Monte Carlo Methods

Computational Statistics

Johan Larsson

Department of Mathematical Sciences, University of Copenhagen

March 31, 2026

Rejection Sampling Recap

Suppose that $f(y) \propto q(y)$ is the **target** density, known up to normalization.

Suppose that $g(y) \propto p(y)$ is another density, known up to normalization, that we can simulate from, and suppose

$$\alpha' q \leq p$$

for some $\alpha' > 0$.

If Y has density g and U is uniform on $(0, 1)$ then the conditional distribution

$$Y \mid U \leq \alpha' \frac{q(Y)}{p(Y)}$$

has density f.

Algorithm 1: Rejection sampling to simulate from a random variable X with density $f \propto q$ using proposal Z with density $g \propto p$.

repeat
    Generate $x \sim Z$;
    Generate $u \sim \text{Uniform}(0, 1)$;
    if $u \leq \frac{\alpha' q(x)}{p(x)}$ **then**
        Accept x;
until x *is accepted*;

Two Challenges

- Finding an envelope p/α' of q.
- Finding a **tight-enough** envelope.

Today

Monte Carlo Methods

General class of methods for estimating integrals and means.

Rejection sampling is a type of Monte Carlo method.

Importance Sampling

Focus on integrating functions and estimating means.

Useful when we cannot analytically compute the integral.

It is also a variance reduction technique.

Basic, Simple Idea

Use random sampling to obtain numerical results.

Useful when analytical solutions are infeasible.

Monte Carlo Integration

With $X_1, \dots, X_n$ i.i.d. with density f,

$$\hat{\mu}_{\mathrm{MC}} = \frac{1}{n} \sum_{i=1}^{n} h(X_i) \to \mu = \mathrm{E}\, h(X_1) = \int h(x) f(x)\, dx$$

for $n \to \infty$ by the law of large numbers (LLN).

By the central limit theorem (CLT), we know that

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \stackrel{\text{approx}}{\sim} \mathcal{N}(\mu, \sigma^2_{\text{MC}}/n)$$

where

$$\sigma^2_{\text{MC}} = \operatorname{Var} h(X_1) = \int (h(x) - \mu)^2 f(x)\; dx.$$

We can estimate σ^2_{MC} using the empirical variance

$$\hat{\sigma}^2_{\mathrm{MC}} = \frac{1}{n-1} \sum_{i=1}^{n} (h(X_i) - \hat{\mu}_{\mathrm{MC}})^2,$$

then the variance of $\hat{\mu}_{\mathrm{MC}}$ is estimated as $\hat{\sigma}^2_{\mathrm{MC}}/n$ and a standard 95% confidence interval for μ is

$$\hat{\mu}_{\mathrm{MC}} \pm 1.96 \frac{\hat{\sigma}_{\mathrm{MC}}}{\sqrt{n}}.$$

We want to estimate the mean of a $\text{Gamma}(8, 1)$ distribution.

```
B <- 1000
x <- rgamma(B, 8)

mu_hat <- cumsum(x) / 1:B
sigma_hat <- sd(x)
mu_hat[B] # Theoretical: 8
```

```
[1] 8.096
```

```
sigma_hat # Theoretical: √8
```

```
[1] 2.862
```

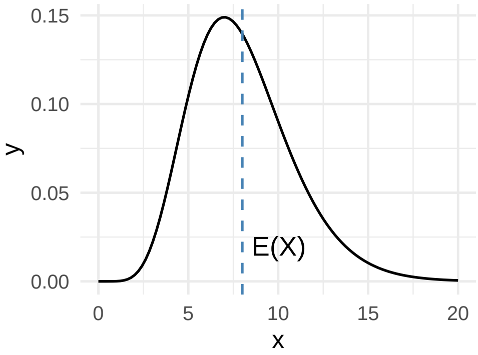


Figure 1: Density of a Gamma(8,1) distribution with mean indicated.

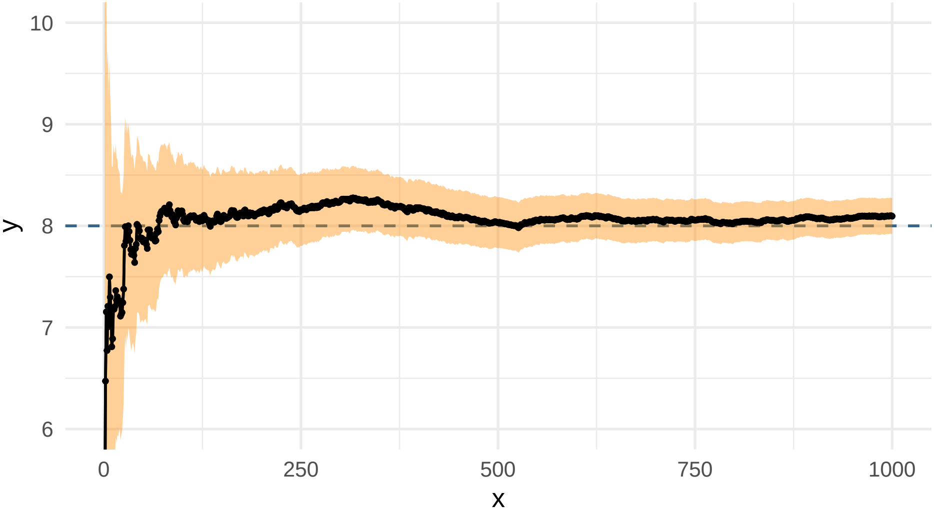


Figure 2: Convergence of the Monte Carlo estimate of the mean of a Gamma(8,1) distribution.

When to Stop?

We could use concentration inequalities (Chebyshev/Chernoff) to get a stopping rule.

But need variance estimate for Chebychev-based bound and moment generating functions for Chernoff-based bounds.

More commonly, just run until the empirical confidence interval is sufficiently narrow.

Trade-Offs
If you overshoot then you waste computer resources, and if you undershoot then you get a bad estimate.

When we are only interested in Monte Carlo integration, we do not need to sample from the target distribution.

Basic Idea

Some regions of the target distribution are often more important than others.

So let's oversample these regions, but correct for this by weighting the samples.

We want to find $\mu = \mathrm{E}\, h(X)$. Observe that

$$\mu = \int h(x) f(x)\ dx = \int h(x) \frac{f(x)}{g(x)} g(x)\ dx = \int h(x) w^*(x) g(x)\ dx$$

whenever g is a density fulfilling that

$$g(x) = 0 \Rightarrow f(x) = 0.$$

Weights

With $X_1, \ldots, X_n$ i.i.d. with density g define the *weights*

$$w^*(X_i) = \frac{f(X_i)}{g(X_i)}.$$

g is the *importance* density and f/g is called the *likelihood ratio*.

Estimator

$$\hat{\mu}^*_{\mathrm{IS}} := \frac{1}{n} \sum_{i=1}^{n} h(X_i) w^*(X_i).$$

It has mean μ.

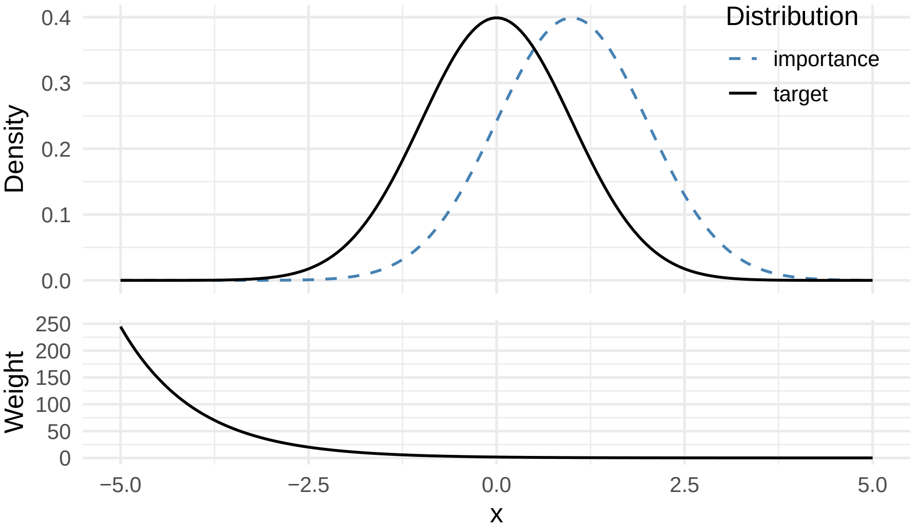


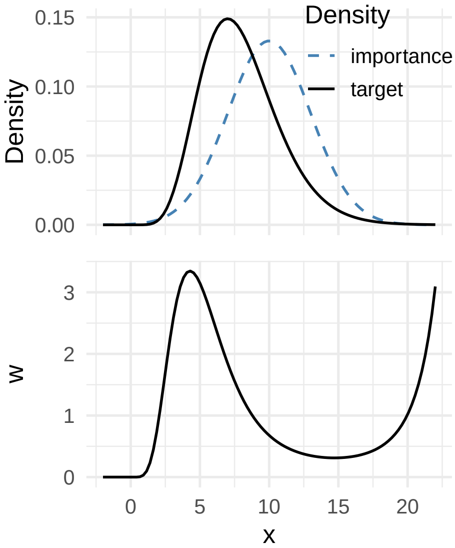
Figure 3: Target and importance densities with corresponding importance weights.

Gamma Importance Sampling

```r
B <- 1000
x <- rnorm(B, 10, 3)
w_star <-
  dgamma(x, 8) / dnorm(x, 10, 3)
mu_hat_IS <-
  cumsum(x * w_star) / (1:B)
mu_hat_IS[B]
```

```
[1] 7.995
```

Close to theoretical value (8).



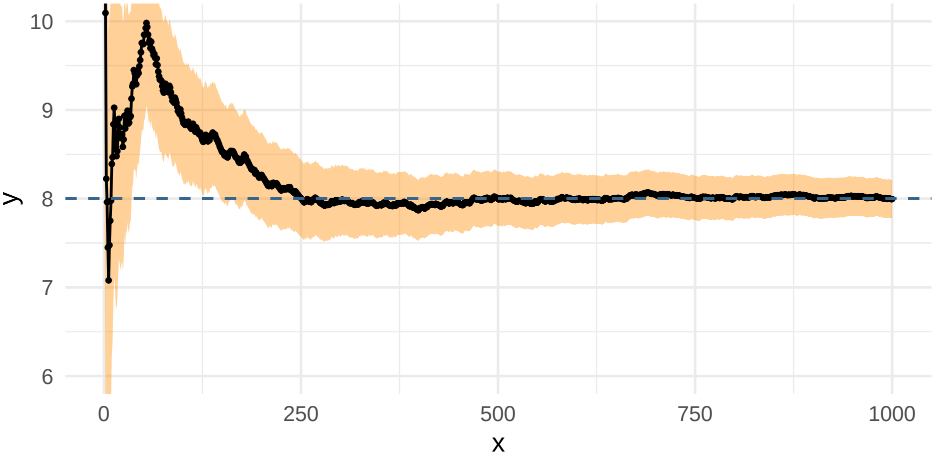


Figure 4: Convergence of the importance sampling estimate of the mean of a Gamma(8,1) distribution using a N(10,3) importance distribution.

Importance Sampling Variance

Again by the LLN

$$\hat{\mu}^*_{\mathrm{IS}} \to \mathrm{E}\, h(X_1) w^*(X_1) = \mu \quad \text{as } n \to \infty.$$

And by the CLT

$$\hat{\mu}^*_{\mathrm{IS}} \overset{\text{approx}}{\sim} \mathcal{N}(\mu, \sigma^{*2}_{\mathrm{IS}}/n)$$

where

$$\sigma^{*2}_{\mathrm{IS}} = \mathrm{Var}\left(h(X_1) w^*(X_1)\right) = \int (h(x) w^*(x) - \mu)^2 g(x)\, dx.$$

The IS variance can be estimated analogously with the MC variance,

$$\hat{\sigma}_{\mathrm{IS}}^{*2} = \frac{1}{n-1} \sum_{i=1}^{n} (h(X_i) w^*(X_i) - \hat{\mu}_{\mathrm{IS}}^*)^2.$$

And a 95% standard confidence interval is

$$\hat{\mu}_{\mathrm{IS}}^* \pm 1.96 \frac{\hat{\sigma}_{\mathrm{IS}}^*}{\sqrt{n}}.$$

We may have $\sigma_{\mathrm{IS}}^{*2} > \sigma_{\mathrm{MC}}^2$ or $\sigma_{\mathrm{IS}}^{*2} < \sigma_{\mathrm{MC}}^2$ depending on h and g.

Goal
Choose g so that $|h(x)|w^*(x)$ becomes as constant as possible.

Gamma Importance Sampling

```
sigma_hat_IS <- sd(x * w_star)
sigma_hat_IS
```

```
[1] 3.5
```

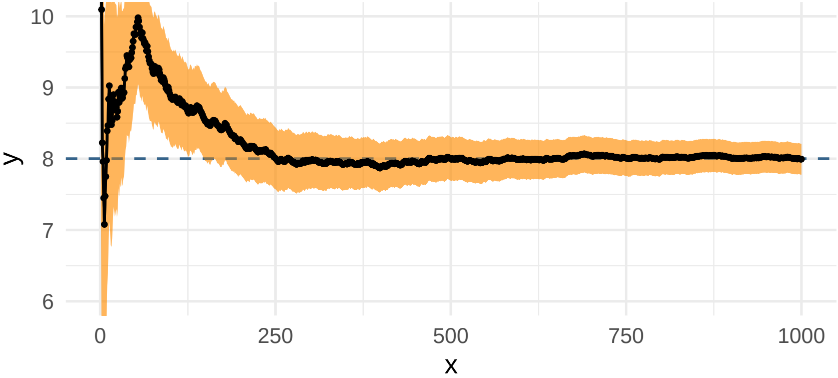


Figure 5: Importance sampling estimate of the mean of a Gamma(8,1).

It is not always the case that the best importance distribution is close to the target distribution.

Example

Suppose we want to estimate

$$\mathrm{E}(X^4) = \int x^4 \, f(x) \, dx = 3$$

where $f(x)$ is the standard normal density.

If we use $g(x) = f(x) = \mathcal{N}(0,1)$, the weights are $w^*(x) = 1$, so the estimator is just the sample mean of x^4.

But $h(x) = x^4$ is large in the tails, so sampling from a normal may not be efficient, as most samples are near zero.

Instead, we can try a heavy-tailed importance distribution, such as Student-t.

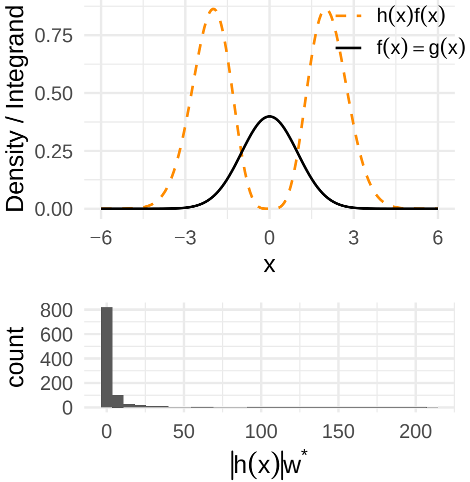


Figure 6: Densities, integrand, and weights for the importance sampling example with a normal importance distribution.

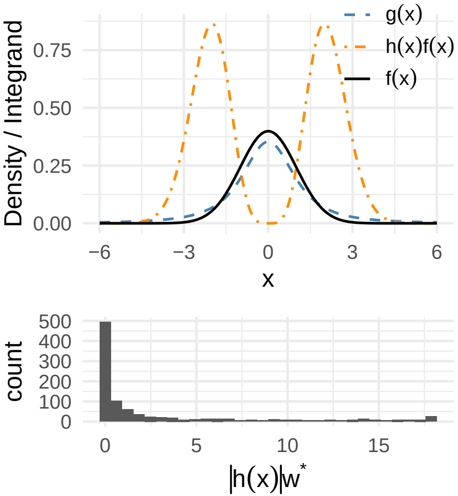


Figure 7: Densities, integrand, and weights for the importance sampling example with Student-t as importance distribution.

Normalized Weights

If $f = c^{-1}q$ with c unknown then

$$c = \int q(x)dx = \int \frac{q(x)}{g(x)} g(x)dx.$$

And

$$\mu = \frac{\int h(x)w^*(x)g(x)\ dx}{\int w^*(x)g(x)\ dx},$$

where

$$w^*(x) = \frac{q(x)}{g(x)}.$$

An importance sampling estimate of μ is thus

$$\hat{\mu}_{\text{IS}} = \frac{\sum_{i=1}^{n} h(X_i) w^*(X_i)}{\sum_{i=1}^{n} w^*(X_i)} = \sum_{i=1}^{n} h(X_i) w(X_i),$$

where $w^*(X_i) = q(X_i)/g(X_i)$ and

$$w(X_i) = \frac{w^*(X_i)}{\sum_{i=1}^{n} w^*(X_i)}$$

are the *normalized weights*.

Works irrespectively of the value of the normalizing constant c, and also if an unnormalized g is used.

Gamma Sampling with Normalized Weights

Let's pretend that we do not know the normalizing constant of the target ($\text{Gamma}(8, 1)$) and importance ($\mathcal{N}(10, 3^2)$) distributions. Then

$$w^*(x) = \frac{x^7 \exp(-x)}{\exp\left(-\frac{(x-10)^2}{18}\right)} = \exp\left(\frac{(x-10)^2}{18} - x + 7\log(x)\right).$$

```r
B <- 1000
x <- rnorm(B, 10, 3)
w_star <- numeric(B)
x_pos <- x[x > 0]
w_star[x > 0] <- exp((x_pos - 10)^2 / 18 - x_pos + 7 * log(x_pos))

mu_hat_IS <- cumsum(x * w_star) / cumsum(w_star)
mu_hat_IS[B] # Theoretical value 8
```

```
[1] 8.102
```

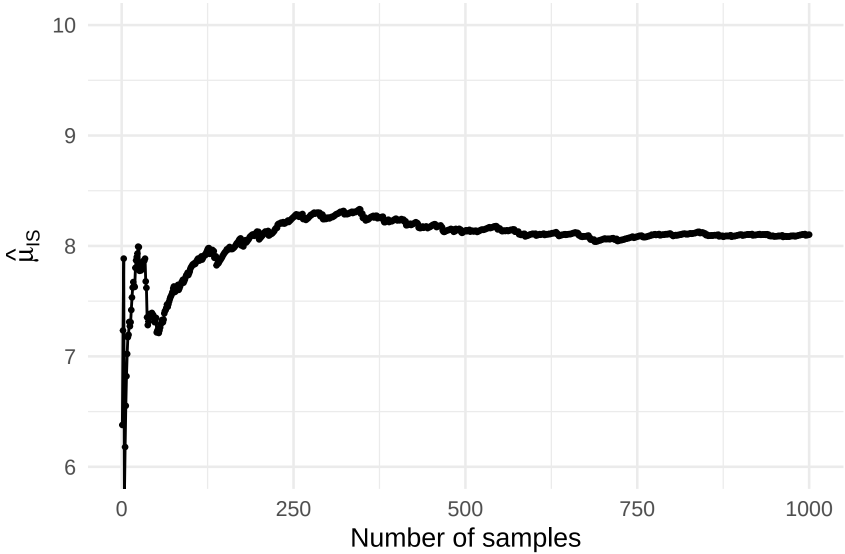


Figure 8: Convergence of the importance sampling estimate of the mean of a Gamma(8,1) distribution using normalized weights.

Variance for IS with Normalized Weights

Computing the variance of IS with normalized weights is more complicated because the estimator is a ratio of random variables.

From the multivariate CLT, we have

$$\frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix} h(X_i)w^*(X_i) \\ w^*(X_i) \end{bmatrix} \overset{\text{approx}}{\sim} \mathcal{N}\left(c\begin{bmatrix}\mu \\ 1\end{bmatrix}, \frac{1}{n}\begin{bmatrix}\sigma^2_{\text{IS}} & \gamma \\ \gamma & \sigma^2_{w^*}\end{bmatrix}\right),$$

where

$$\begin{aligned} \sigma^2_{\text{IS}} &= \text{Var}\left(h(X_1)w^*(X_1)\right) \\ \gamma &= \text{Cov}(h(X_1)w^*(X_1), w^*(X_1)) \\ \sigma^2_{w^*} &= \text{Var}(w^*(X_1)) \end{aligned}$$

We can then apply the Δ-method with $t(x, y) = x/y$.

Observe that $\nabla t(x,y) = \begin{bmatrix} 1/y \\ -x/y^2 \end{bmatrix}$, whence

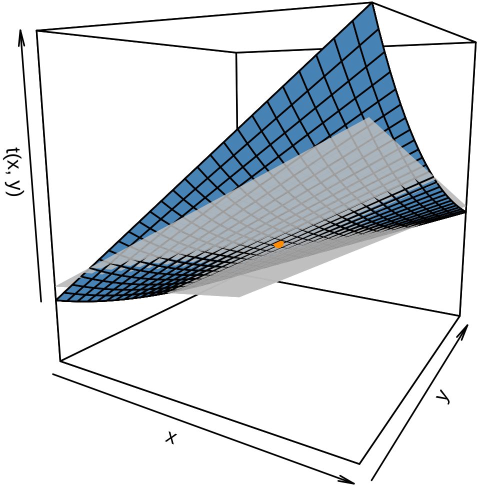
$$\nabla t(c\mu, c)^T \begin{pmatrix} \hat{\sigma}^2_{\mathrm{IS}} & \gamma \\ \gamma & \sigma^2_{w^*} \end{pmatrix} \nabla t(c\mu, c) = \frac{\sigma^2_{\mathrm{IS}} + \mu^2 \sigma^2_{w^*} - 2\mu\gamma}{c^2}.$$

By the Δ-method,

$$\hat{\mu}_{\mathrm{IS}} \overset{\mathrm{approx}}{\sim} \mathcal{N}\left(\mu, \frac{\sigma^2_{\mathrm{IS}} + \mu^2 \sigma^2_{w^*} - 2\mu\gamma}{c^2 n}\right).$$

For $c = 1$, the asymptotic variance can be estimated by plugging in empirical estimates. For $c \neq 1$ it is necessary to estimate c as

$$\hat{c} = \frac{1}{n} \sum_{i=1}^{n} w^*(X_i)$$



t(x, y)
x
y

Gamma Normalized Weights, Variance

```r
c_hat <- mean(w_star)
sigma_hat_IS <- sd(x * w_star)
sigma_hat_w_star <- sd(w_star)
gamma_hat <- cov(x * w_star, w_star)

sigma_hat_IS_w <- sqrt(
  sigma_hat_IS^2 +
    mu_hat_IS[B]^2 * sigma_hat_w_star^2 -
    2 * mu_hat_IS[B] * gamma_hat
) /
  c_hat

sigma_hat_IS_w
```

```
[1] 3.198
```

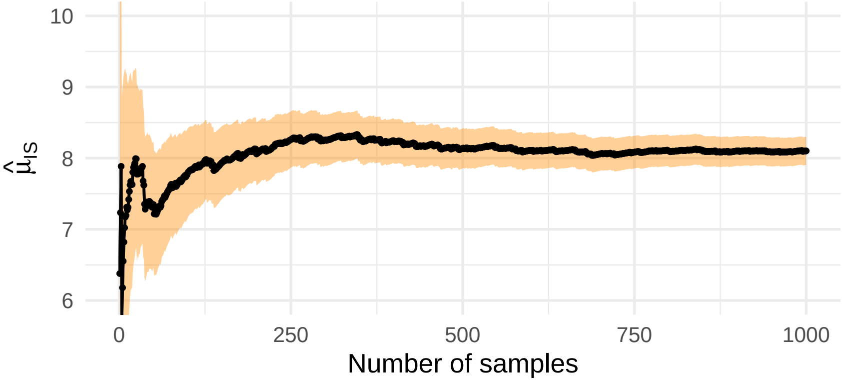


Figure 10: Importance sampling estimate of the mean of a Gamma(8,1) distribution with 95% confidence bands using normalized weights.

When to Use Importance Sampling?

Rare Events

Estimating probabilities or expectations involving rare events.

High-Variance Monte Carlo

Importance sampling can reduce variance by focusing samples where the integrand is large.

Difficult-to-Sample Targets

When direct sampling from the target distribution is hard, but sampling from a related importance distribution is easy.

Poor Importance Distribution

If the importance distribution does not cover regions where the target is large, weights become highly variable and estimates unreliable.

Remedies

Increase sample size (if feasible).

Redesign importance density to better match the target density.

Diagnostics for Importance Sampling

Weight Distribution

Visualize the weight distribution with, for example, histograms or box plots to identify degeneracies.

Effective Sample Size (ESS)

ESS measures how many samples are effectively contributing to the estimate.

Coefficient of Variation (CV)

CV quantifies the relative variability of the weights.

Comparison with Standard Monte Carlo

Comparing with standard Monte Carlo shows whether importance sampling actually improves efficiency.

Suppose that

$$X \sim \mathcal{N}(0, I_2)$$

and that we want to estimate $\mathrm{E}[f(X)]$ where $f(X)$ is large near the origin.

If we use a importance density $g(x)$ from

$$\mathcal{N}(\mu, I_2)$$

with large $|\mu|$, most samples are far from the origin, so $f(X)$ is nearly zero and weights are highly variable.

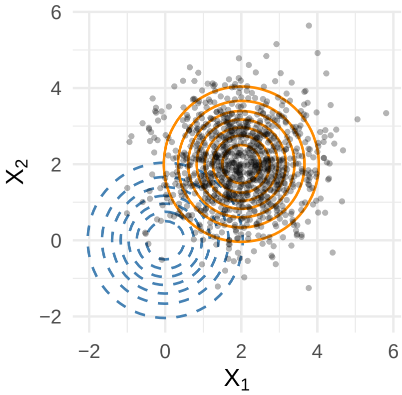


Figure 11: Target (blue) and importance (orange) densities with samples from the importance distribution.

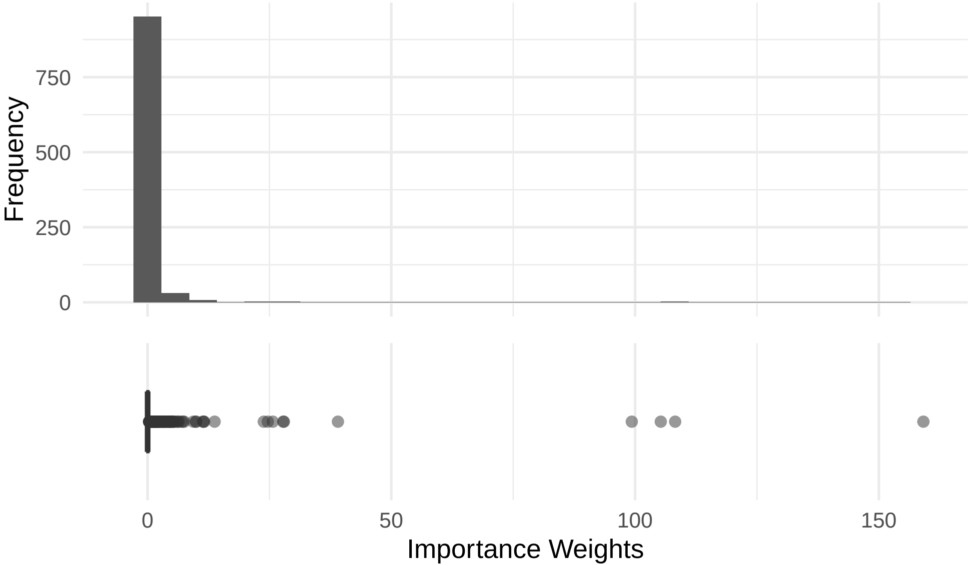


Figure 12: Histogram and boxplot of importance weights, showing weight degeneracy.

Effective Sample Size (ESS)

The effective sample size (ESS) quantifies how many samples effectively contribute to the estimate:

$$\text{ESS} = \frac{1}{\sum_{i=1}^{n} w_i^2} = \frac{\left(\sum_{i=1}^{n} w_i^*\right)^2}{\sum_{i=1}^{n} (w_i^*)^2}$$

where w_i are the normalized importance weights.

If all weights are equal, $\text{ESS} = n$. If one weight dominates, $\text{ESS} \approx 1$, and the estimate is unreliable.

Gamma Importance Example

```
ess <- (sum(w_star))^2 / sum(w_star^2)
ess # out of n = 1000 samples
```

```
[1] 14.46
```

Coefficient of Variation (CV)

The coefficient of variation (CV) of the weights measures their relative variability.

It is defined as

$$\mathrm{CV} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(w_i - \bar{w})^2}}{\bar{w}}$$

where $\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$ is the mean weight.

Interpretation

A high CV indicates that a few samples dominate the estimate, leading to high variance.

CV near 0 means weights are nearly uniform—great! As a rule of thumb, CV below 1 $(\mathrm{ESS} > n/2)$ is good.

Connection to ESS

There is a direct relationship between ESS and CV:

$$\mathrm{ESS} = \frac{n}{1 + \mathrm{CV}^2}.$$

Choosing the Importance Distribution

We want low-variance estimates of

$$I = \int h(x) f(x) \, dx$$

The importance distribution determines the weights

$$w^*(x) = \frac{f(x)}{g(x)}$$

and the variance of these weights drives the variance of the IS estimator.

Principles for Choosing the Importance Distribution

Cover the Support

Ensure the importance distribution covers all regions where the target distribution is significant.

Match the Bulk

The mean and spread of the importance distribution should roughly match the regions of large $|h(x)|f(x)$.

Give it Heavy-Enough Tails

The importance distribution should have heavier tails than the target to avoid extreme weights.

Common Strategies

Laplace Approximation
Use a Gaussian approximation centered at the mode of the target distribution.

Adaptive Importance Sampling
Iteratively refine the importance distribution based on pilot runs.

Defensive Mixtures
Combine a well-chosen importance distribution with a broader distribution to ensure coverage of the entire target support.

Mixture Distributions
Use a mixture of distributions to capture multiple modes or complex shapes.

Laplace Approximation

Basic Idea
Take a Gaussian approximation of the target density at its mode x^* as the importance distribution.

Motivation
Connection comes from the second-order Taylor expansion of $\log f(x)$ at the mode x^*,

$$\log f(x) \approx \log f(x^*) - \frac{1}{2}(x - x^*)^T H (x - x^*)$$

where H is the Hessian of $-\log f(x)$ at x^*.

Exponentiating both sides gives:

$$f(x) \approx f(x^*) \exp\left(-\frac{1}{2}(x - x^*)^T H (x - x^*)\right)$$

Steps

1. Find the mode $x^* = \arg\max_x f(x)$.
2. Compute the Hessian H of the first-order Taylor approximation of $-\log f(x)$ around x^*.
3. Use $\mathcal{N}(x^*, H^{-1})$ as importance distribution.

When to Use

Best for unimodal, smooth, and symmetric target densities.

Example: Gamma Distribution

The density of the $\text{Gamma}(3,1)$ distribution is

$$f(x) = \frac{1}{2}x^2 e^{-x}, \quad x > 0.$$

The mode is at $x^* = 2$ and the Hessian of $-\log f(x)$ at x^* is

$$H = \frac{\alpha - 1}{(x^*)^2} = \frac{1}{2}.$$

So the Laplace approximation is $\mathcal{N}(2, 2^2)$.

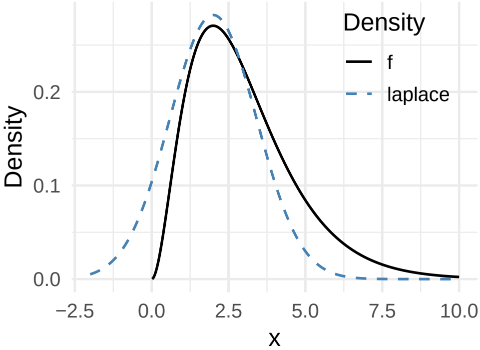


Figure 13: Target (solid black) and Laplace approximation (dashed blue) densities for a Gamma(3,1) distribution.

Adaptive importance sampling iteratively refines the importance distribution based on the samples obtained.

Basic Idea
Start with an initial importance distribution and refine it based on samples and weights.

Steps

1. Choose an initial importance distribution $g_0(x)$.
2. Draw samples from g_0 and compute weights.
3. Update the importance distribution parameters based on weighted samples.
4. Repeat steps 2-3 until convergence.

One simple adaptive strategy is to use moment-matching to update a Gaussian importance distribution.

At iteration t, draw samples $x_1, \dots, x_n \sim g_t(x)$ and compute weights

$$w_i^* = \frac{f(X_i)}{g_t(X_i)}.$$

Then update the importance distribution parameters as

$$\mu_{t+1} = \frac{\sum_{i=1}^n w_i^* x_i}{\sum_{i=1}^n w_i^*},$$
$$\sigma_{t+1}^2 = \frac{\sum_{i=1}^n w_i^* (x_i - \mu_{t+1})^2}{\sum_{i=1}^n w_i^*}.$$

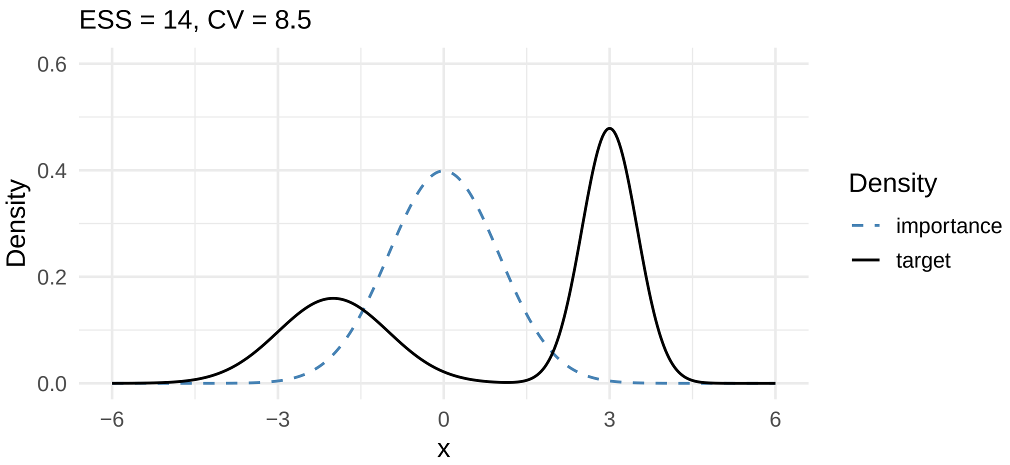


Figure 14: Adaptive importance sampling step 1.

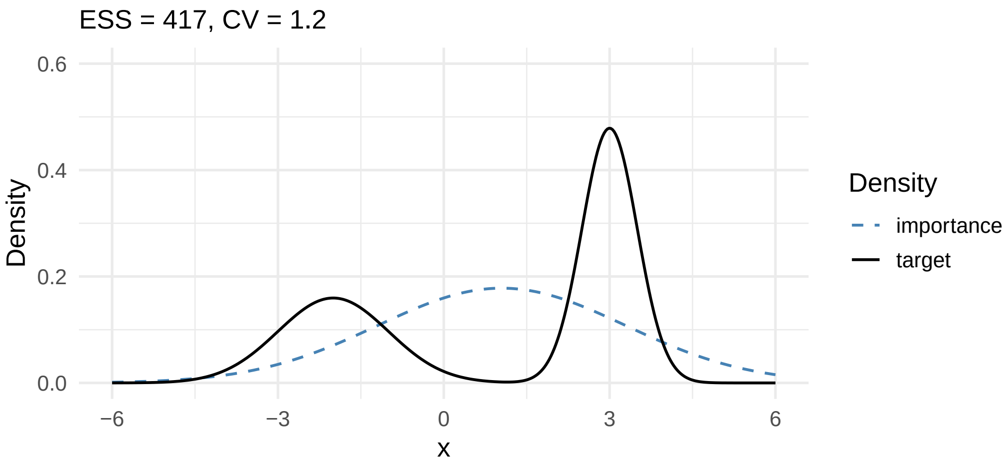


Figure 15: Adaptive importance sampling, step 2.

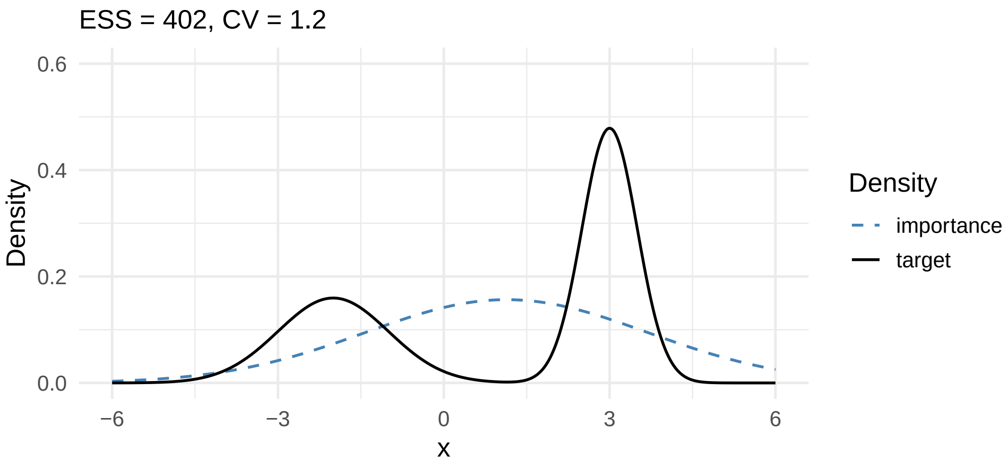


Figure 16: Adaptive importance sampling, step 3.

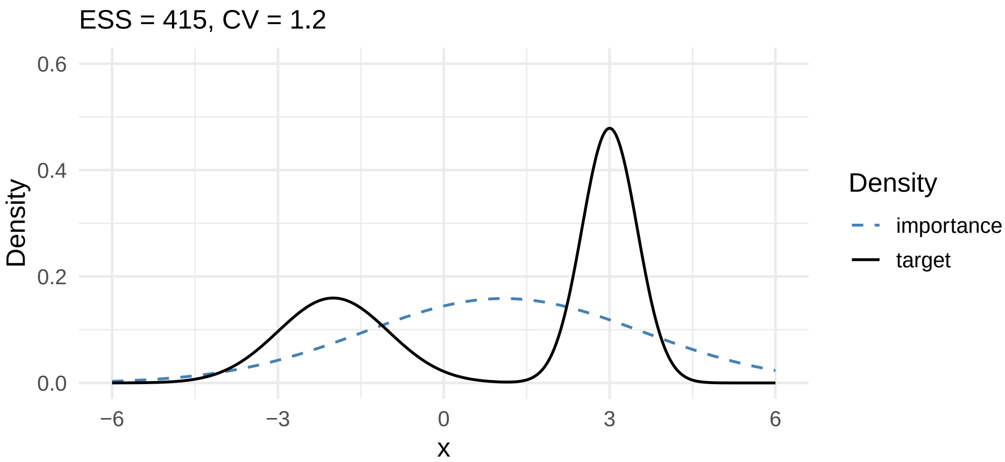


Figure 17: Adaptive importance sampling, step 4.

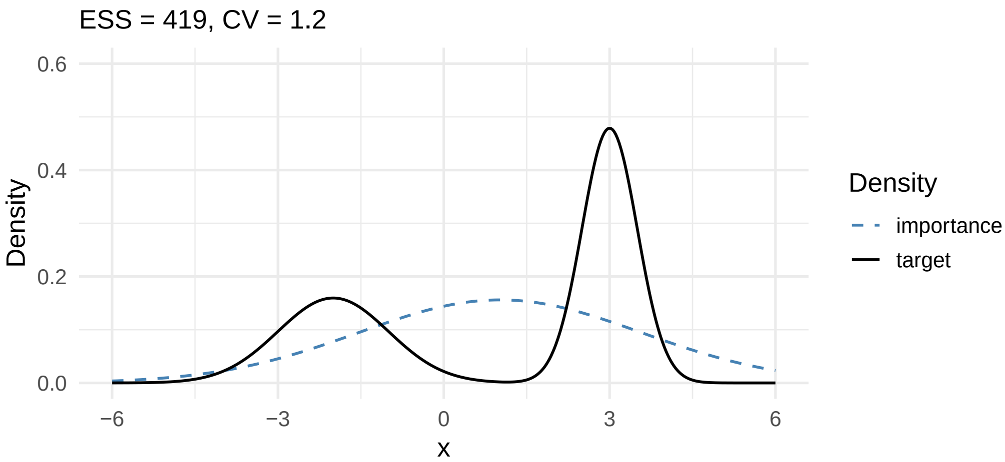


Figure 18: Adaptive importance sampling, step 5.

Defensive importance sampling uses a mixture of a well-chosen importance distribution and a broader distribution to ensure coverage of the entire target support.

Recall the Gamma(8,1) example where we used $g(x) = \mathcal{N}(10, 3^2)$.

Instead we can use a mixture

$$g_{\mathrm{mix}}(x) = 0.8 \cdot \mathcal{N}(10, 3^2) + 0.2 \cdot \mathcal{N}(10, 10^2)$$

with heavier tails to ensure coverage.

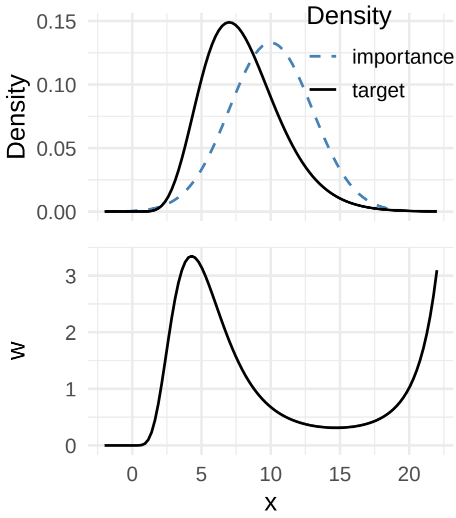


Figure 19: Target (black) and importance (blue) densities with samples from the importance distribution.

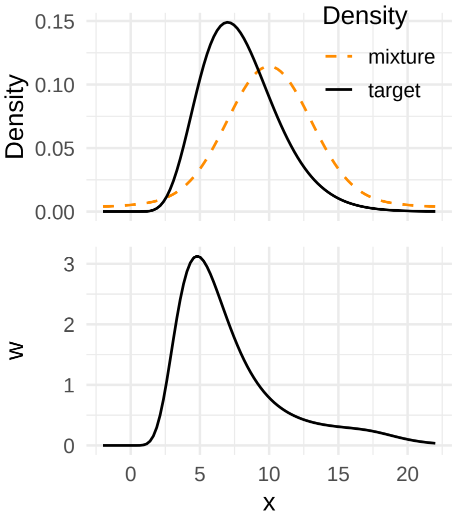


Figure 20: Target (black) and importance (orange) densities with samples from the defensive importance mixture distribution.

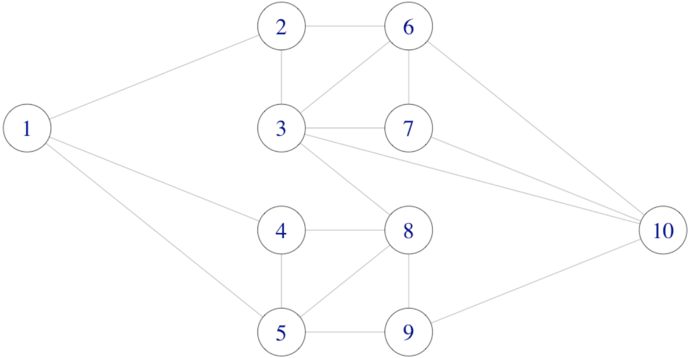


Figure 21: A network graph. Different edges "fail" independently with probability p.

Problem: What is the probability that nodes 1 and 10 are disconnected?

We can represent a graph using an adjacency matrix.

```
A # Graph adjacency matrix
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1    0    1    1    0    0    0    0     0
 [2,]    1    0    1    0    0    1    0    0    0     0
 [3,]    0    1    0    0    0    1    1    1    0     1
 [4,]    1    0    0    0    1    0    0    1    0     0
 [5,]    1    0    0    1    0    0    0    1    1     0
 [6,]    0    1    1    0    0    0    1    0    0     1
 [7,]    0    0    1    0    0    1    0    0    0     1
 [8,]    0    0    1    1    1    0    0    0    1     0
 [9,]    0    0    0    0    1    0    0    1    0     1
[10,]    0    0    1    0    0    1    1    0    1     0
```

```
Aup <- A
Aup[lower.tri(Aup)] <- 0
```

Check Connectivity

We begin with a function that checks if nodes 1 and 10 are disconnected.

```
discon <- function(Aup) {
  A <- Matrix::forceSymmetric(Aup, "U")
  i <- 3
  Apow <- A %*% A %*% A # A^3

  while (Apow[1, 10] == 0 && i < 9) {
    Apow <- Apow %*% A
    i <- i + 1
  }

  Apow[1, 10] == 0 # TRUE if nodes 1 and 10 are not connected
}
```

Sampling a Graph

```r
sim_net <- function(Aup, p) {
  ones <- which(Aup == 1)
  Aup[ones] <- sample(
    c(0, 1),
    length(ones),
    replace = TRUE,
    prob = c(p, 1 - p)
  )
  Aup
}
```

```r
bench::bench_time(replicate(1e5, sim_net(Aup, 0.5)))
```

```
process    real
  5.68s   6.11s
```

Estimating Probability of Nodes 1 and 10 Disconnected

```
set.seed(27092016)
n <- 1e5
tmp <- replicate(n, discon(sim_net(Aup, 0.05)))

mu_hat <- mean(tmp)
mu_hat
```

```
[1] 0.00034
```

Estimate with confidence interval using $\sigma^2 = \mu(1 - \mu)$.

```
mu_hat + 1.96 * sqrt(mu_hat * (1 - mu_hat) / n) * c(-1, 0, 1)
```

```
[1] 0.0002257 0.0003400 0.0004543
```

Importance Sampling

We will simulate with failure probability p_0 and compute the importance weights.

```
weights <- function(Aup, Aup0, p0, p) {
  w <- discon(Aup0)

  if (w) {
    s <- sum(Aup0)
    w <- (p / p0)^18 * (p0 * (1 - p) / (p * (1 - p0)))^s
  }

  as.numeric(w)
}
```

For the IS estimator the weights will be multiplied by the indicator that 1 and 10 are disconnected.

Estimating Probability of Nodes 1 and 10 Disconnected

```
tmp <- replicate(n, weights(Aup, sim_net(Aup, 0.2), 0.2, 0.05))
mu_hat_IS <- mean(tmp)
mu_hat_IS
```

```
[1] 0.0002819
```

Confidence interval using empirical variance estimate $\hat{\sigma}^2$.

```
mu_hat_IS + 1.96 * sd(tmp) / sqrt(n) * c(-1, 0, 1)
```

```
[1] 0.0002493 0.0002819 0.0003144
```

```
c(sd(tmp), sqrt(mu_hat * (1 - mu_hat))) # Est. standard deviations
```

```
[1] 0.005257 0.018436
```

Comparison

The ratio of variances is estimated as

```
mu_hat * (1 - mu_hat) / var(tmp)
```

```
[1] 12.3
```

Need about 12 times more naive samples compared to importance sampling for same precision.

Benchmarking would should that the extra computing time for importance sampling is small compared to the reduction of variance.

It is worth the coding effort if used repeatedly, but not if it is a one-off computation.

Enumeration

There are $2^{18} = 262,144$ different networks with any number of the edges failing, so complete enumeration is possible.

```r
ones <- which(Aup == 1)
Atmp <- Aup
p <- 0.05
prob <- numeric(2^18)

for (i in 0:(2^18 - 1)) {
  on <- as.numeric(intToBits(i)[1:18])
  Atmp[ones] <- on

  if (discon(Atmp)) {
    s <- sum(on)
    prob[i + 1] <- p^(18 - s) * (1 - p)^s
  }
}
```

Probability of 1 and 10 Being Disconnected

Here is the true value:

```
sum(prob)
```

```
[1] 0.0002883
```

And here's the MC estimate with confidence interval:

```
mu_hat + 1.96 * sqrt(mu_hat * (1 - mu_hat) / n) * c(-1, 0, 1)
```

```
[1] 0.0002257 0.0003400 0.0004543
```

And the IS estimate with confidence interval:

```
mu_hat_IS + 1.96 * sd(tmp) / sqrt(n) * c(-1, 0, 1)
```

```
[1] 0.0002493 0.0002819 0.0003144
```

Summary

Monte Carlo Methods
Estimate integrals and means using random sampling when analytic solutions are not possible.

Importance Sampling
Focus samples where they matter most, but choose your proposal carefully to avoid large or variable weights.

Diagnostics
Examine weight distribution and check effective sample size to ensure reliable estimates.

Picking Importance Distributions
It's hard! But use adaptive and defensive methods to help fix poor coverage and improve robustness.

Exercise: Von Mises Importance Sampling

Step 1

Implement an importance sampler for computing the mean of the von Mises distribution. Plot the result and check convergence towards μ.

```r
dvmises <- function(x, mu = 0, kappa = 2) {
  exp(kappa * cos(x - mu)) / (2 * pi * besselI(kappa, 0))
}
```

Step 2

Compute the estimate of the mean and its standard error.

Step 3

Compute the effective sample size and coefficient of variation of the importance weights. Experiment with different importance densities to see if you find a better one.