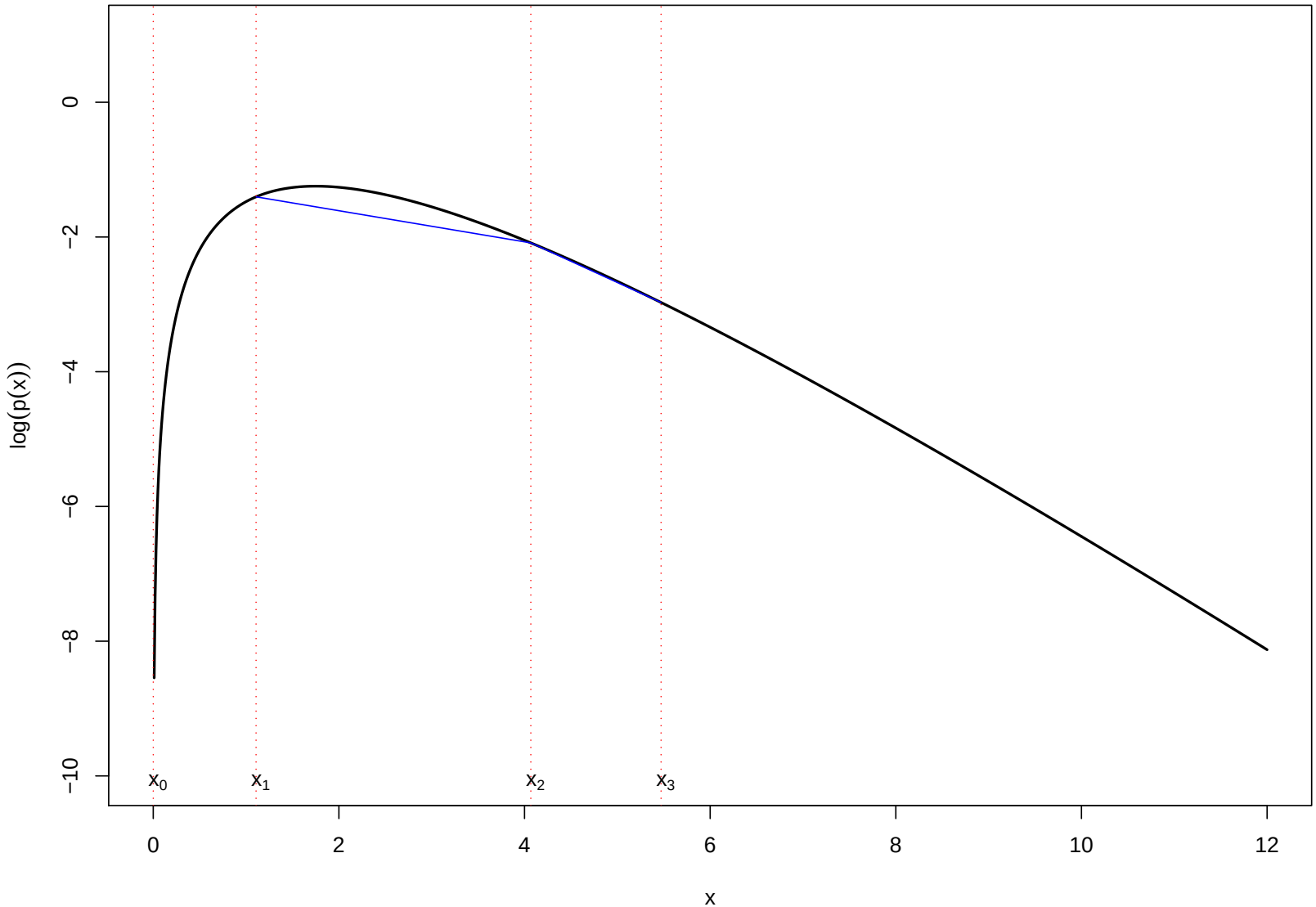
Adaptive Rejection sampling

Log-density:



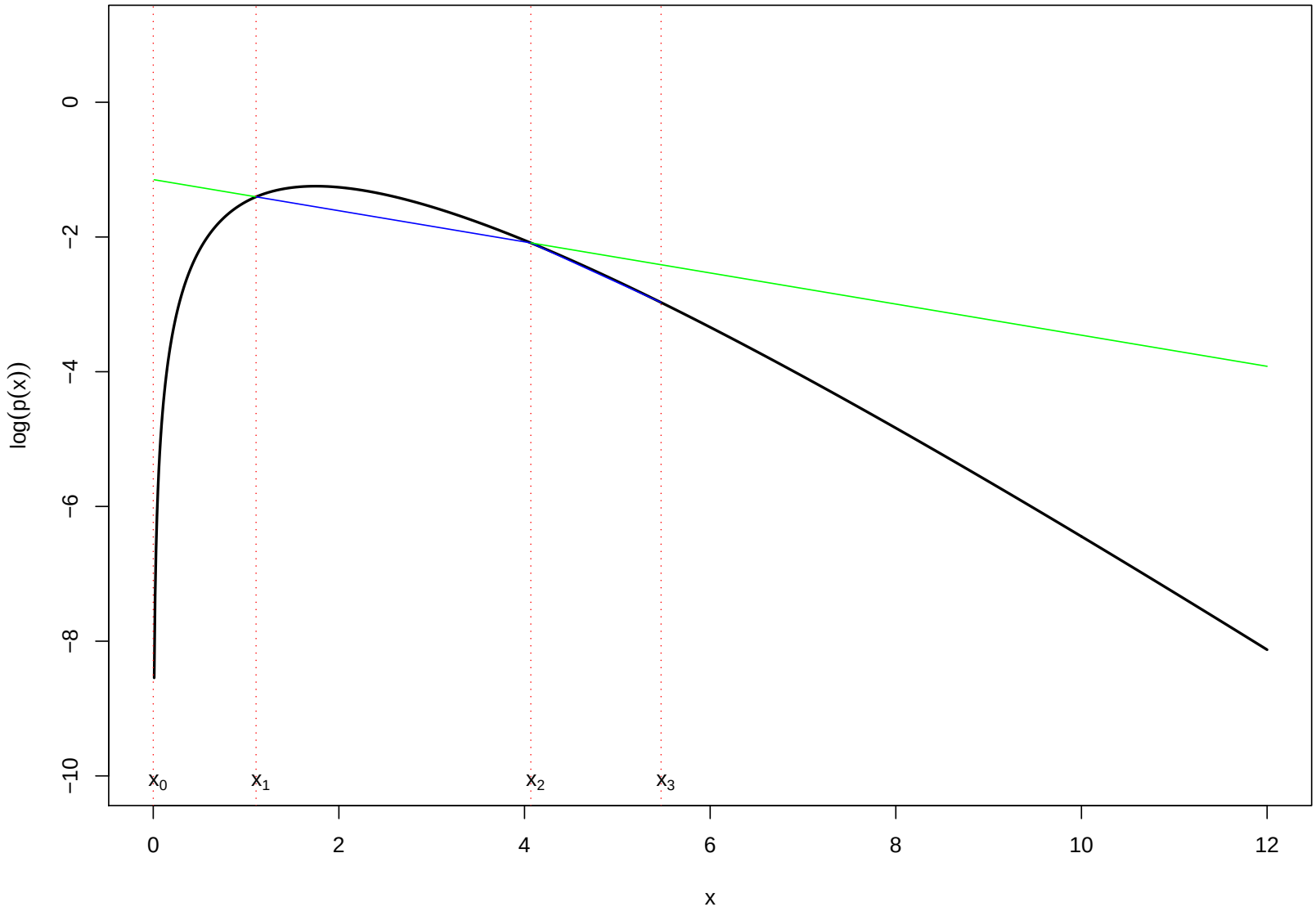
Adaptive Rejection sampling

Lower Envelope



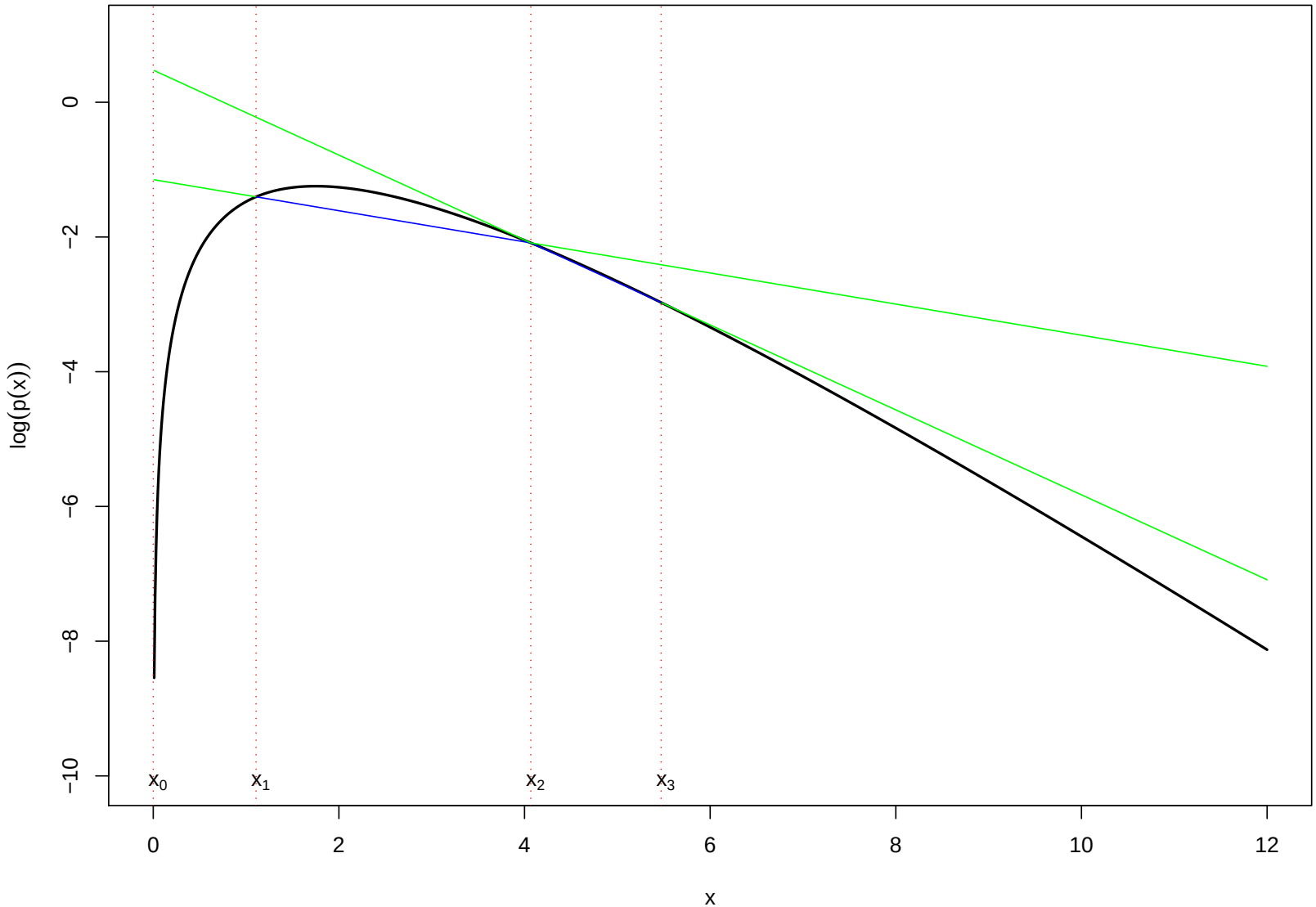
Adaptive Rejection sampling

Upper envelope (part 1):



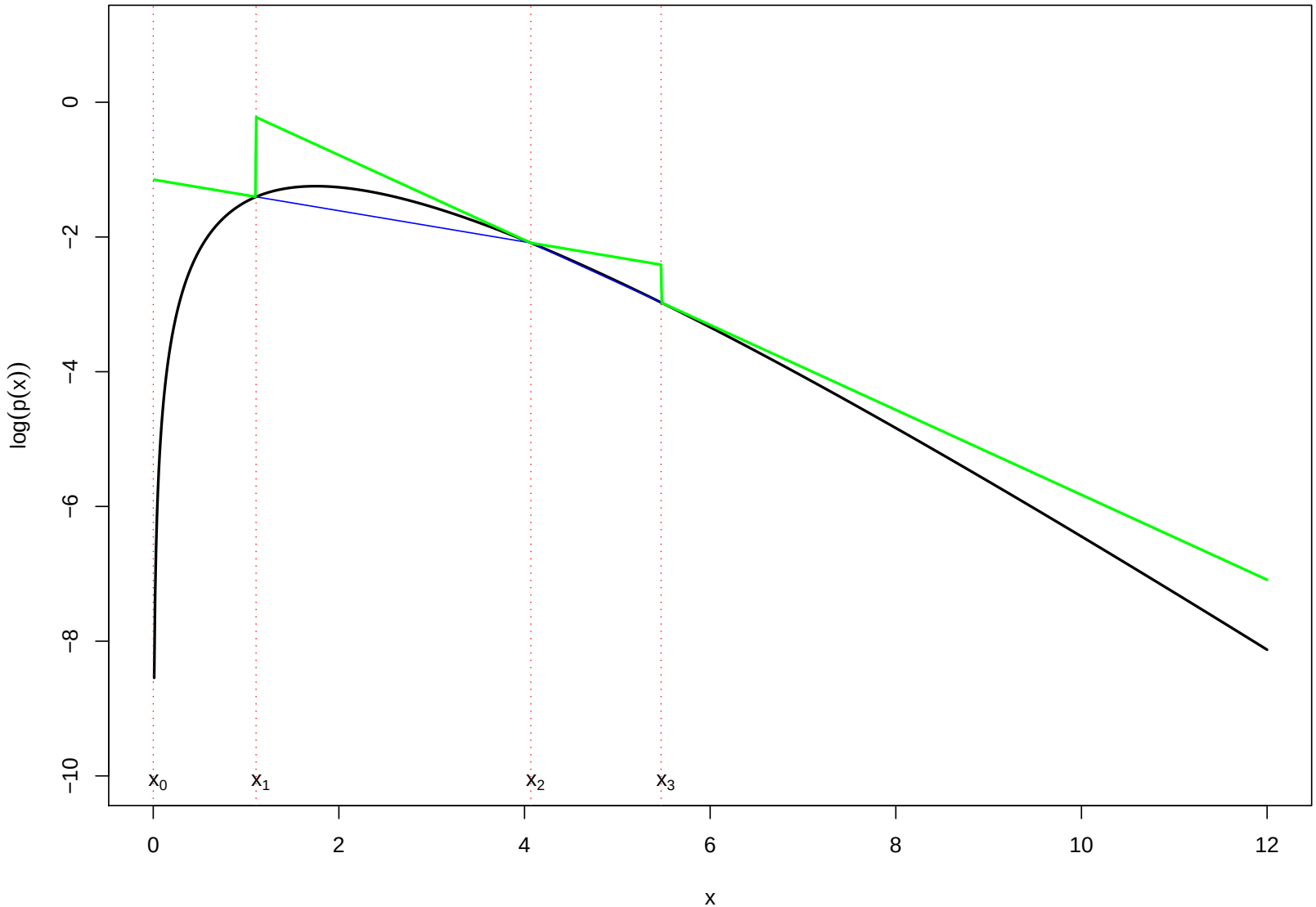
Adaptive Rejection sampling

Upper envelope (part 2):



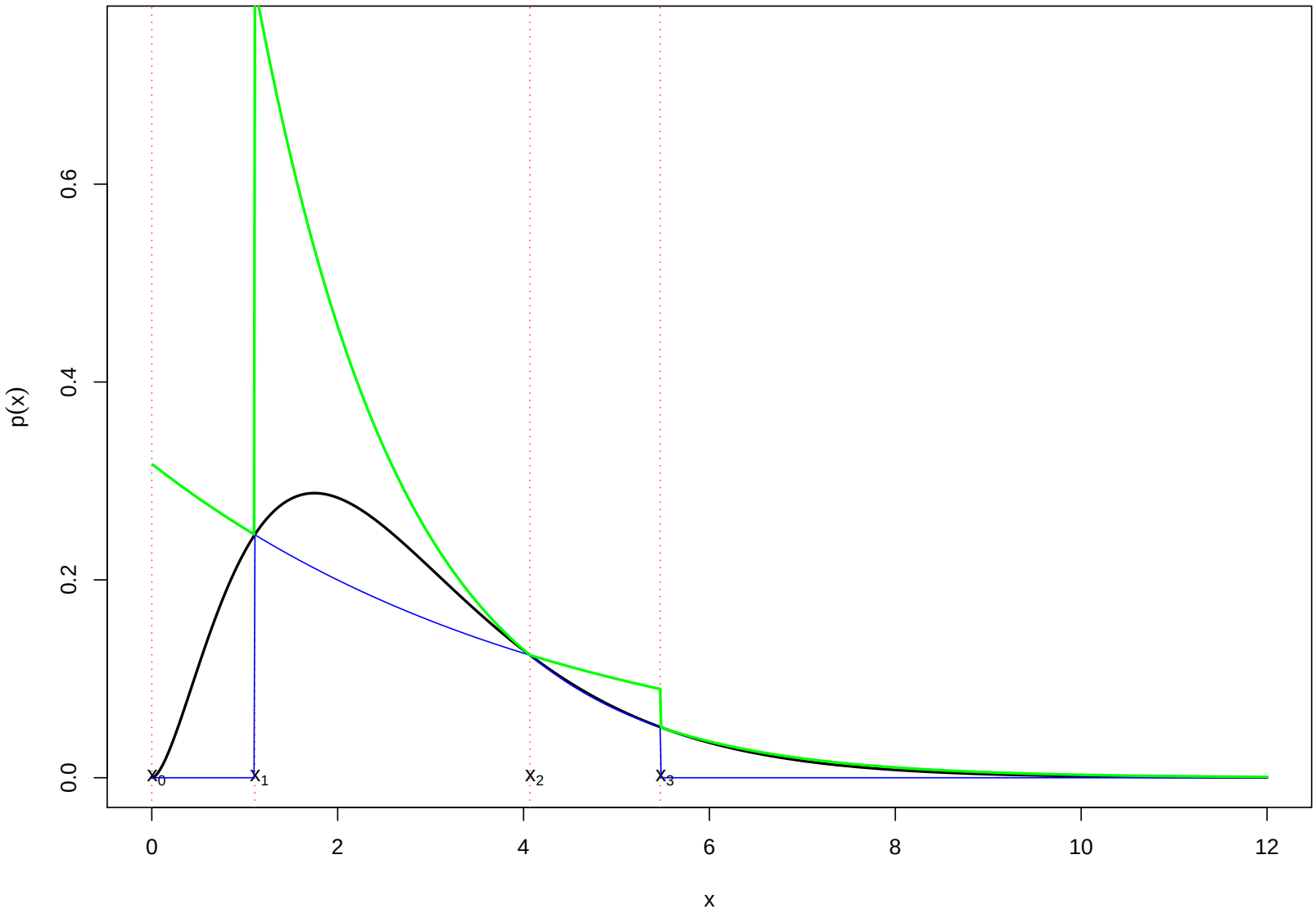
Adaptive Rejection sampling

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 $p(x)$, and 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.

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 $p(x)$.

Sampling Importance Resampling

Sampling Importance Resampling (SIR) can be used to sample (approximately) from density $p(x)$ by *re-weighting* and then *resampling* samples from $p_0(x)$:

1. Generate variates $x_1, \ldots, x_N$ from p_0.
2. Compute renormalized weights $w_1, \ldots, w_N$ given by

$$w_i = \frac{p(x_i)/p_0(x_i)}{\sum_{j=1}^{N} (p(x_i)/p_0(x_i))} \qquad i = 1, \ldots, N.$$

3. Resample y_i from the discrete distribution on $\{x_1, \ldots, x_N\}$ with masses $\{w_1, \ldots, w_N\}$.

Sampling Importance Resampling

Returning to the previous example: suppose

$$\begin{aligned} p(x) &= \frac{1}{4}\phi(x+2.5) + \frac{3}{4}\phi(x-1) \\ p_0(x) &= \phi(x/\sigma) \end{aligned}$$

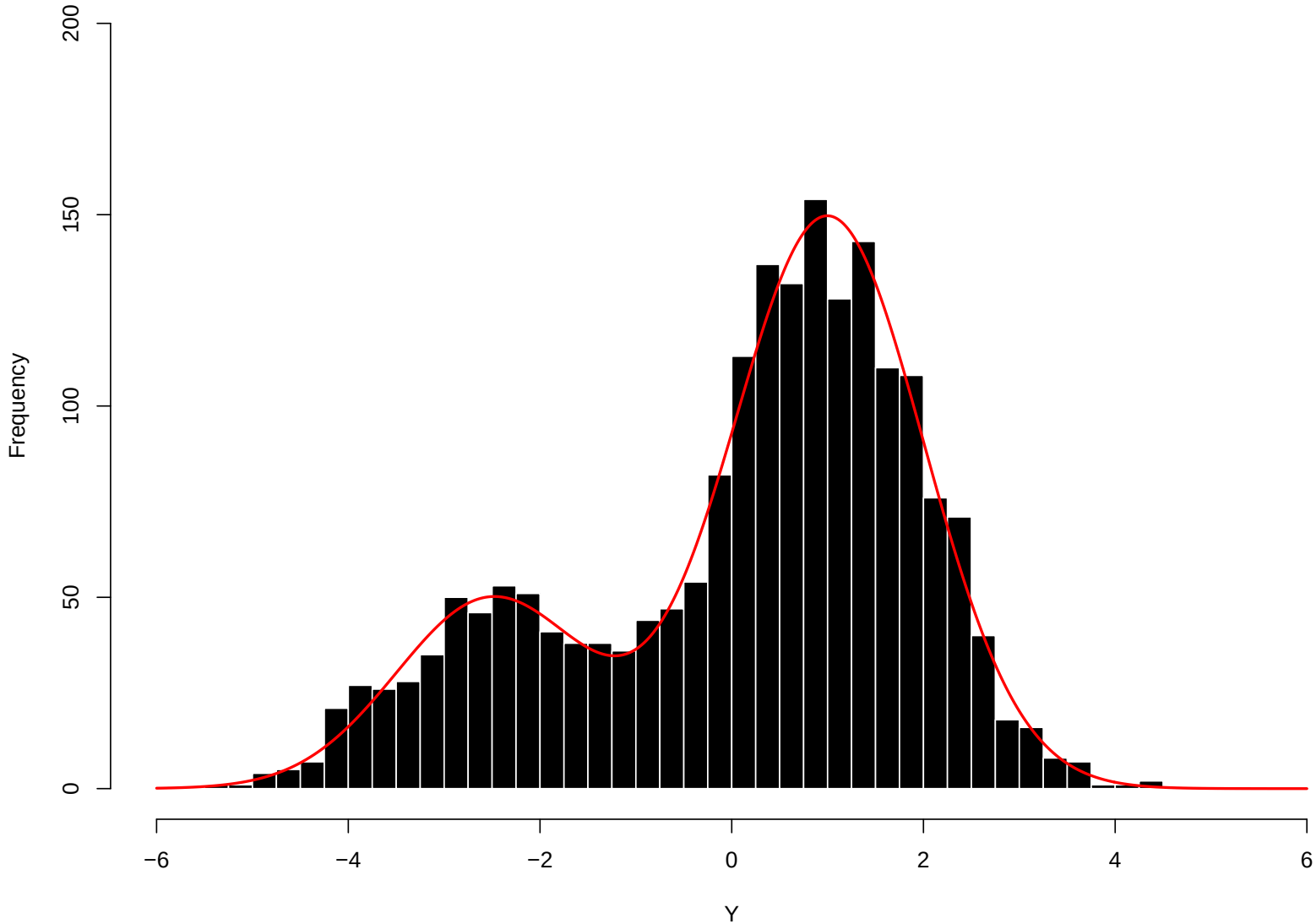
for some $\sigma > 0$.

Choosing σ poorly can disrupt the performance of the SIR algorithm.

- need to ensure there are samples in the tails of $p(x)$.

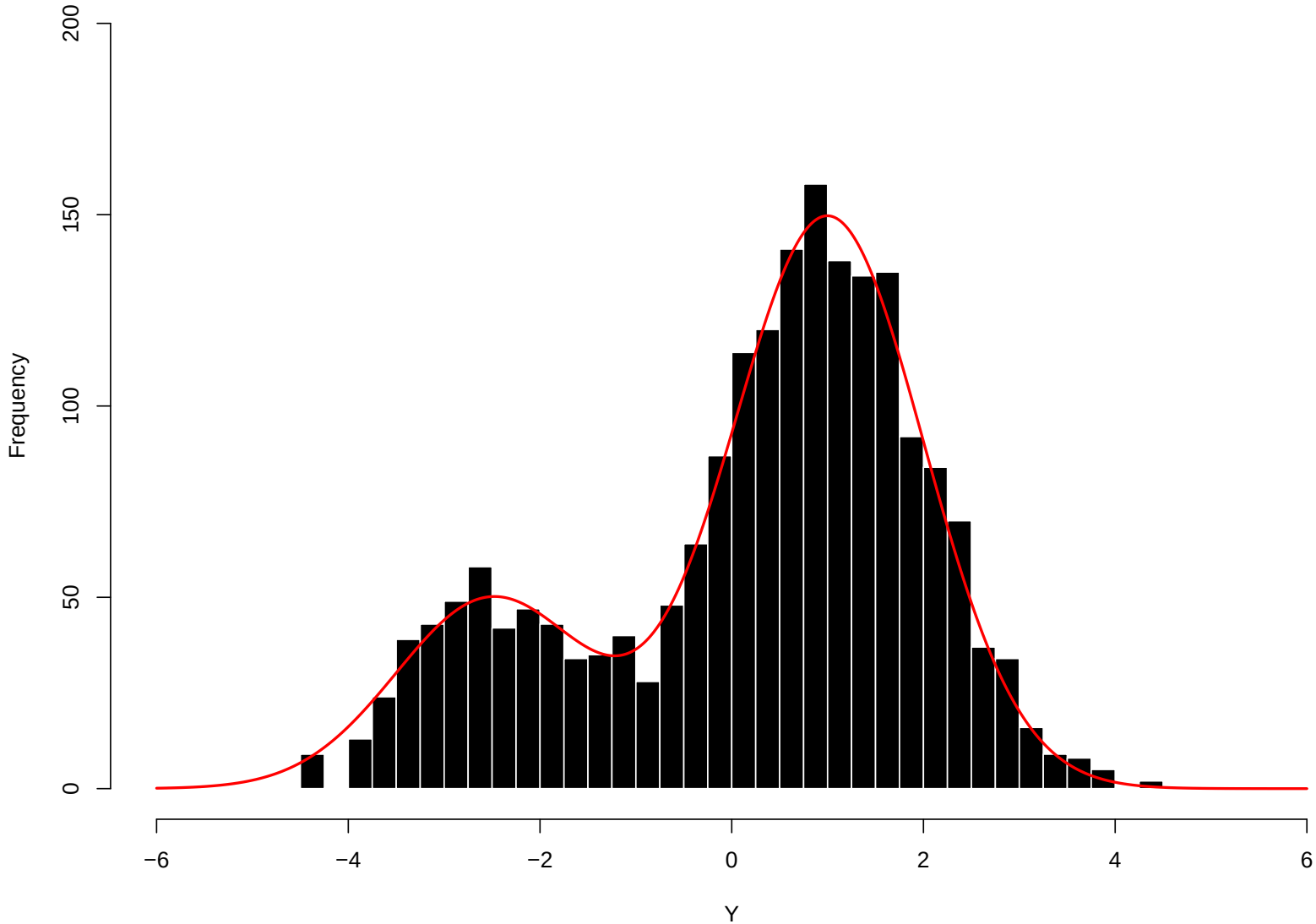
Sampling Importance Resampling

SIR with $\sigma = 2$ ($N = 100000$, 2000 resampled points):



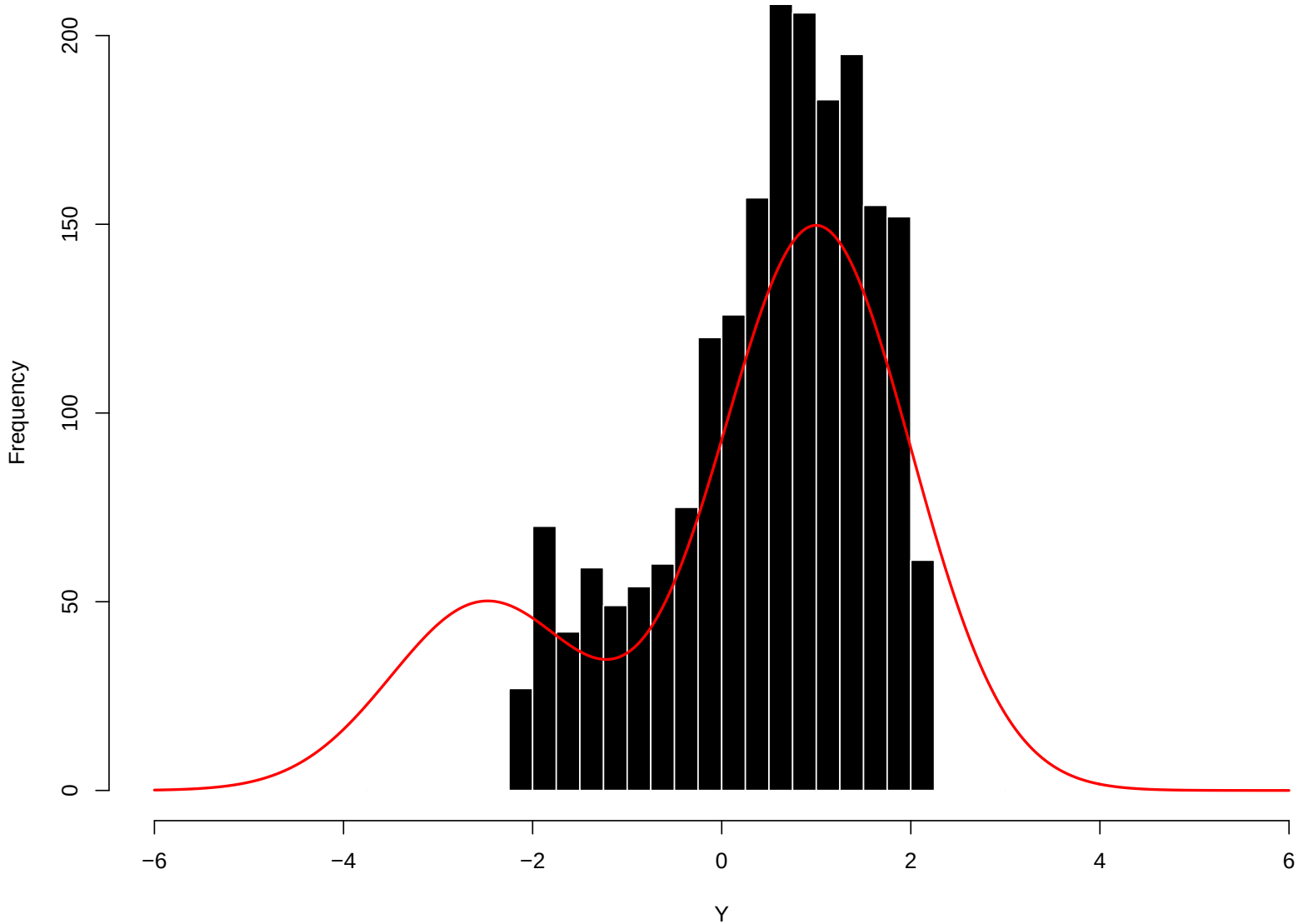
Sampling Importance Resampling

SIR with $\sigma = 1$ ($N = 100000$, 2000 resampled points):



Sampling Importance Resampling

SIR with $\sigma = 0.5$ ($N = 100000$, 2000 resampled points):



Variance Reduction: Antithetic Variables

Suppose that the integral

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

is to be estimated via Monte Carlo. The usual estimator

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

with $X_1, \ldots, X_N$ i.i.d. from p, has variance

$$\frac{\mathbb{V}ar[g(X)]}{N}$$

We aim to reduce this variance.

Variance Reduction: Antithetic Variables

Consider the estimator

$$\breve{I}_{N_A}(g) = \frac{1}{2N_A} \sum_{i=1}^{N_A} \left[g(X_i) + g(Y_i) \right],$$

where pairs $(X_1, Y_1), \ldots, (X_{N_A}, Y_{N_A})$ are i.i.d. from some bivariate distribution with identical marginal distribution p, and $2N_A = N$.

The variance of this estimator is

$$\frac{\mathbb{V}ar[g(X)]}{2N_A} + \frac{\mathbb{C}ov[g(X), g(Y)]}{2N_A}$$

Therefore if $\mathbb{C}ov[g(X), g(Y)]$ is **negative**, this variance is smaller than the usual Monte Carlo variance. This is termed the method of *antithetic variables.*

Variance Reduction: Antithetic Variables

Example (Bivariate Normal)

Suppose $(X, Y)^\intercal \sim \mathcal{N}_2(\mu \mathbf{1}^\intercal, \Sigma)$, with variances 1 and

$$\mathbb{C}ov[X, Y] = \rho$$

In the estimation of μ by Monte Carlo, the relative variances of the Monte Carlo and Antithetic Variable estimators is

$$\frac{\mathbb{V}ar[X]}{\mathbb{V}ar[X] + \mathbb{C}ov[X, Y]} = \frac{1}{1 - \rho}$$

Thus there is automatic variance reduction for the same total simulation size.

There is a slight computation overhead in computing $\breve{I}_{N_A}(g)$.

Variance Reduction: Control Variates

Suppose

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

with Monte Carlo estimator $\widehat{I}_N(g)$. Consider an adjusted Monte Carlo estimator of $I(g)$

$$\widehat{I}_N(g; g_0, \lambda) = \widehat{I}_N(g) + \lambda(\widehat{I}_N(g_0) - I(g_0))$$

for some function g_0, where $\widehat{I}_N(g_0)$ is an unbiased estimator of $I(g_0)$, and λ is some real constant. This estimator is unbiased for $I(g)$, and has variance

$$\mathbb{V}ar[\widehat{I}_N(g)] + \lambda^2\mathbb{V}ar[\widehat{I}_N(g_0)] + 2\lambda\mathbb{C}ov[\widehat{I}_N(g), \widehat{I}_N(g_0)]$$

Variance Reduction: Control Variates

Therefore, choosing

$$\lambda = -\frac{\mathbb{C}ov[\widehat{I}_N(g), \widehat{I}_N(g_0)]}{\mathbb{V}ar[\widehat{I}_N(g_0)]}$$

yields the variance of $\widehat{I}_N(g; g_0, \lambda)$ as

$$(1 - \mathbb{C}orr[\widehat{I}_N(g), \widehat{I}_N(g_0)]^2)\mathbb{V}ar[\widehat{I}_N(g)] \leqslant \mathbb{V}ar[\widehat{I}_N(g)]$$

This optimal choice of λ yields the maximum possible reduction for a given g_0, but requires knowledge of the sign of the covariance value.

In practice, the sign of λ can be assessed by regressing $g(x_i)$ on $g_0(x_i)$.

Variance Reduction: Control Variates

Example (Normal tail probability)

Consider

$$I(g) = \int_{-\infty}^{\infty} g(x)p(x)\,dx = \int_{x_0}^{\infty} \phi(x)\,dx = \mathbb{P}[X > x_0]$$

Consider

$$g_0(x) = \mathbb{1}_{(0,\infty)}(x)$$

so that $I(g_0) = 1/2$. We examine the variance of the estimator

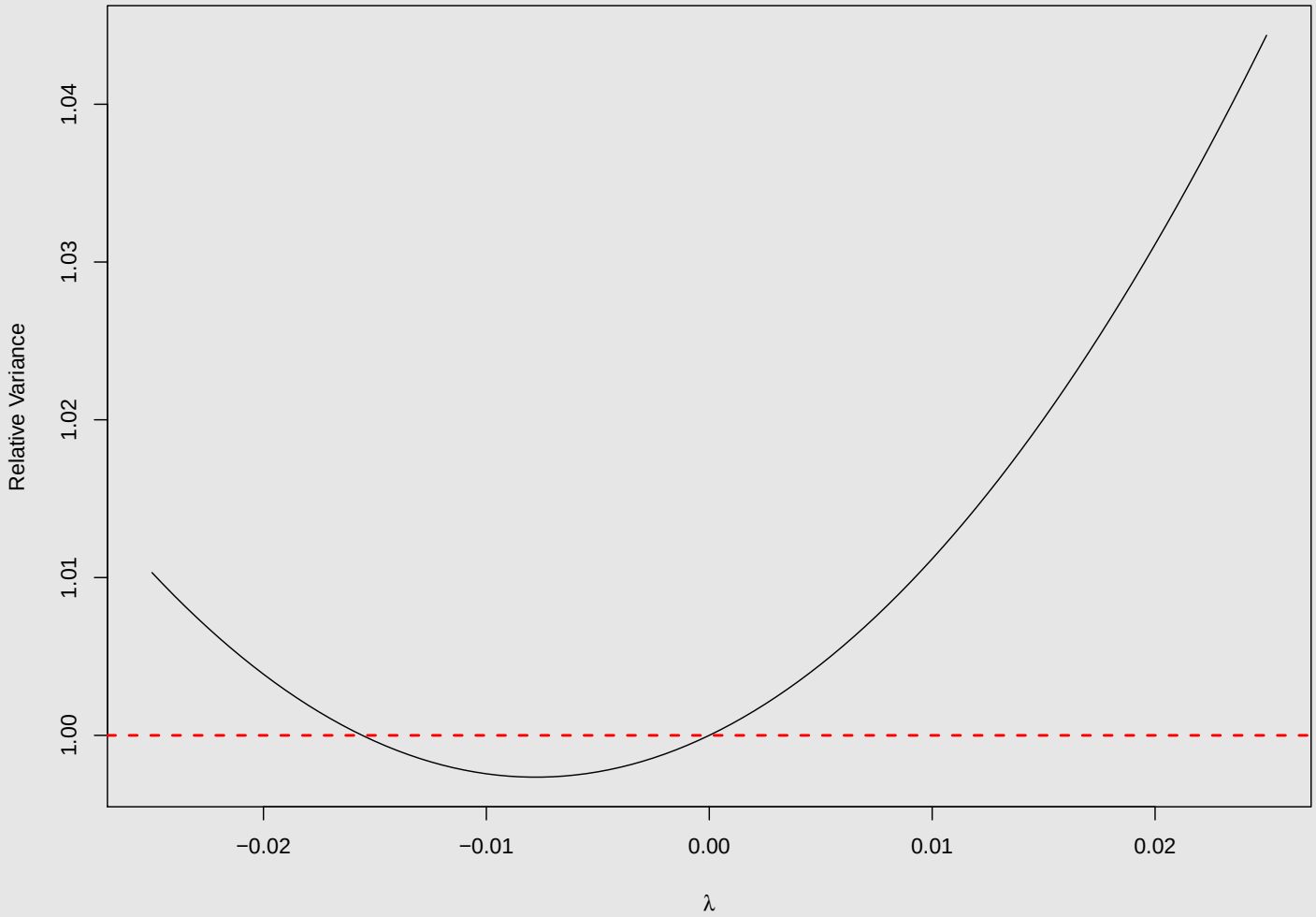
$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(x_0,\infty)}(X_i) + \lambda\left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(0,\infty)}(X_i) - \frac{1}{2}\right)$$

relative to the Monte Carlo estimator $\widehat{I}_N(g)$.

Variance Reduction: Control Variates

Example (Normal tail probability)

Relative variance for varying λ: $x_0 = 2.5$, $N = 20000$.



Variance Reduction: Control Variates

Example (Normal tail probability)

The best choice seems to be $\lambda \approx -0.008$.

In this case, finite sample bias might be exhibited. Therefore we also examine mean-square error (MSE)

$$\mathrm{MSE}(\lambda) = \mathbb{E}[(I(g) - \widehat{I}_N(g; g_0, \lambda))^2]$$

We see that the MSE is also lowest when $\lambda \approx -0.008$, indicating that if there is finite sample bias, it is relatively insignificant.

Variance Reduction: Control Variates

Example (Normal tail probability)

MSE for varying λ: $x_0 = 2.5$, $N = 20000$.

