Monte Carlo Integration

Sujit Sandipan Chaugule[1*], Dr. Amiya Ranjan Bhowmick[2]

[1*]Department of Pharmaceutical Sciences and Technology, Institute of Chemical Technology, Mumbai

[2]Department of Mathematics, Institute of Chemical Technology, Mumbai

1. Introduction

Two major classes of numerical problems that arise in statistical inference are optimization problems and integration problems. Indeed, numerous examples (see Rubinstein, 1981, Gentle, 2002, or Robert, 2001) show that it is not always possible to analytically compute the estimators associated with a given paradigm (maximum likelihood, Bayes, method of moments, etc.). Thus, whatever the type of statistical inference, we are often led to consider numerical solutions. The previous chapter introduced a number of methods for the computer generation of random variables with any given distribution and hence provides a basis for the construction of solutions to our statistical problems. A general solution is indeed to use simulation, of either the true or some substitute distributions, to calculate the quantities of interest. In the setup of decision theory, whether it is classical or Bayesian, this solution is natural since risks and Bayes estimators involve integrals with respect to probability distributions.Note that the possibility of producing an almost infinite number of random variables distributed according to a given distribution gives us access to the use of frequentist and asymptotic results much more easily than in the usual inferential settings, where the sample size is most often fixed. One can therefore apply probabilistic results such as the Law of Large Numbers or the Central Limit Theorem, since they allow assessment of the convergence of simulation methods (which is equivalent to the deterministic bounds used by numerical approaches).

2. Monte Carlo Integration

Suppose we want to compute the following integral

$$I = \int_0^1 e^{-x^2} dx.$$

We can write the above integral as an expectation. Let $f(x) = I_{(0,1)}(x)$ be the uniform $(0,1)$ density function.

$$\int_0^1 e^{-x^2} dx = \int_{-\infty}^{\infty} e^{-x^2} f(x) dx = \mathbb{E}_{X\sim \text{uniform}(0,1)} \left(e^{-X^2}\right).$$

Suppose, we simulate n numbers $X_1, X_2, \ldots, X_n$ from the uniform $(0,1)$ density function, and compute $Y_1 = e^{-X_1^2}, Y_2 = e^{-X_2^2}, \ldots, Y_n = e^{-X_n^2}$, and take the average of Y_i's. Then by the Weak Law of Large Numbers (WLLN),

$$\overline{Y_n} \rightarrow \mathbb{E}(Y) = \mathbb{E}\left(e^{-X^2}\right), \quad \text{in probability}.$$

In [1]:
```julia
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [2]:
```julia
g(x) = exp.(-x.^2) # define the function
```

Out[2]:
```
g (generic function with 1 method)
```

In [3]:
```julia
exact_integral, er = quadgk(g, 0,1)
println("The exact integral is:", exact_integral)
```

```
The exact integral is:0.746824132812427
```

In [4]:
```julia
n = 1000  # no of variable
x = rand(Uniform(0,1),n)
y = g(x)
println("the average of y value: ", mean(y))
```

```
the average of y value: 0.7461736018235385
```

In [5]:
```julia
n_vals = 1:1000
approx_integral = zeros(length(n_vals))

for n in n_vals
    x = rand(Uniform(0,1),n)
    y = g(x)
    approx_integral[n] = mean(y)
end
```

In [6]:
```julia
plot(n_vals, approx_integral, color = "magenta", lw = 2, ylabel= L"I_n",
    xlabel = L"n", label = "")
hline!([exact_integral], color = "blue", linestyle = :dash, lw = 2,label = "")
```

Out[6]:

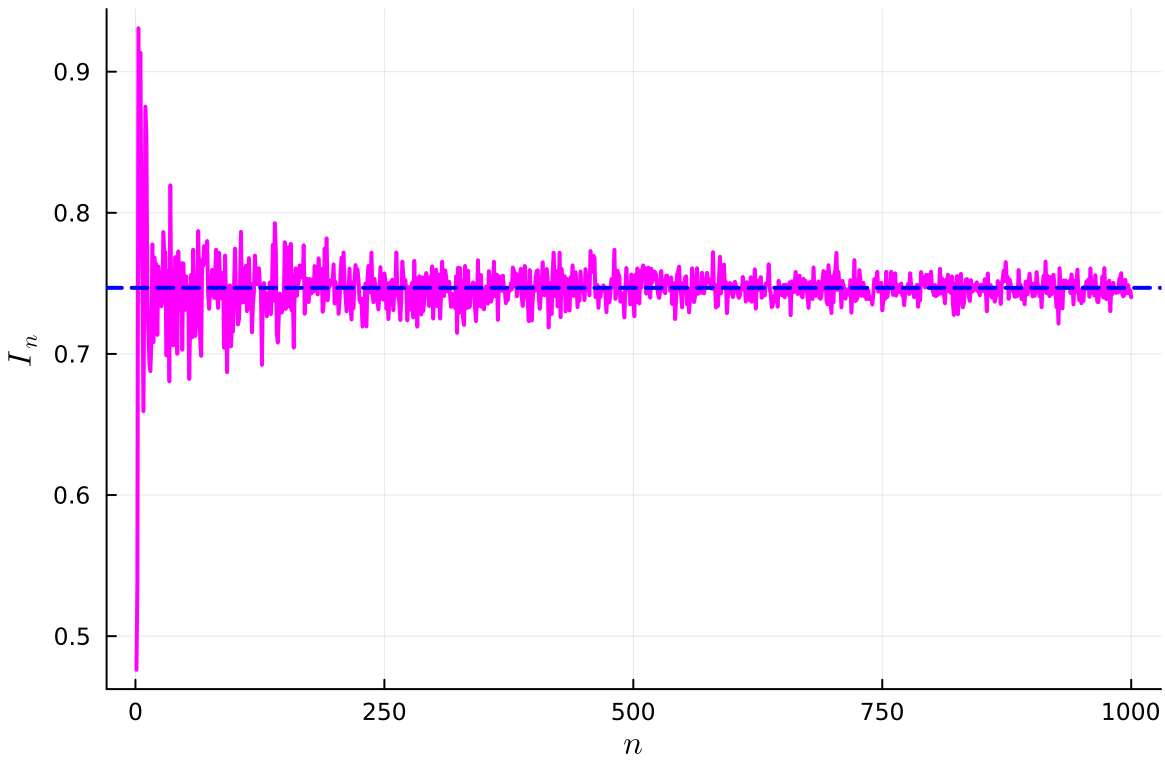

Figure 1: As $n \to \infty$, the Monte Carlo integral converges to the true integral.

3. Sampling distribution of I_n

Set $\overline{Y_n} = I_n$, the approximate integral based on a sample of size n. For every n, I_n is a random variable, and we shall see that the sampling distribution of $\hat{I_n}$ is well approximated by the normal density function for large n.
For $n = 10$, we simulate the sampling distribution of $\hat{I_n}$. Following the previous scheme, we replicate the process 1000 times and obtain 1000 Monte Carlo estimates of the integral. Then, visualize the sampling distribution of the estimator using histogram approximation.

In [7]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase
```

In [8]:
```
rep = 1000
n = 10
I_n = zeros(rep)
g(x) = exp.(-x.^2) # define the function
```

Out[8]:
```
g (generic function with 1 method)
```

In [9]:
```
for i in 1:rep
    x = rand(Uniform(0,1),n)
    I_n[i] = mean(g(x))
end
```

In [10]:
```
histogram(I_n, normalize = true, bins = 30,title= "n = $n",
         xlabel = L"\widehat{I_n}",label = "")
x = range(minimum(I_n), stop=maximum(I_n), length=500)
normal_curve = pdf.(Normal(exact_integral, std(I_n)), x)
plot!(x, normal_curve, color="red", lw=2, label="")
```

Out[10]:

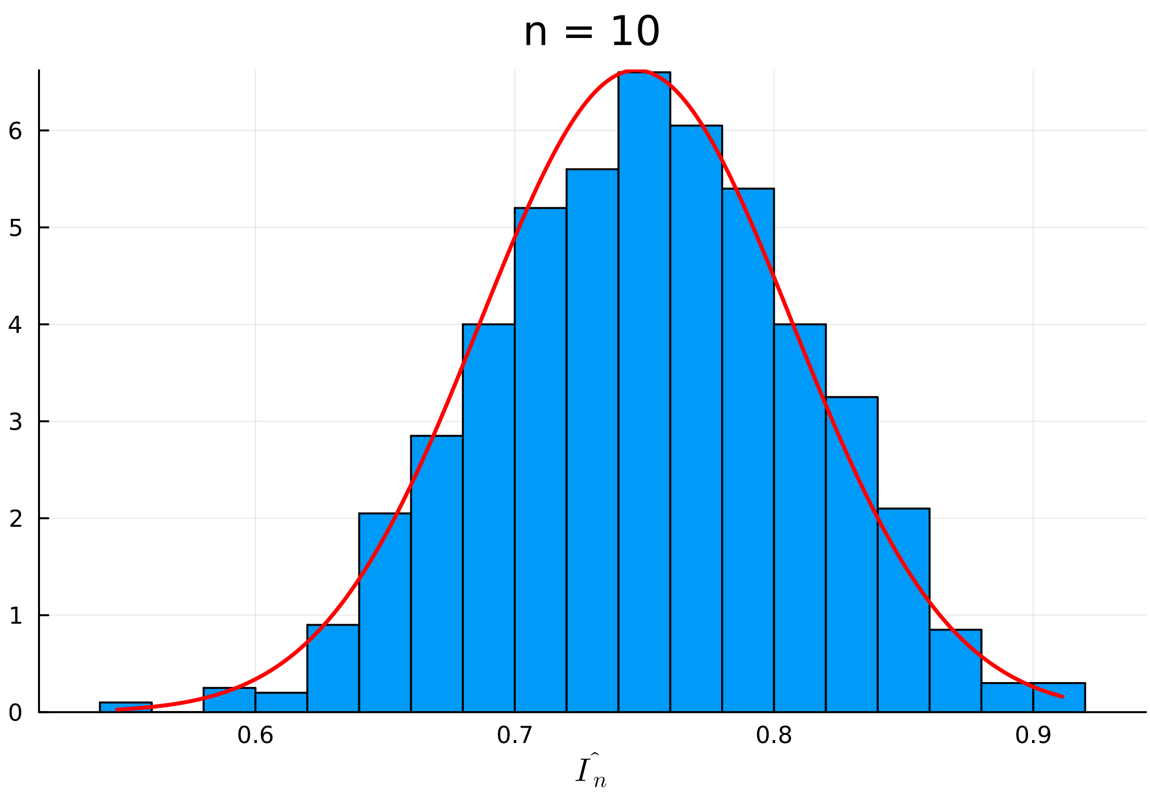

Figure 2: The sampling distribution of $\widehat{I_n}$ is well approximated by the normal distribution for large n

Let us generalize the problem.

Step 1: - Suppose that we want to solve the following integral, how will you proceed?

$$I = \int_a^b g(x)dx$$

Step 2: - We can write this integral as an expectation of a function of a random variable in the following way.

$$\int_a^b g(x)dx = (b-a)\int_a^b g(x)\cdot\frac{1}{b-a}dx = (b-a)\int_{-\infty}^{\infty} g(x)f_{[a,b]}(x)dx,$$

where $f_{[a,b]}(x)$ is the density function of a $\text{uniform}(a,b)$ random variable. Thus,

$$I = (b-a)\mathbb{E}_{X\sim\text{uniform}(a,b)}\left[g(X)\right]$$

Step 3: - If we have a random sample of size n, $X_1, X_2, \ldots, X_n$, and g is a function, then

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \to \mathbb{E}[g(X)] \quad \text{in probability.}$$

Step 4: - If we extend our thought a little bit further, the expectation is nothing but an integral. So basically, we are approximating an integral $\int_{-\infty}^{\infty} g(x)f(x)dx$.

Step 5: - Implementation We will simulate $X_1, X_2, \ldots, X_n \sim \text{uniform}(a,b)$, and then compute $g(X_1), g(X_2), \ldots, g(X_n)$, then take the average

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i),$$

which will be close to $\mathbb{E}[g(X)]$. From the rule of convergence in probability,

$$(b-a)\frac{1}{n}\sum_{i=1}^{n} g(X_i) \to (b-a)\mathbb{E}[g(X)] = \int_a^b g(x)dx.$$

In [11]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [12]:
```
a = 1.5
b = 2.5
g(x) = x.^6 .* exp.(-x.^2)
x_vals = range(0, 6, length=500)  # Range for the full curve
plot(x_vals, g.(x_vals), color=:red, lw=2, label="g(x)",
    xlabel="x", ylabel="g(x)")

plot(g, 0, 6, color="red", lw=2, xlabel=L"x", ylabel=L"g(x)", label="")
x_vals = range(a, b, step=0.01)
y_vals = g.(x_vals)
plot!(x_vals, y_vals, fill=(0, :grey, 0.5), lw=0, label="")
plot!(g, 0, 6, color="red", lw=2, label="")
```

Out[12]:

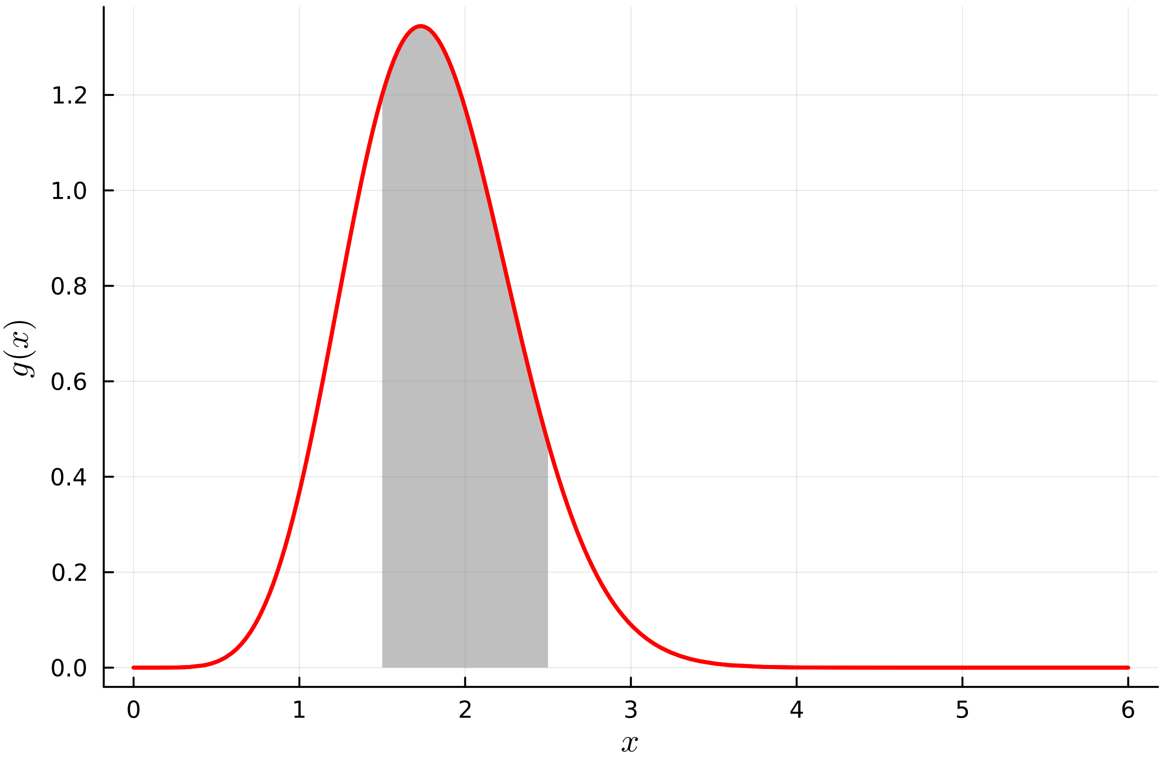

Figure 3: Required integral is the area of the shaded region which can be obtained by integrating the function $g(x)$ in the interval (a, b)

In [13]:
```
# Compute the exact integral
exact_integral = quadgk(g, a, b)[1]

println("The area of the shaded region is: ", exact_integral)
```

```
The area of the shaded region is: 1.05590836146286
```

Required integral is the area of the shaded region which can be obtained by integrating the function $g(x)$ in the interval (a, b)

In [14]:
```
n = 1000 # sample size
x = rand(Uniform(a,b), n) # simulate from uniform(a,b)
y = g(x)
I_n = (b-a)*mean(y)
println("The approx value of the integral by Monte carlo method:",
    I_n)
```

```
The approx value of the integral by Monte carlo method:1.0463472080472724
```

In the following code, we check the convergence of the approximation by the Monte Carlo Method

In [15]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [16]:
```
a = 2  # lower limit
b = 4  # upper limit
g(x) = x .^ 6 .* exp.(-x .^ 2)
```

Out[16]:
```
g (generic function with 1 method)
```

In [17]:
```
exact_integral, er = quadgk(g, a, b)
```

Out[17]: (0.5525956440284987, 1.5878329900909094e-11)

In [18]:
```julia
n_vals = 1:1000
I_n = zeros(length(n_vals))

for n in n_vals
    x = rand(Uniform(a, b), n)  # Simulate from Uniform(a, b)
    I_n[n] = (b - a) * mean(g(x))  # Monte Carlo estimate
end
```

In [19]:
```julia
p1 = plot(n_vals, I_n, color = "magenta", xlabel = L"n", ylabel = L"I_n",
    label = "", lw = 2)
hline!([exact_integral], color = "blue", lw = 2, linestyle = :dash,
     label = L"I")

p2 = plot(n_vals, I_n .- exact_integral, color = "red", xlabel = L"n",
    ylabel = L"{I_n} - I", label = "", lw = 2)
hline!([0], color = "blue", lw = 2, linestyle = :dash, label = "")

plot(p1, p2, layout = (1, 2))
```

Out[19]:

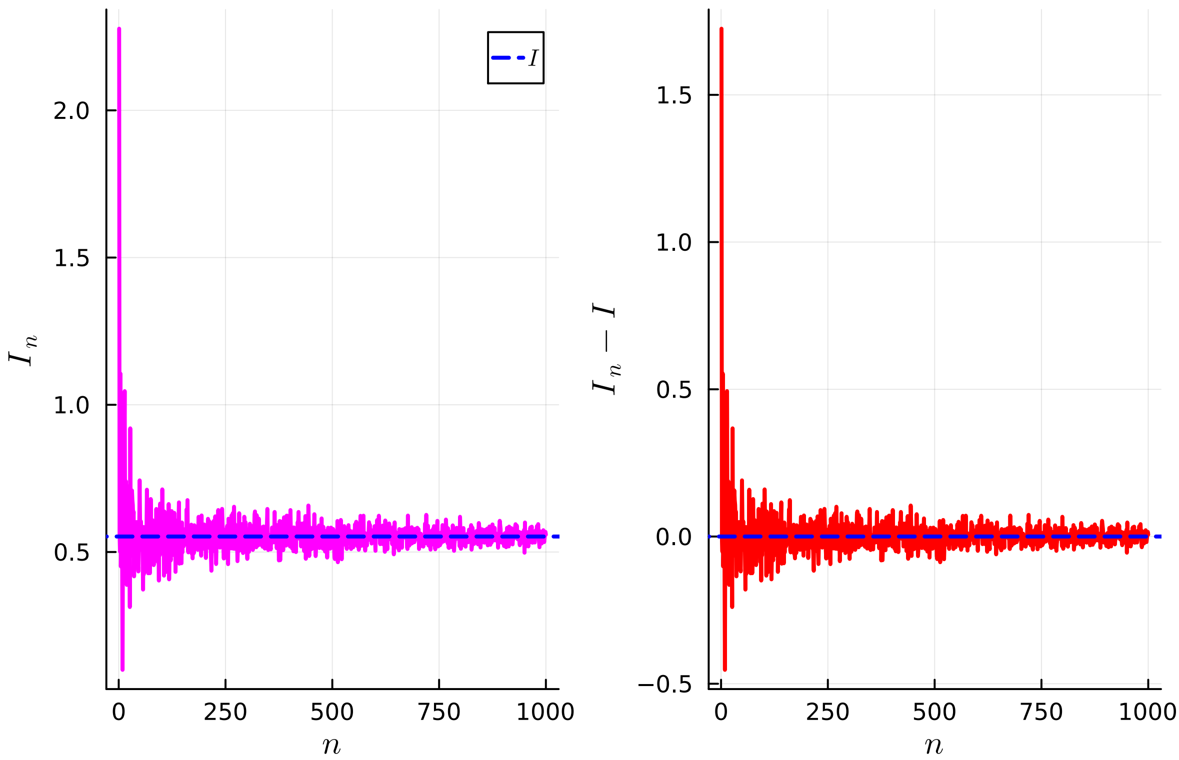

Figure 4: As $n \to \infty$ the approximate integral converges to the true integral. The right panel depicts the difference between the approximate integral and exact integral. The errors are centered around 0 and as the number of simulation increases, the errors are getting highly concentrated at 0.

The idea of Monte Carlo integration is to convert the integral to the expectation of some function of a random variable. It can be easily understood that there can be multiple ways to convert the integral to an expectation. For example if $f_1(x)$ and $f_2(x)$ are two density functions having the same support $(0, \infty)$

Then the integral

$\int_0^\infty \phi(x)dx$ can be expressed as

$$\int_0^\infty \phi(x)dx = \begin{cases} \int_0^\infty \frac{\phi(x)}{f_1(x)} f_1(x)dx = \mathbb{E}_{X\sim f_1}\left(\frac{\phi(X)}{f_1(X)}\right) \\ \int_0^\infty \frac{\phi(x)}{f_2(x)} f_2(x)dx = \mathbb{E}_{X\sim f_2}\left(\frac{\phi(X)}{f_2(X)}\right) \end{cases}$$

provided both the expectations exist.

Suppose that we want to compute integrals of the following form:

$$\int_a^\infty g(x)dx$$

and the integral is finite. How can we apply the concepts of convergence in probability to compute the integral? The idea is very simple. You just have to write down the integral in the form of expectation of some random variable. Suppose that $a = 0, g(x) = e^{-x^3}$

$$\int_0^\infty e^{-x^3}dx = \int_0^\infty e^x e^{-x^3} e^{-x} dx$$

$$= \int_0^\infty e^{x-x^3} f(x)dx, \quad f \sim \text{exponential}(1)$$

$$= \mathbb{E}\left(e^{X-X^3}\right), \quad X \sim \text{exponential}(1)$$

In [20]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [21]:
```
g(x) = exp.(-x.^3).*(x>0)
val_integral, er = quadgk(g,0,Inf)
h(x) = exp.(x.- x.^3)
n = 1000
x = rand(Exponential(1), n)
println("The approximate value of the integral is: ",
    mean(h(x)))
```

```
The approximate value of the integral is: 0.8643435670897627
```

In [22]:
```
println(" Let us check the convergence of the sequence for increasing sample size n")
```

```
 Let us check the convergence of the sequence for increasing sample size n
```

In [23]:
```
n_vals = 1:1000
approx_integral = zeros(length(n_vals))

for n in n_vals
    x = rand(Exponential(1), n)
    approx_integral[n] = mean(h(x))
end
```

In [24]:
```
plot(n_vals, approx_integral, color = "grey", lw = 2,
    xlabel = L"Sample~size(n)", ylabel = L"approx.integral",
    label = "")
hline!([val_integral], color = "blue", linestyle = :dash,
    lw = 2, label = "" )
```

Out[24]:

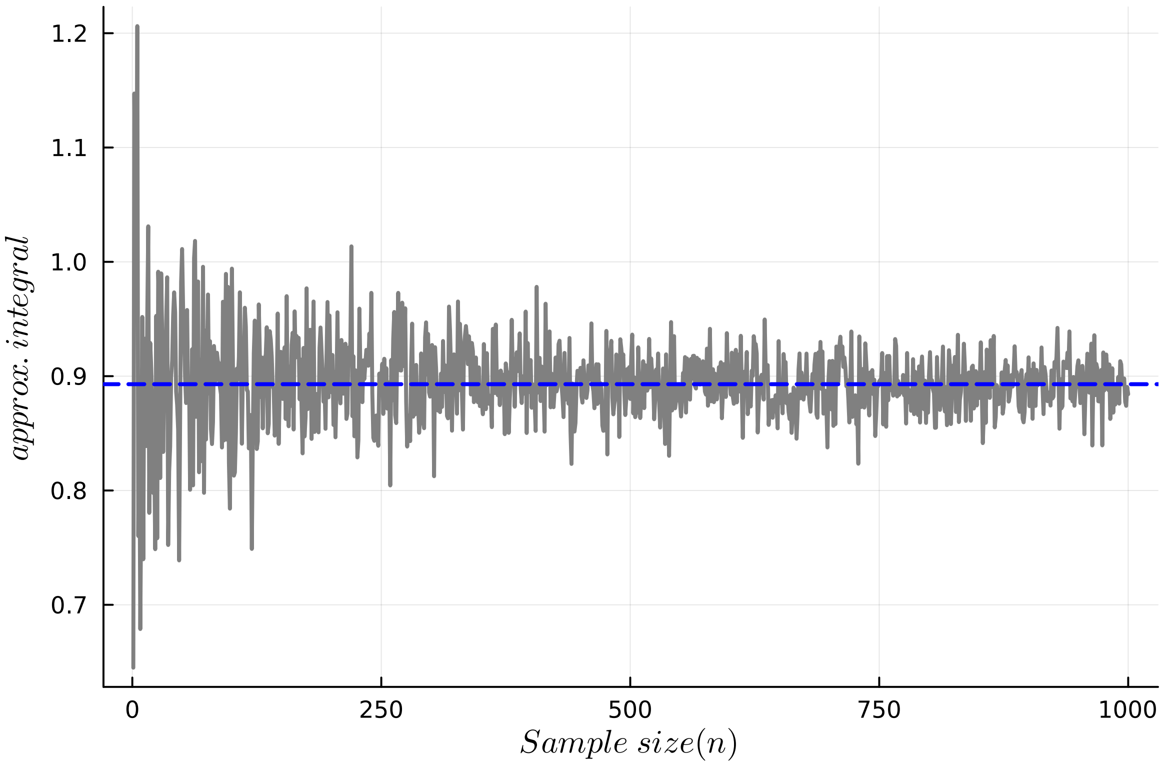

Figure 5: It is clear from the plot that as the sample size increases the Monte Carlo integral converges to the exact integral value.

Therefore, we can solve the problem of integration of the following forms

$$\int_a^\infty g(x)dx \quad \int_{-\infty}^b g(x)dx \quad \int_{-\infty}^\infty g(x)dx.$$

The supporting distribution with respect to which the expectation will be computed, needs to have the support equal to the range of integration.

4. Verification of Probability Inequalities

As a practitioner one may ask that how could you convince yourself that for any random variable with finite second moment $(\mathbb{E}(X) = \mu)$ and $(\text{Var}(X) = \sigma^2)$, the simulated values of the random variable will fall within the 3σ limit with a high probability? One of the inequality which is very well known is

$$P(|X - \mu| < 3\sigma) > \frac{8}{9}$$

that is $P(\mu - 3\sigma < X < \mu + 3\sigma) > 8/9 = 0.89$ \). It is remarkable that the inequality is true for any random variable with finite second order moment. In the following we shall verify this inequality by means of simulation. Let us evaluate the exact probability that X\) lies between $\mu - 3\sigma$ and $\mu + 3\sigma$. We do this exercise for $exponential(1)$ density function. Let us first compute $\mu = \mathbb{E}(X)$ and $\sigma^2 = \text{Var}(X)$ by integration. We are doing it here to show that the process can be done for any function.

In [25]:

```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [26]:
```
lambda = 1  # Rate parameter
fun_exponential(x) = lambda * exp(-lambda * x)
mean_fun(x) = x * fun_exponential(x)
mu, er = quadgk(mean_fun, 0, Inf)   # computing mean
println("Mean (mu): $mu")
var_fun(x) = ((x - mu)^2) * fun_exponential(x)
sigma2, er = quadgk(var_fun, 0, Inf)  # computing variance
sigma = sqrt(sigma2)
println("Variance (sigma^2): $sigma2")
println("Standard Deviation (sigma): $sigma")
```

```
Mean (mu): 0.9999999999999998
Variance (sigma^2): 1.000000000000021
Standard Deviation (sigma): 1.0000000000000104
```

In [27]:
```
plot( x -> fun_exponential(x),0,5,
    linewidth = 2,color = "red", label = "")
scatter!([mu - 3 * sigma, mu + 3 * sigma, 0], [0, 0, 0],
       color = "blue", label = "")
vline!([0, mu - 3 * sigma, mu + 3 * sigma], color = "blue",
      linestyle = :dash, label = "")
```

Out[27]:

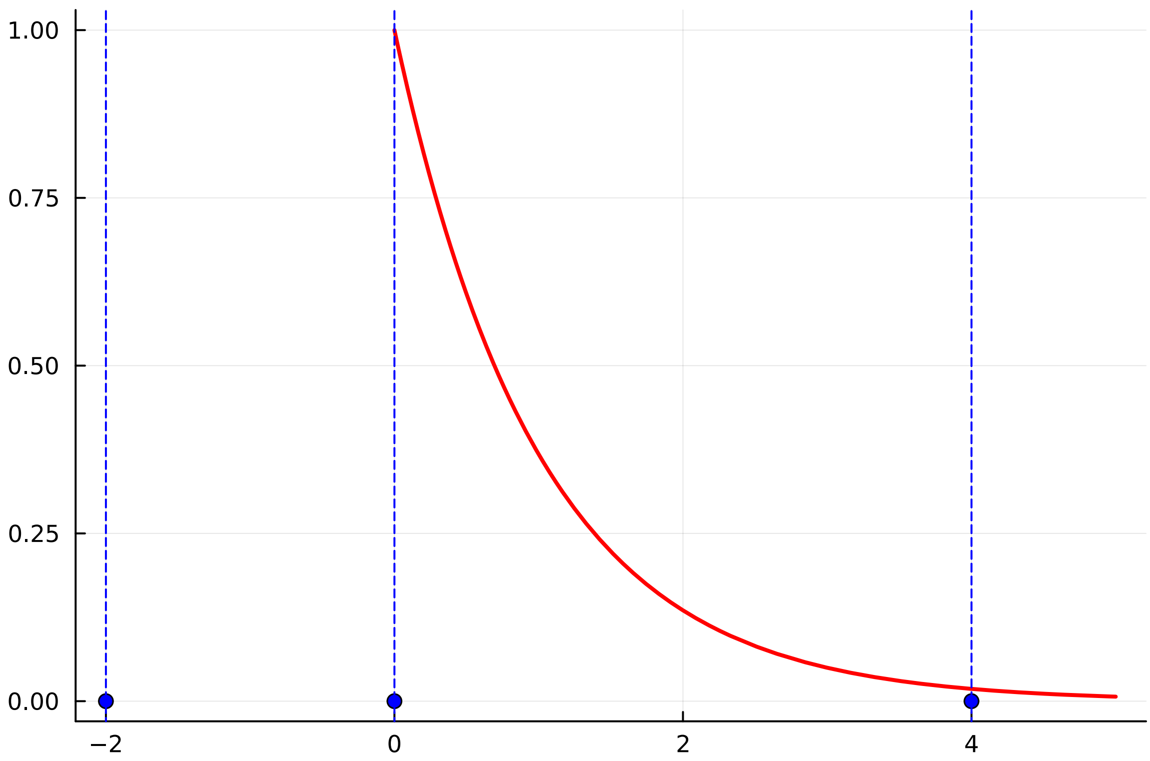

Figure 6: The 3σ limit is depicted by the blue vertical lines. By integrating the density function one can compute the exact probability and verify that the exact probability is bigger than $\frac{8}{9}$.

In [28]:
```
exact_prob, _ = quadgk(fun_exponential, 0, mu + 3 * sigma)
println("The exact probability (area under the curve) is: $exact_prob")
```

```
The exact probability (area under the curve) is: 0.9816843611112664
```

In the above code, you are encouraged to see that the computations of mean and variance have been carried out by explicitly writing functions in and then integrating them over the support of the distribution. It should be clear to the reader that the number of lines in the above chunk can be reduced significantly. This has been done for example purpose as there can be situation where there no ready-made function is available in for your desired density function. The exact probability is 0.9816844 which is bigger than $8/9 = .89$. We can look at this problem in an alternative way. If we simulate 1000 numbers from exponential(1) density function, thenat least 890 of the observations will fall within the 3σ limit. Let us simulate numbers from exponential and check how many of them are falling within the limit

In [29]:
```
n = 1000  # no of simulations
x = rand(Exponential(1), n) # simulation from exp(1)
println(first(x, 6))
```

```
[0.009722707710964019, 0.1738184735824817, 0.549855036784792, 0.1361509334667214, 0.04
0310030597028666, 1.3219535788658052]
```

In [30]:
```
sucess_exp = sum(x.< mu+3*sigma)
println("Proportion of numbers falling within the limit is: ", sucess_exp/n)
```

```
Proportion of numbers falling within the limit is: 0.984
```

5. Computing Probabilities

Suppose that we want to compute the probability $\mathbb{P}(a < X < b)$ for some random variable X. We essentially need to do an integration. For a specific example, we consider the $\mathcal{N}(0,1)$ distribution. We compute the probability $\mathbb{P}(0 < X < 1)$ by integrating the standard normal density function in $(0,1)$.

In [31]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [32]:
```
dist = Normal(0, 1)
x = 0:0.1:1
plot(-4:0.01:4, pdf.(dist, -4:0.01:4), color = :red,
    lw = 2, xlabel = "x", ylabel = "f(x)", label = "")

for i in 1:(length(x) - 1)
    fill_between = [x[i], x[i], x[i+1], x[i+1]]
    fill_pdf = [0, pdf(dist, x[i]), pdf(dist, x[i+1]), 0]
    plot!(fill_between, fill_pdf, color = :grey, label = "")
end

plot!(-4:0.01:4, pdf.(dist, -4:0.01:4), color = :red,
    lw = 2, label = "")
plot!(legend=:topright)
```

Out[32]:

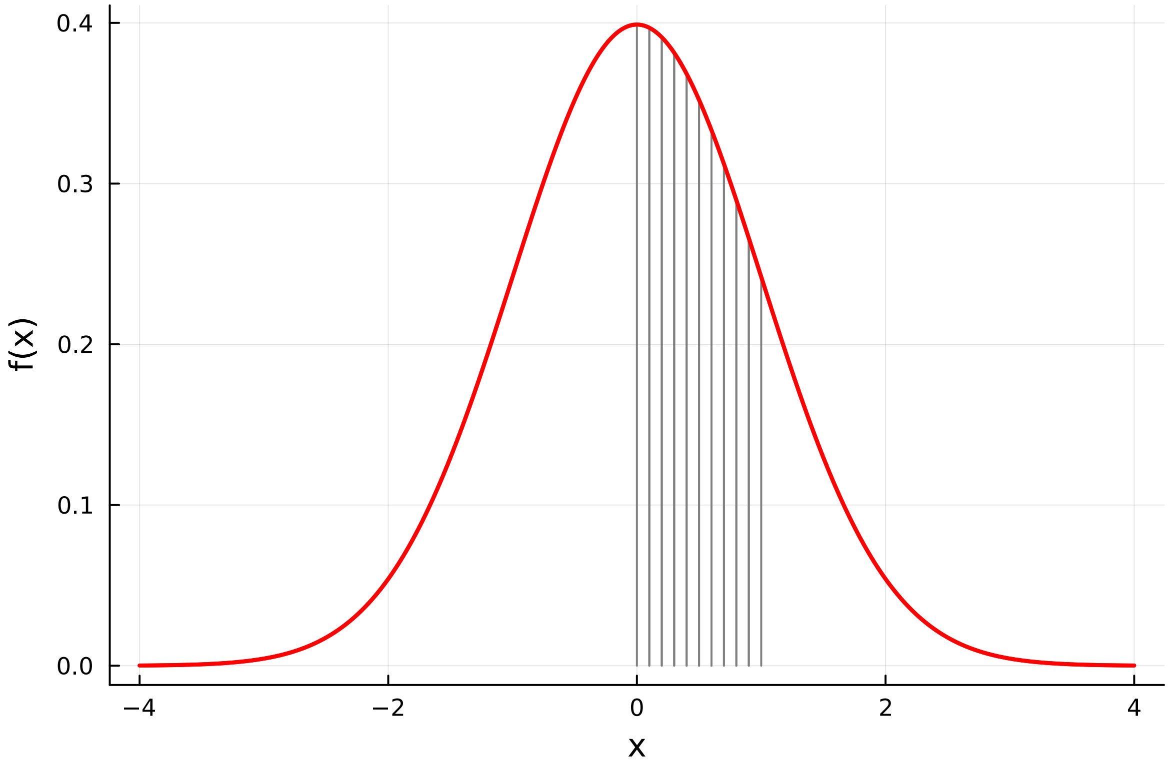

Figure 7:The probability density function (PDF) of the standard normal distribution $N(0, 1)$. The red curve represents the PDF, while the grey-shaded region highlights the area under the curve over a specific range, corresponding to the probability $P(0

In [33]:
```
dist = Normal(0, 1)
exact_prob = cdf(dist, 1) - cdf(dist, 0)
println("Exact probability P(0 < X < 1) is ", exact_prob)
```

```
Exact probability P(0 < X < 1) is 0.34134474606854304
```

We consider the following scheme for computing the probability by simulation. We will simulate 1000 random numbers from $N(0, 1)$ distribution and check how many of them falls between (0,1) and compute the proportion. The claim is that the proportion will converge to the exact probability as the number of points simulated are increased.

In [34]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [35]:
```
n = 1000
x = rand(Normal(0,1), n)
counter = 0

for j in 1:length(x)
    if x[j]<1 && x[j]>0
        counter = counter + 1
    end
end

est_prob = counter/n
println("Approximate probability based on a sample of size ",n," is ", est_prob)
```

```
Approximate probability based on a sample of size 1000 is 0.37
```

We can have nice visualization to check how the estimated probabilities are changing as the sample size changes. Let P_n be the estimate of that as the sample size increases

$P(0 < X < 1)$ based on a sample of size . In the following code, we see n converges to the exact probability.

In [36]:
```
n_vals = 1:10000
est_prob = zeros(length(n_vals))

for n in n_vals
    x = rand(Normal(0,1), n)
    counter = 0
    for j in 1:length(x)
        if x[j]<1 && x[j]>0
            counter = counter + 1
        end
    end
    est_prob[n] = counter/n
end
```

In [37]:
```
println(first(est_prob,6))
```

```
[1.0, 0.5, 0.0, 0.25, 0.8, 0.5]
```

In [38]:
```
plot(n_vals,est_prob, color = "red", xlabel = "sample size(n)",
     ylabel = L"P_n", label = "estimated")
hline!([exact_prob], color = "blue", linestyle = :dash,
      lw = 2 , label = "exact")
```

Out[38]:

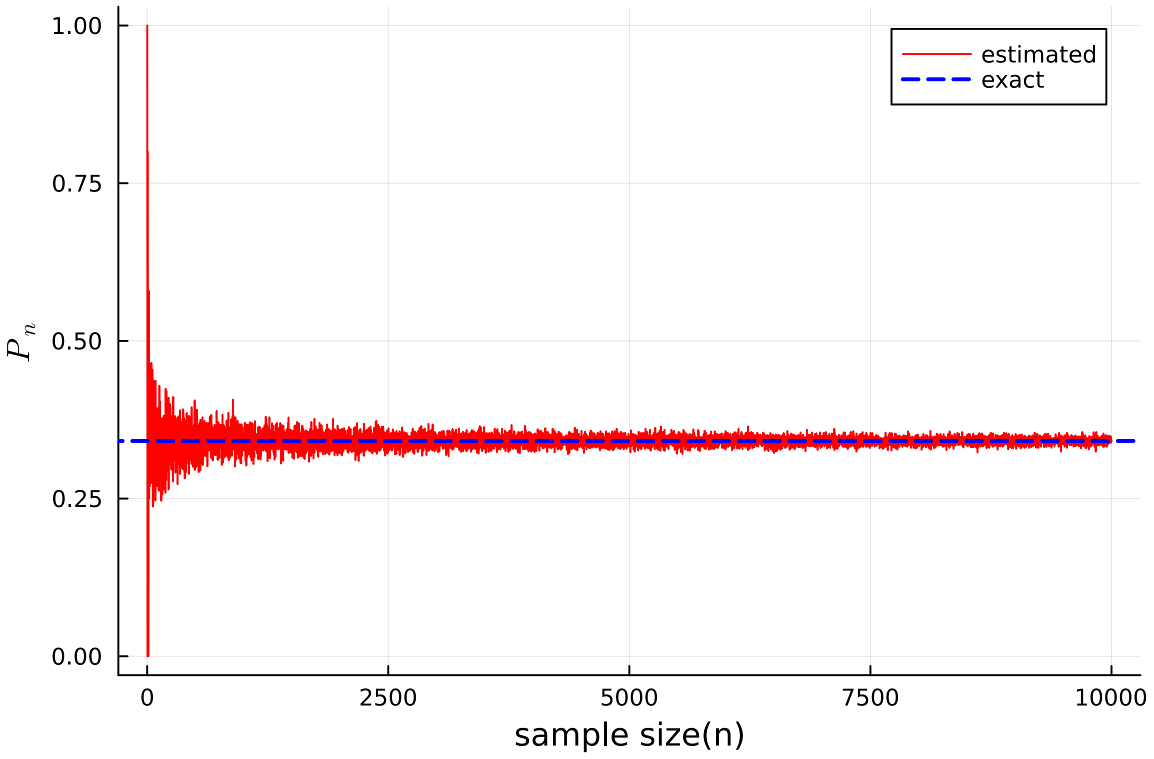

Figure 8: As the sample size of Monte Carlo simulation increases, the approximate probability is close to exact probability

It is important to note that if the reader runs the above piece of codes in her computer, different set of random numbers will be simulated. Hence, the generated figure may look slightly di erent. We can ask $Julia$ to ensure that every time the same set of random numbers being simulated by specifying a seed value as follows. The following problem deals with the approximation of $P(X<0)$ and $x \sim N(0,1)$ probability is equal to 0.5. We know that the exact.

In [39]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, Random
```

In [40]:
```
Random.seed!(1) # Reproducibility
x =  rand(Normal(0,1), 100)
println("First six values are,", first(x,6))
```

```
First six values are,[0.06193274031408013, 0.2784058141640002, -0.5958244153640522,
0.04665938957338174, 1.0857940215432762, -1.5765649225859841]
```

In [41]:
```
sum(x.<0)/length(x)
```

Out[41]:
```
0.43
```

In [42]:
```
counter = 0
for i in 1:length(x)
    if x[i]<1 && x[i]>0
        counter = counter + 1
    end
end

println("The number of values which are less than 0 is: ", counter)
```

```
The number of values which are less than 0 is: 38
```

In [43]:
```
est_prob = counter/length(x)
```

Out[43]:
```
0.38
```

Run the above piece of code multiple times, every time the estimate will be the same as `set.seed` ensures that your research output is reproducible. This approximation of the exact probability lies on the concept of the weak law of large numbers. As the sample size increases, the sample mean converges to the population mean in probability. WLLN talks about the approximation to the expectation. However, here we are talking about the approximation of the probability of some events. Try to think about how you can write $\mathbb{P}(a<X<b)$ as an expected value of some random variable Y and $\mathbb{E}(Y)=\mathbb{P}(a<X<b)$. Then if we simulate independent observations $(Y_1, Y_2, \ldots, Y_n)$ from the distribution of Y, then

$$\frac{Y_1+Y_2+\cdots+Y_n}{n}$$

will be approximately close to $\mathbb{E}(Y)$ for large n values. This idea can be easily extended for any integration within a bounded or unbounded interval. Define the random variables Y_i, $1 \leq i \leq n$ as follows:

$$Y_i = \begin{cases} 1, & \text{if } a < X_i < b \\ 0, & \text{otherwise} \end{cases}$$

Each Y_i takes values 1 with probability $\mathbb{P}(a<X<b)$ and

$$\mathbb{E}(Y_i) = 1 \times \mathbb{P}(Y_i=1) + 0 \times \mathbb{P}(Y_i=0) = \mathbb{P}(a<X<b).$$

Basically, each Y_i is a Bernoulli(p) random variable with success probability $p = \mathbb{P}(a < X < b)$. Now by WLLN, $\overline{Y_n} \to p$ in probability. In the following code, we demonstrate this convergence for large n values.

In [44]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [45]:
```
using Distributions
using QuadGK

a, b = 0, 1
normal_dist = Normal(0, 1)
fun_normal(x) = pdf(normal_dist, x)
exact_prob, er = quadgk(fun_normal, a, b)

println("The estimated probability P(a <= X <= b) is: ", exact_prob)
```

```
The estimated probability P(a <= X <= b) is: 0.341344746068543
```

In [46]:
```
n_vals = 1:1000
prob_n = zeros(length(n_vals))

for n in n_vals
    x = rand(Normal(0,1), n)
    prob_n[n] = mean((0 .< x) .& (x .< 1))
end
```

In [47]:
```
plot(n_vals, prob_n, color = "red", lw = 2, xlabel = "Sample Size (n)",
    ylabel = L"\bar{Y_n}", label = "")
hline!([exact_prob], color = "blue", lw = 2, linestyle = :dash, label = "Exact prob'
```

Out[47]:

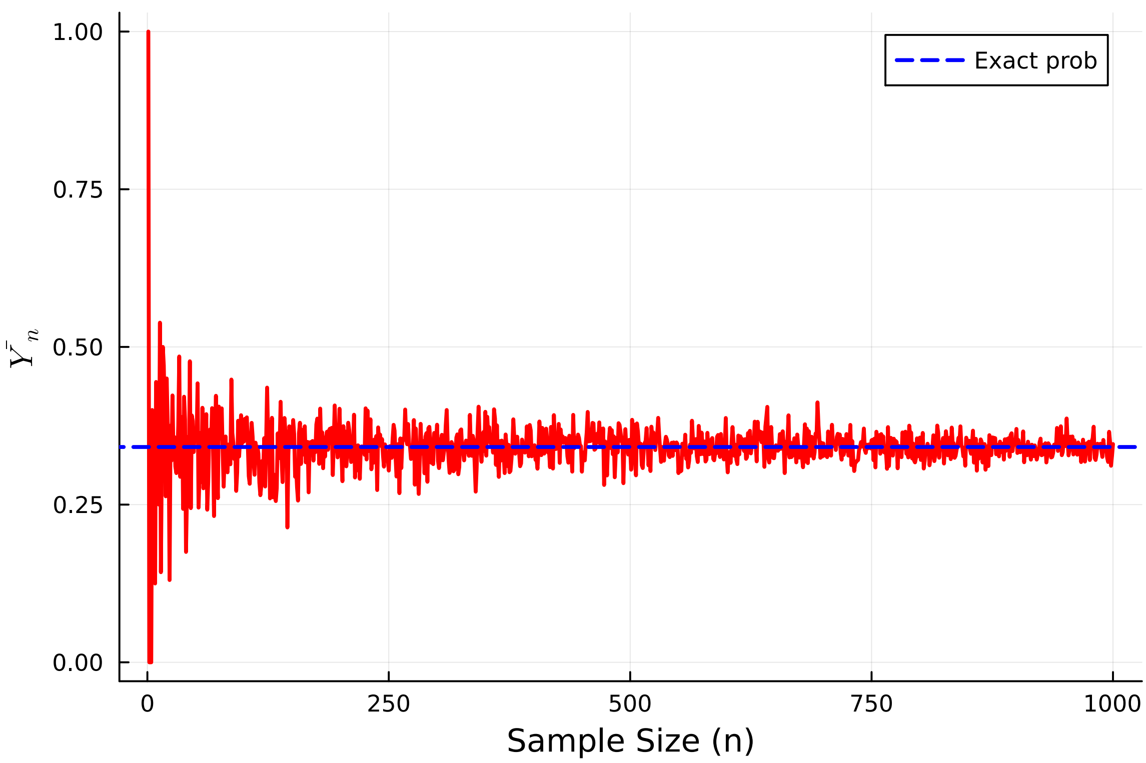

Figure 9: As the number of simulation increases, the sample proportions of success converges to the true proportion. This is a demonstration of WLLN when we are interested in approximating probabilities of some event

6. Variance of Monte Carlo Approximation

Suppose, we consider the $exponential(1)$ density function which has the required support. So the integral can be written as

$$\int_0^\infty e^{-x^3} dx = \int_0^\infty e^{-x^3} e^{-x} dx$$

$$= \int_0^\infty e^{x-x^3} e^{-x} dx$$

$$= E_{X\sim\text{exponential}(1)} \left(e^{X-X^3}\right)$$

$$= E_{X\sim\text{exponential}(1)} \left(\psi(X)\right)$$

Basically, we need to approximate the expectation $E(\psi(X))$, where $X \sim \text{exponential}(1)$.

- Simulate $X_1, X_2, \ldots, X_M \sim \text{exponential}(1)$.
- Compute $\psi(X_1), \ldots, \psi(X_M)$.
- $\hat{I}_M = \frac{1}{M} \sum_{j=1}^{M} \psi(X_j)$.

As $M \to \infty$, $\hat{I}_M \to I$ in probability. As we understand that $\hat{I}_M$ is a random variable for every M, the standard error of $\hat{I}_M$ will give us the error associated with the approximation. The standard error of the estimate is given by

$$\hat{s}e\left(\hat{I}_M\right) = \frac{s}{\sqrt{M}}$$

where

$$s^2 = \frac{\sum_{j=1}^{M} \left(Y_j - \hat{I}_M\right)^2}{M-1}$$

where $Y_j = \psi(X_j)$. A $(1-\alpha)$ confidence interval for I is

$$\hat{I}_M \pm z_{\alpha/2} \hat{s}e.$$

We have utilized the fact that for large M, $\hat{I}_M$ has an approximate normal distribution.

In [48]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [49]:
```
g(x) = exp.(-x.^3)
exact_integral, er = quadgk(g,0,Inf)
println(exact_integral)
```

```
0.8929795115692495
```

In [50]:
```
psi(x) = exp.(x.-x.^3)
M = 1000
x = rand(Exponential(1), M)
approx_integral = mean(psi(x))
println(approx_integral)
```

```
0.8768311765964493
```

In [51]:
```julia
se = sqrt(var(psi(x))/M)
println("Standard error of the approximation is : ", se)
```

```
Standard error of the approximation is : 0.017789243017344143
```

In [52]:
```julia
println("95% CI of the integral is: ", approx_integral - 1.96*se, " ",
    approx_integral + 1.96*se)
```

```
95% CI of the integral is: 0.8419642602824547 0.9116980929104438
```

We compare the Monte Carlo approximation of an integral using two di erent supporting density function exp(1) and exp(10) density function. Let us simulate random numbers from the sampling distribution of $\widehat{I_M}$ m=1000 many times.

In [53]:
```julia
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [54]:
```julia
m = 1000 # no of replications
M = 10000 # monte carlo sample size
approx_integral = zeros(m)
for i in 1:m
    x = rand(Exponential(1),M)
    approx_integral[i] = mean(psi(x))
end
```

In [55]:
```julia
println(first(approx_integral, 5))
```

```
[0.8914633500517565, 0.893945266454704, 0.8889845432970686, 0.8922001808693331, 0.885
100534978075]
```

In [56]:
```julia
histogram(approx_integral, normalize = true, label = "",
          xlabel = L"\widehat{I_M}", title = "M = $M")
scatter!([exact_integral], [0], color = "red", marker = :circle,
       markersize = 8, label = "")
```

Out[56]:

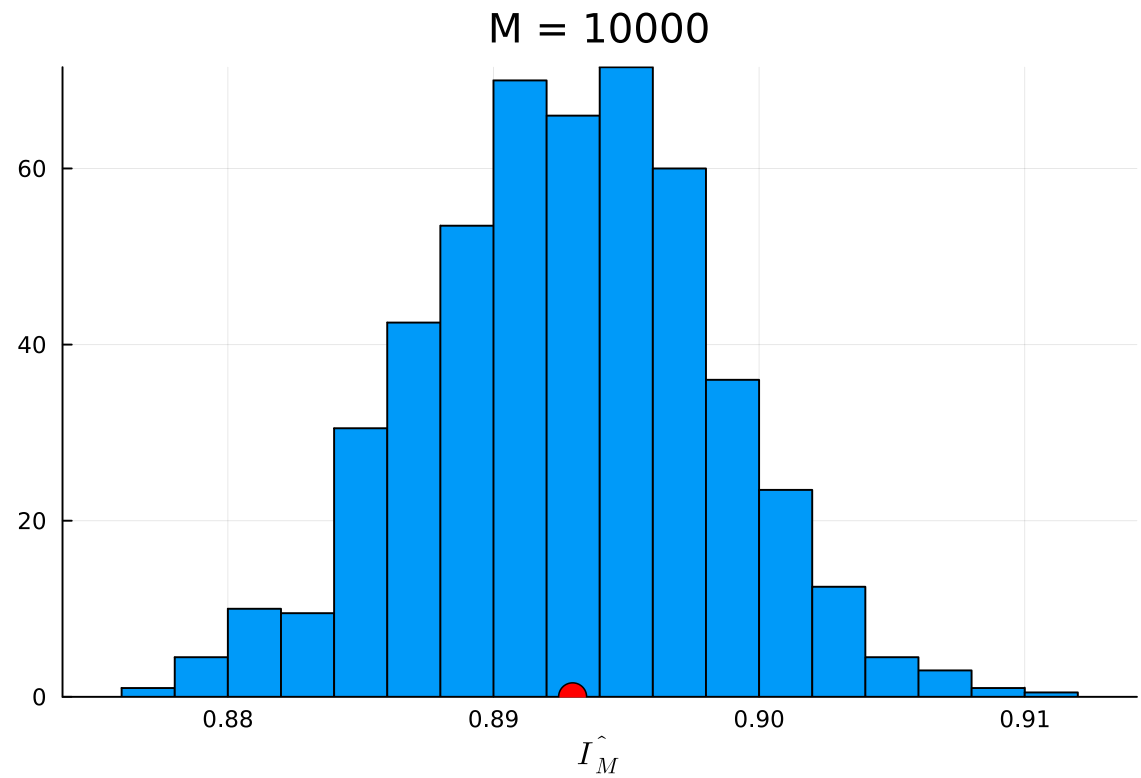

Figure 10: Approximate sampling distribution of the Monte Carlo estimator for the integral. It is clear that as M the histogram will be shrinked to the true value shown using red dot

As M increases, the approximation is going to be more and more accurate

In [57]:
```
M_vals = 1:10000
approx_integral = zeros(length(M_vals))

for M in M_vals
    x = rand(Exponential(1),M)
    approx_integral[M] = mean(psi(x))
end
```

In [58]:
```
plot(M_vals, approx_integral, color = "red", xlabel = "M",
     ylabel = L"\widehat{I_m}", label = L"\widehat{I_M}")
hline!([exact_integral], color = "blue", lw = 2, linestyle = :dash,
       label = L"I_M")
```

Out[58]:

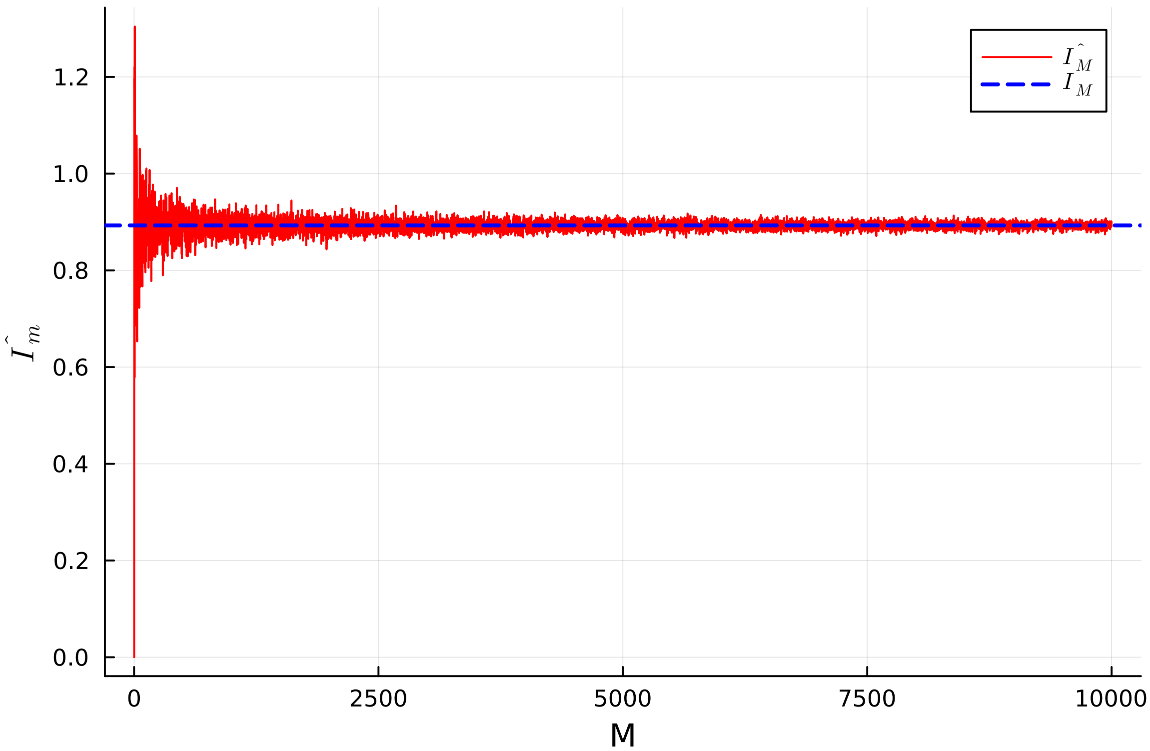

Figure 11: Shape of the sampling distribution of the Monte Carlo approximator of the integral. This gives us a visuals about the possible deviations of the Monte Carlo estimate of the integral. As M increases, the Monte Carlo integral conerges to the true integral. The idea is to show that the variance goes to zero. As the number of sample increases, the accuracy also increases.

Note that we have used **exponential(1)** density function to compute the integral. However, there are many possibilities of the density functions that can be used to compute this integral.

$$\int_0^\infty e^{-x^3} dx = \int_0^\infty e^{-x^3} \frac{1}{10} e^{10x} \times \left(10e^{-10x}\right) dx$$

$$
=
$$

$$
\int_0^\infty \frac{1}{10} e^{10x - x^3} f(x) dx
$$

$$
=
$$

$$
\int_0^\infty \psi(x) f(x) dx
$$

$$
= E_{X \sim \text{exponential}(10)} \left(\psi(X) \right)
$$

In [59]:
```
M_vals = 1:10000
psi_10(x) = (1/10) .* exp.(10 .* x .- x.^3)
approx_integral = zeros(length(M_vals))

for M in M_vals
    x = rand(Exponential(1/10),M)
    approx_integral[M] = mean(psi_10(x))
end
```

In [60]:
```
plot(M_vals,approx_integral, color = "red", lw = 2,
    ylabel = L"\widehat{I_M}", xlabel = "M",
    title = L"X~~exponential(10)", label = L"\widehat{I_M}")
hline!([exact_integral], color = "blue", lw = 2,
      linestyle = :dash, label = L"I_M")
```

Out[60]:

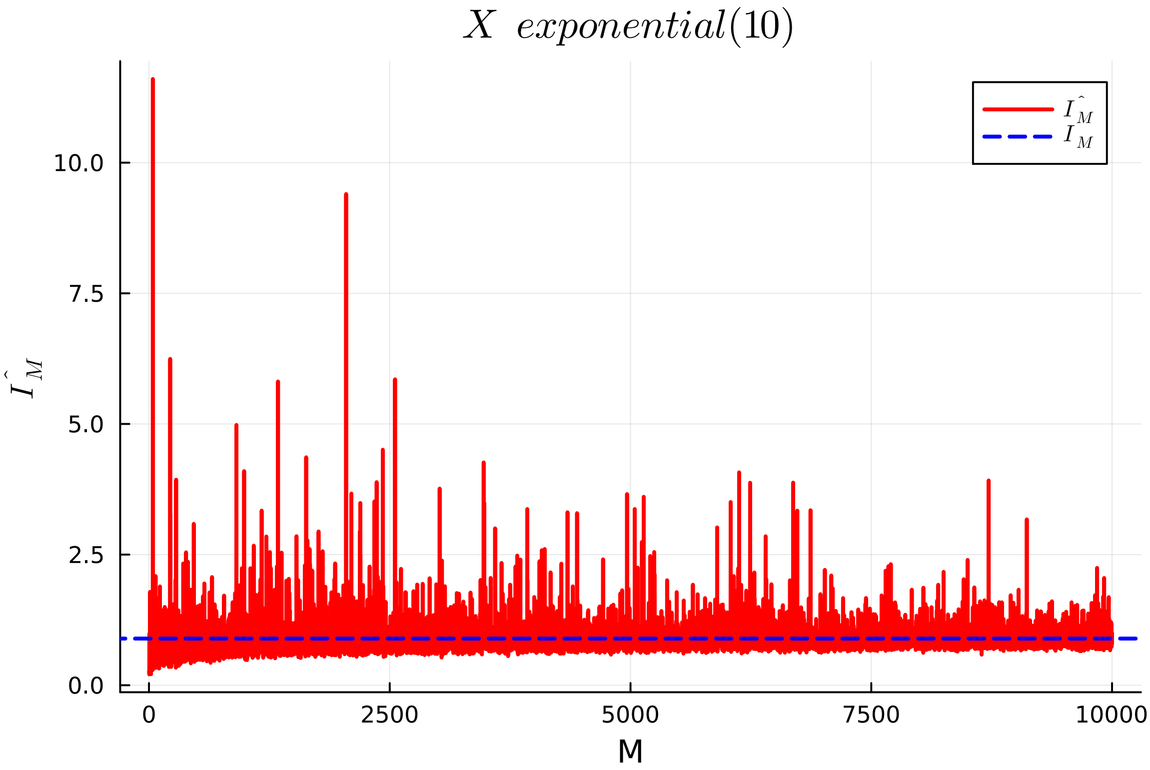

Figure 12: It is important to consider with respect to which density function we are writing the integral as an expected value. The same exercise, we have performed by writing the expected with respect to exponential density with large λ (mean) value. From the plot, it is clear that the convergence of the approximation is not satisfactory even for large M values

Note that the variance associated with $\widehat{I_M}$ when sampling from exponential(1) and exponential(10) are very different. One needs to choose the supporting density function in such a way that your approximation has the variance as small as possible.

For fixed M, we can simulate the sampling distribution of $\widehat{I_M}$ using $X \sim exponential(1)$ and $X \sim exponential(10)$

In [61]:
```
psi(x) = exp.(x.-x.^3)
psi_10(x) = (1/10) .* exp.(10 .* x .- x.^3)

M = 10000
m = 1000
I_M = zeros(m)
I_M_10 = zeros(m)

for i in 1:m
    x = rand(Exponential(1), M)
    I_M[i] = mean(psi(x))
    x = rand(Exponential(1/10), M)
    I_M_10[i] = mean(psi_10(x))
end

println(first(I_M,6))
```

```
[0.8924561941375332, 0.8972683082569686, 0.8939887837434615, 0.9015141964744007, 0.88
60465709190601, 0.8954578033416376]
```

In [62]:
```
p1 = histogram(I_M_10, normalize=true, label=false,
     xlabel=L"\widehat{I_M}",  title=L"{exponential}(10)")

scatter!([exact_integral], [0], markershape=:circle,
       color="red", markersize=6, label="I")

p2 = histogram(I_M, normalize=true, label=false, xlabel=L"\widehat{I_M}",
     color="magenta", title=L"{exponential}(1)")

scatter!([exact_integral], [0], markershape=:circle, color="red",
         markersize=6, label="I")

plot(p1,p2, layout = (1,2))
```

Out[62]:

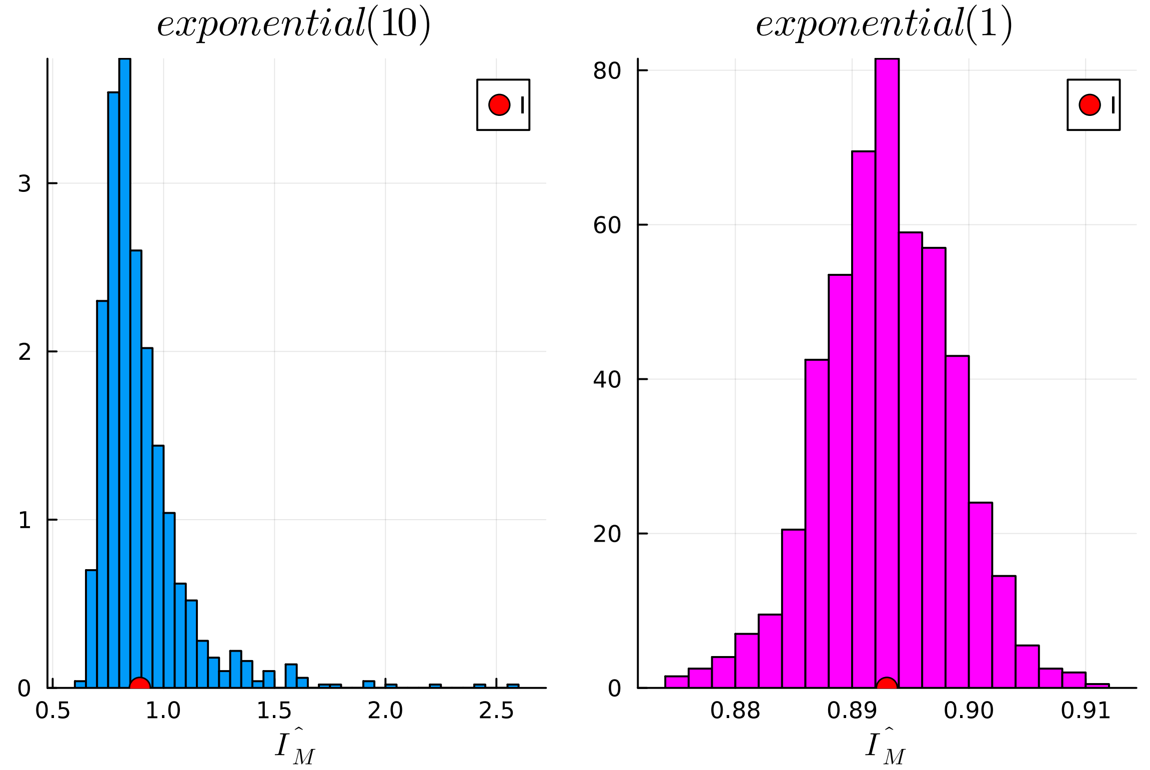

Figure 13: The integral has been evaluated based $M = 1000$ monte carlo simulation. It is important to note that the Monte carlo approximate using $\exp(1)$ as the support density gives more closeness to the true value as compared to the $\exp(10)$ density function. We shall see later what could be a good choice of the supporting density function

7. Some more examples

Suppose we are interested in computing the integral

$$\int_0^\infty \frac{1}{1+x^2+x^4}\,dx.$$

We write down the integration as an expectation where the averaging is considered with respect to the exponential density with rate parameter λ.

$$\int_0^\infty \frac{1}{1+x^2+x^4}\,dx = \int_0^\infty \frac{1}{1+x^2+x^4} e^{\lambda x} e^{-\lambda x}\,dx = \mathbb{E}_{X\sim\mathrm{Exp}(\lambda)}\left(\frac{e^{\lambda X}}{\lambda(1+X^2+X^3)}\right).$$

We call

$$\psi(X) = \frac{e^{\lambda X}}{\lambda(1+X^2+X^3)}$$

and the integral is approximated as

$$\frac{1}{M}\sum_{m=1}^{M}\psi(X_i),$$

where

$$X_1,\ldots,X_M \sim \mathrm{Exp}(\lambda)$$

and they are independent. In the following, we perform experiments with different choices of λ and understand how the average of $\psi(X_i)$ varies.

In [63]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [64]:
```
g(x) = (1 + x.^2 + x.^3).^(-1) .* (x .> 0) # function
plot(g, -1, 6, color="red", lw=2, label="")
```

Out[64]:

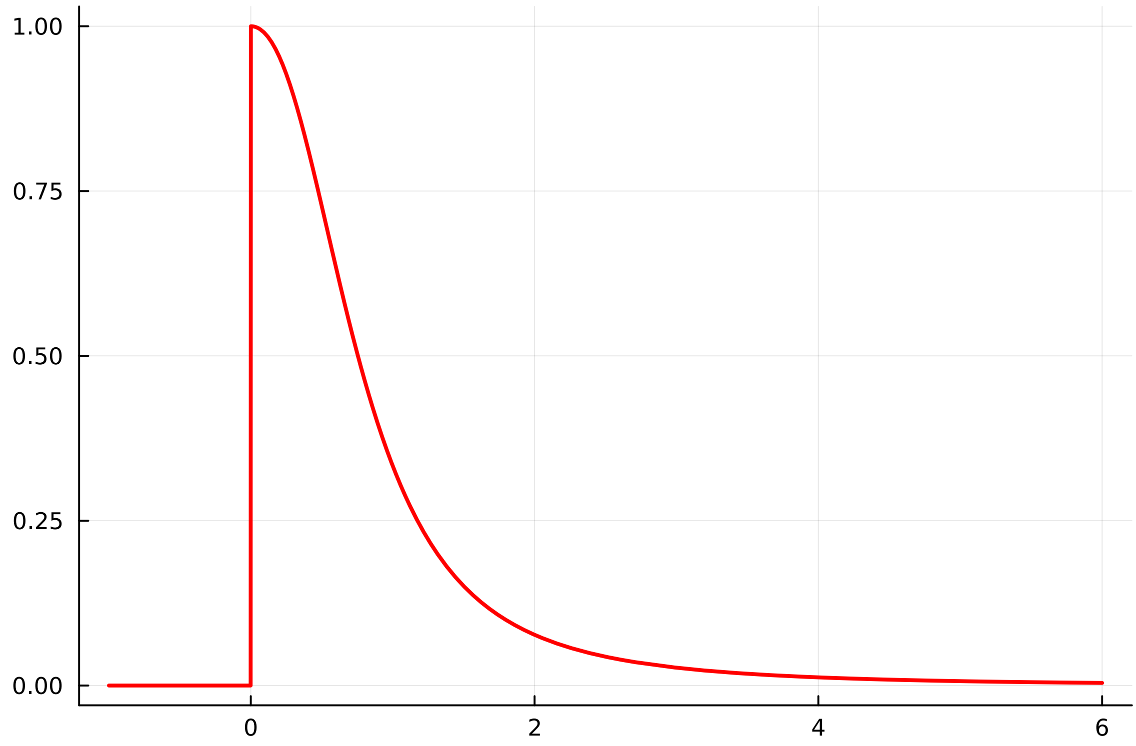

Figure 14 : The plot represents the function $g(x) = \frac{1}{1+x^2+x^3}$ over the range $[-1, 6]$. The red curve illustrates the behavior of $g(x)$, which is used in the integral approximation via Monte Carlo methods with an exponential sampling distribution.

In [65]:
```
I , er =  quadgk(g, 0, Inf) # er = numerical error
println("exact integral: ", I)
```

```
exact integral: 0.9693439633068099
```

In [66]:
```
lambda = 0.05
psi(x) = exp.(lambda .* x) .* (g(x) / lambda)
```

Out[66]:
```
psi (generic function with 1 method)
```

In [67]:
```
n_vals = [2, 5, 10, 25, 50, 100, 500, 1000, 2000, 5000, 10000, 25000]
rep = 1000

fig = plot(layout=(3, 4), size=(900, 600))

for (idx, n) in enumerate(n_vals)
    I_n = zeros(rep)
    for i in 1:rep
        x = rand(Exponential(1 / lambda), n)
        I_n[i] = mean(psi.(x))
    end
    histogram!(I_n, normalize=true, bins=30, color=:lightgray,
               title="n = $n", xlabel=L"I_n", legend=false,
               subplot=idx)
    scatter!([mean(I_n)], [0], color=:red, marker=:circle,
             subplot=idx)
end

display(fig)
```

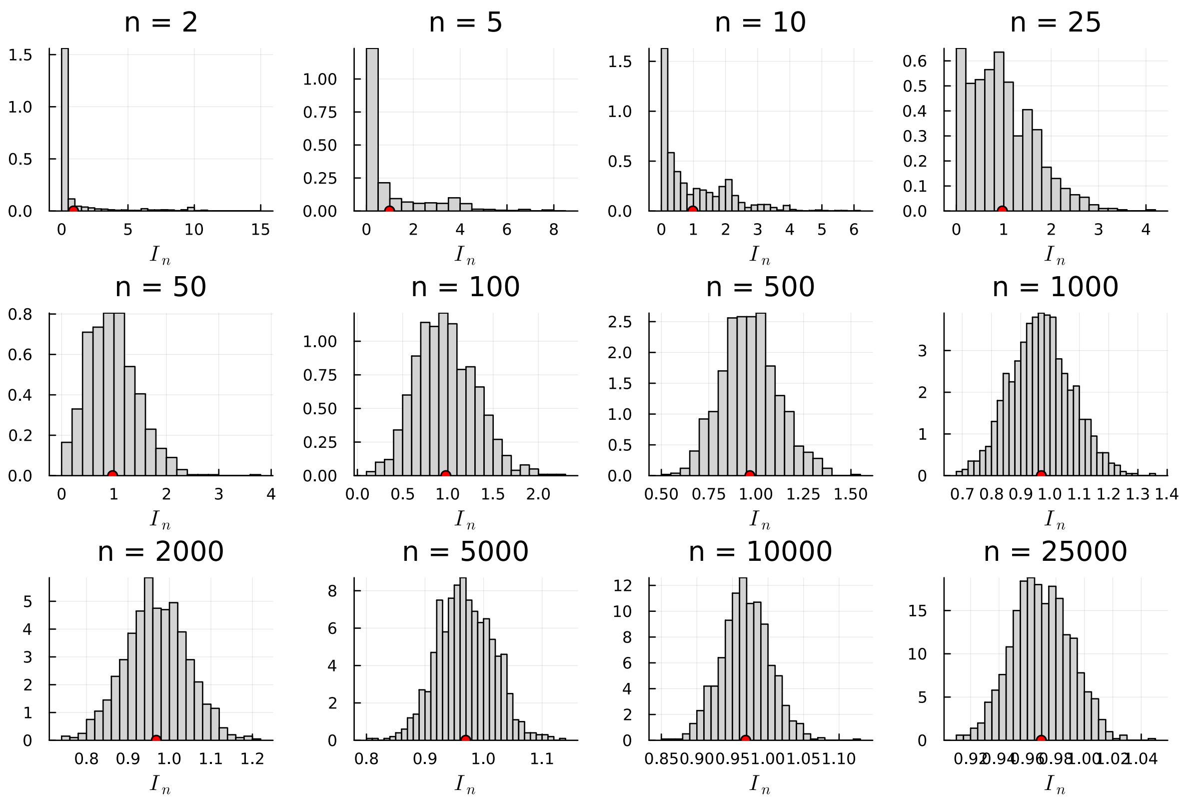

Figure 15: Histogram of the Monte Carlo estimate I_n for various sample sizes n. As $n \to \infty$, the distribution of I_n converges to a normal distribution centered at the expected value $\mathbb{E}[I_n]$, demonstrating the law of large numbers and the central limit theorem.

7. References

- Casella, G., & Berger, R. L. (2002). Statistical inference. 2nd ed. Australia ; Pacific Grove, CA, Thomson Learning.
- Robert, C. P., & Casella, G. (2004). Monte Carlo statistical methods (2nd ed.). Springer.