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

Introduction

Statistical inference typically involves two main types of numerical challenges: optimization problems and integration problems. As highlighted in works such as Rubinstein (1981), Gentle (2002), and Robert (2001), it is often not possible to compute estimators analytically within various inferential frameworks like maximum likelihood, Bayesian methods, or the method of moments. Consequently, numerical solutions are commonly employed across different types of inference.

The previous chapter introduced several methods for generating random variables from arbitrary distributions using computational techniques. These methods serve as a foundation for solving statistical problems through simulation, either from the true distribution or an appropriate approximation. In the context of decision theory, whether classical or Bayesian, this simulation-based approach is a natural choice because risks and Bayes estimators often require the evaluation of integrals over probability distributions.

Additionally, the ability to generate an unlimited number of random variables from a specified distribution facilitates the application of frequentist and asymptotic methods, which is not always feasible in traditional inferential settings where sample sizes are fixed. This flexibility allows for the use of important probabilistic results such as the Law of Large Numbers and the Central Limit Theorem, which help evaluate the convergence behavior of simulation methods. This role is similar to the use of deterministic bounds in standard numerical techniques.

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),

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

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

In [2]:
```
g(x) = exp.(-x.^2) # define the function
exact_integral, er = quadgk(g, 0,1)
println("The exact integral is:", exact_integral)
```

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

In [3]:
```
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: 0.7558857430754845
```

In [4]:
```
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

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[4]:

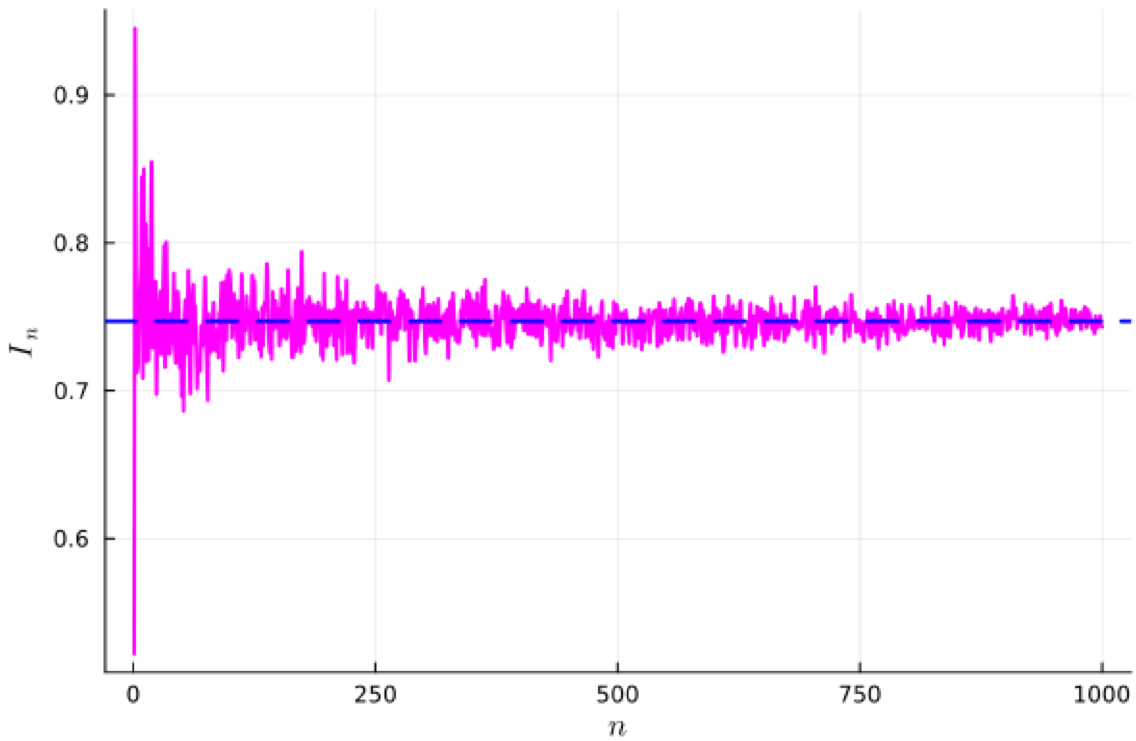

Figure 1: As $n \to \infty$, the Monte Carlo integral converges to the true integral.

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 [5]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase
```

In [6]:
```
rep = 1000
n = 10
I_n = zeros(rep)
g(x) = exp.(-x.^2) # define the function

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

histogram(I_n, normalize = true, bins = 30,title= "n = $n",
          xlabel = L"\widehat{I_n}",ylabel = "Density",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[6]:

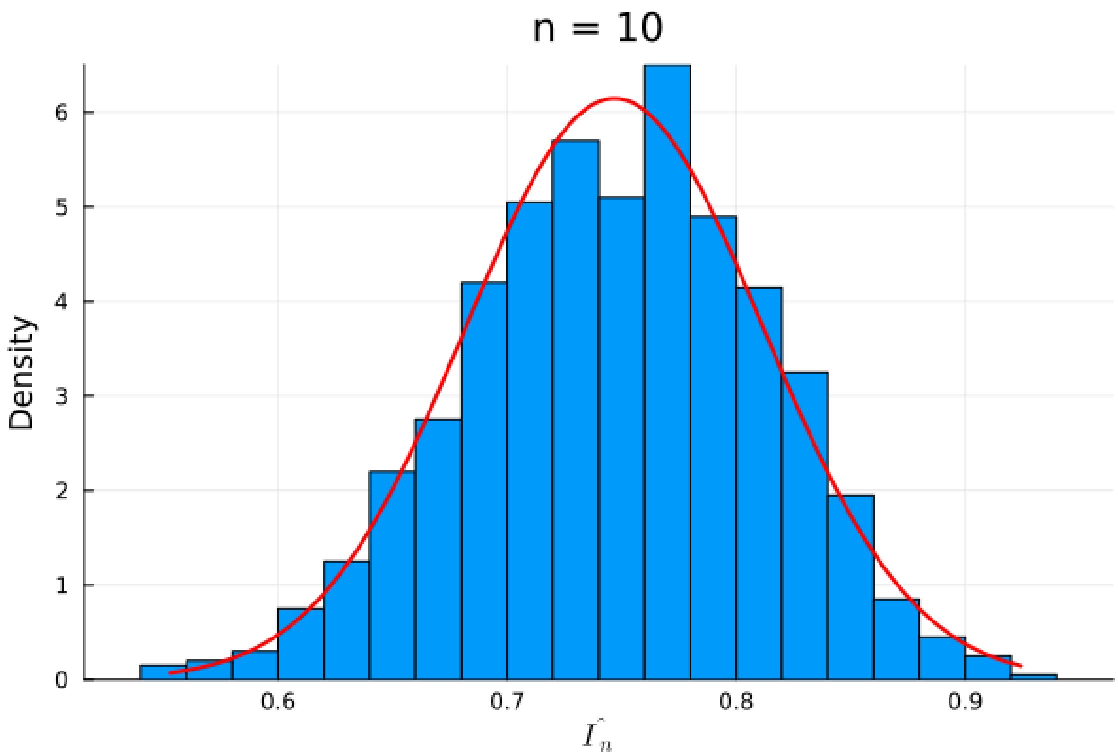

Figure 2: The sampling distribution of $\widehat{I_n}$ is well approximated by the normal distribution for large n

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

In [8]:
```
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[8]:

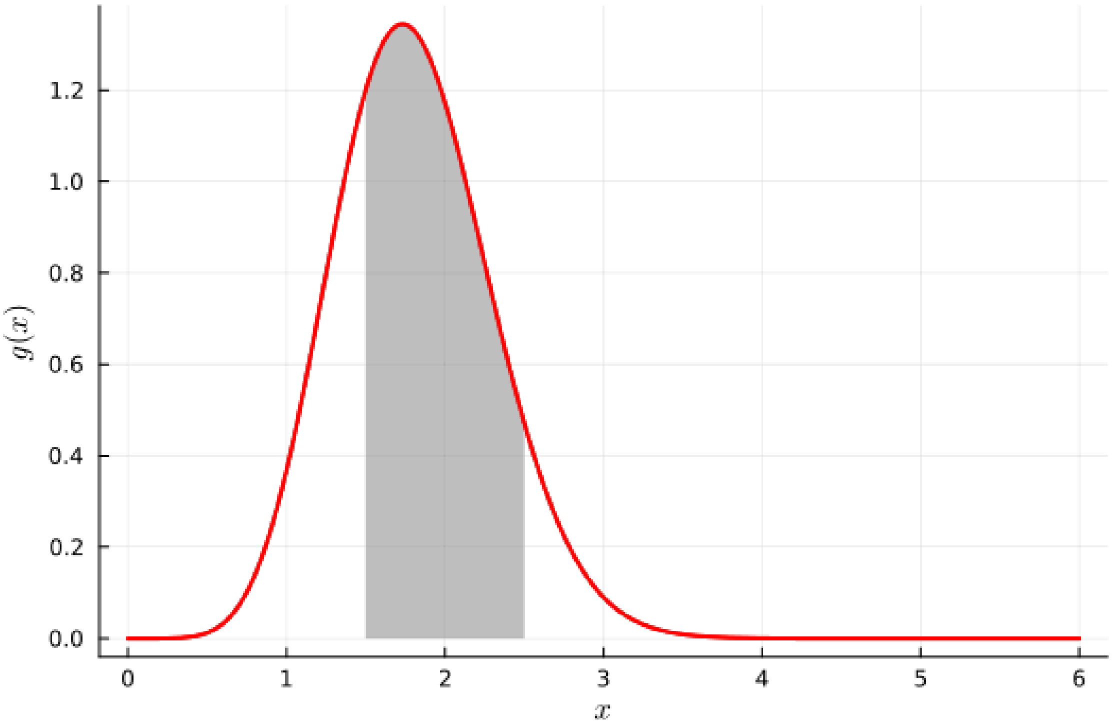

Figure 3: Required integral is the area of the shaded region which can be obtained by integrating the function $g(x)$ in the interval (a, b)

In [9]:
```
# 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 [10]:
```
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:1.0471421374899665
```

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

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

In [12]:
```
a = 2  # lower limit
b = 4  # upper limit
g(x) = x .^ 6 .* exp.(-x .^ 2)
```

Out[12]:
```
g (generic function with 1 method)
```

In [13]:
```
exact_integral, er = quadgk(g, a, b)
```

Out[13]: (0.5525956440284987, 1.5878329900909094e-11)

In [14]:
```
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

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), size = (800, 500))
```

Out[14]:

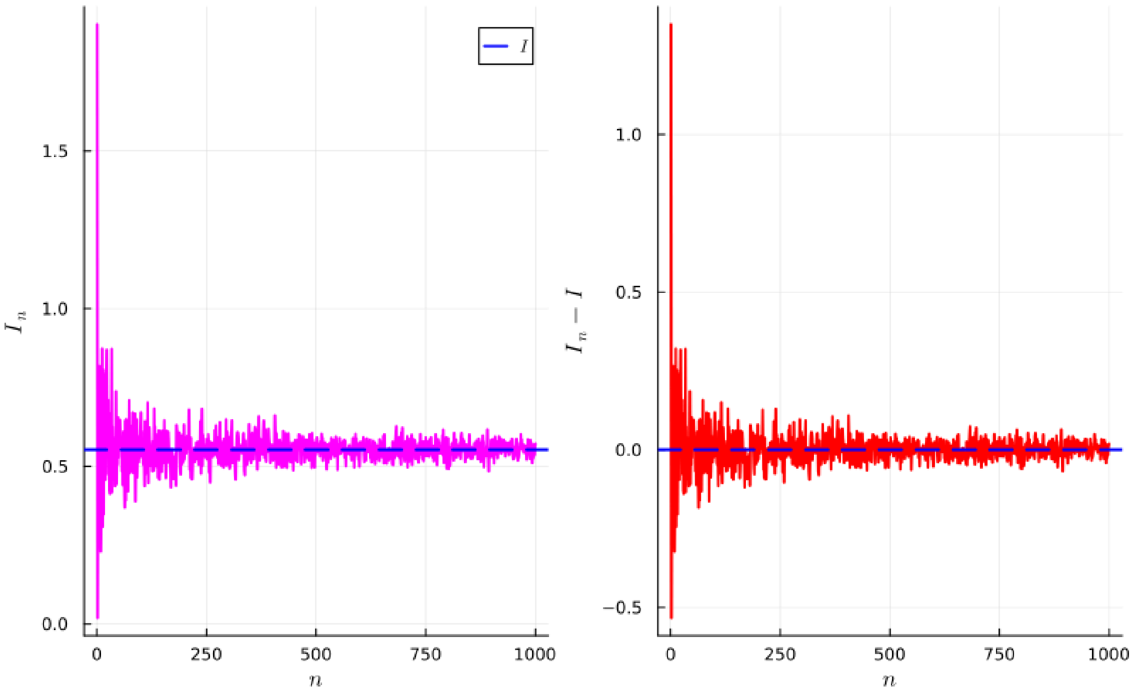

Figure 4: As $n \to \infty$ the approximate integral converges to the true integral. The right panel depicts the difference between the approximate integral and exact integral. The errors are centered around 0 and as the number of simulation increases, the errors are getting highly concentrated at 0.

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

In [16]:
```
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: 0.9001580150908257
```

In [17]:
```
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

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

Out[17]:

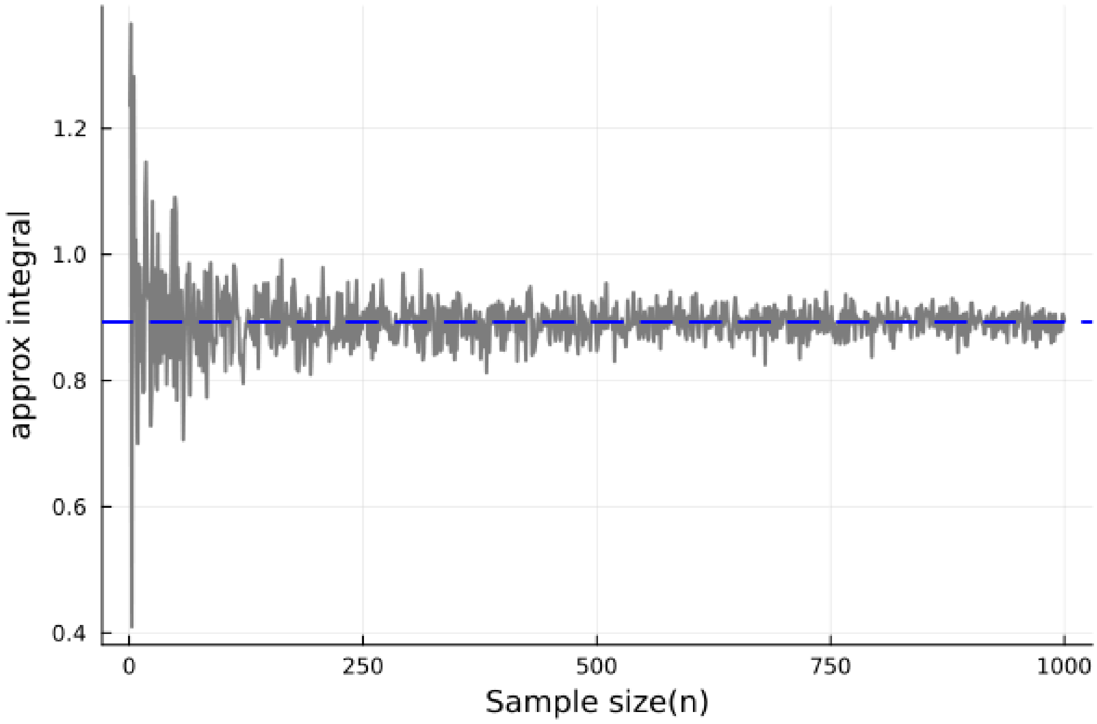

Figure 5: It is clear from the plot that as the sample size increases the Monte Carlo integral converges to the exact integral value.

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.

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 $(\mathrm{Var}(X) = \sigma^2)$, 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 $\sigma^2 = \mathrm{Var}(X)$ by integration. We are doing it here to show that the process can be done for any function.

In [18]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [19]:
```
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 [20]:
```
plot( x -> pdf(Exponential(1), x) ,-5 ,5,
linewidth = 4 ,color = "red", xlabel = "x", ylabel = "pdf(Exponential(1), x)",
label = "")
scatter!([mu - 3 * sigma, mu + 3 * sigma, 0], [0, 0, 0],
       color = "blue", markersize = 8 ,label = "")
vline!([0, mu - 3 * sigma, mu + 3 * sigma], color = "blue", lw = 3,
      linestyle = :dash, label = "")
```

Out[20]:

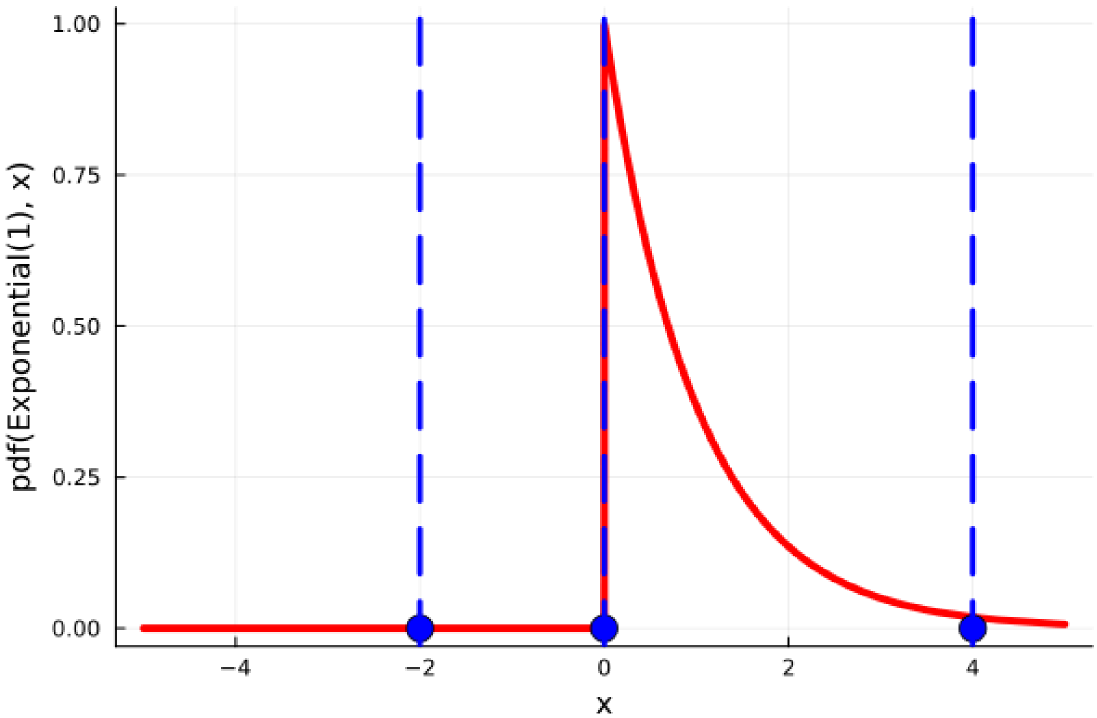

Figure 6: The 3σ limit is depicted by the blue vertical lines. By integrating the density function one can compute the exact probability and verify that the exact probability is bigger than $\frac{8}{9}$.

In [21]:
```
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 [22]:
```
n = 1000  # no of simulations
x = rand(Exponential(1), n) # simulation from exp(1)
println(first(x, 6))
```

```
[0.10393730487360173, 0.6431509536623328, 2.7228824110651613, 0.010749098273635432, 0.011793648858602217, 1.283005023604487]
```

In [23]:
```
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: 0.98
```

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 [24]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [25]:
```
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[25]:

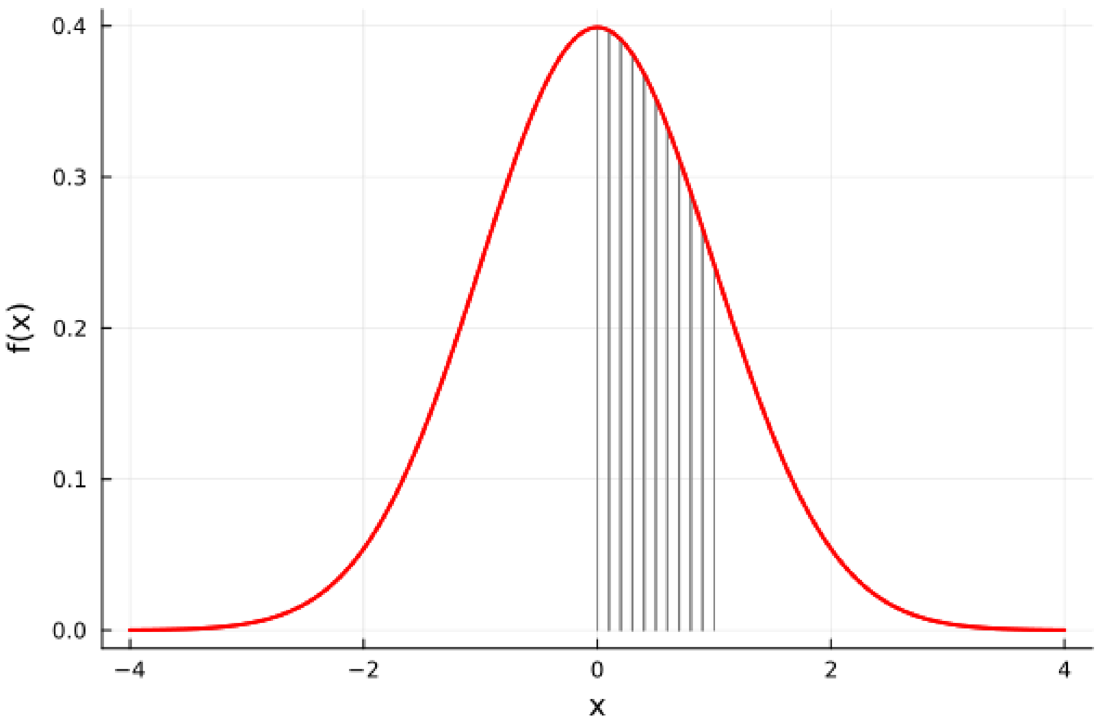

Figure 7:The probability density function (PDF) of the standard normal distribution $N(0, 1)$. The red curve represents the PDF, while the grey-shaded region highlights the area under the curve over a specific range, corresponding to the probability $P(0 < X < 1)$

In [26]:
```
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 [27]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [28]:
```
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 0.339
```

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 [29]:
```
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 [30]:
```
println(first(est_prob,6))
```

```
[0.0, 0.5, 0.3333333333333333, 0.0, 0.2, 0.5]
```

In [31]:
```
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[31]:

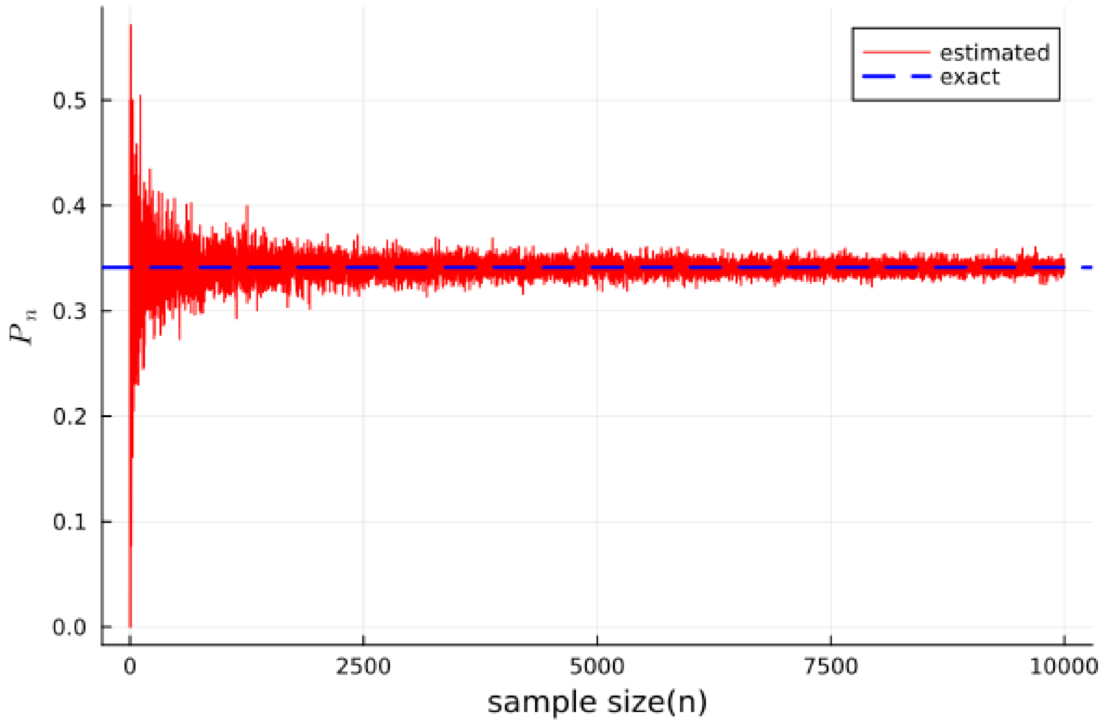

Figure 8: As the sample size of Monte Carlo simulation increases, the approximate probability is close to exact probability

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 [32]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, Random
```

In [33]:
```
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, -0.595824415364052
2, 0.04665938957338174, 1.0857940215432762, -1.5765649225859841]
```

In [34]:
```
sum(x.<0)/length(x)
```

Out[34]:
```
0.43
```

In [35]:
```
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 [36]:
```
est_prob = counter/length(x)
```

Out[36]:
```
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 [37]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [38]:
```
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 [39]:
```
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

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 p
```

Out[39]:

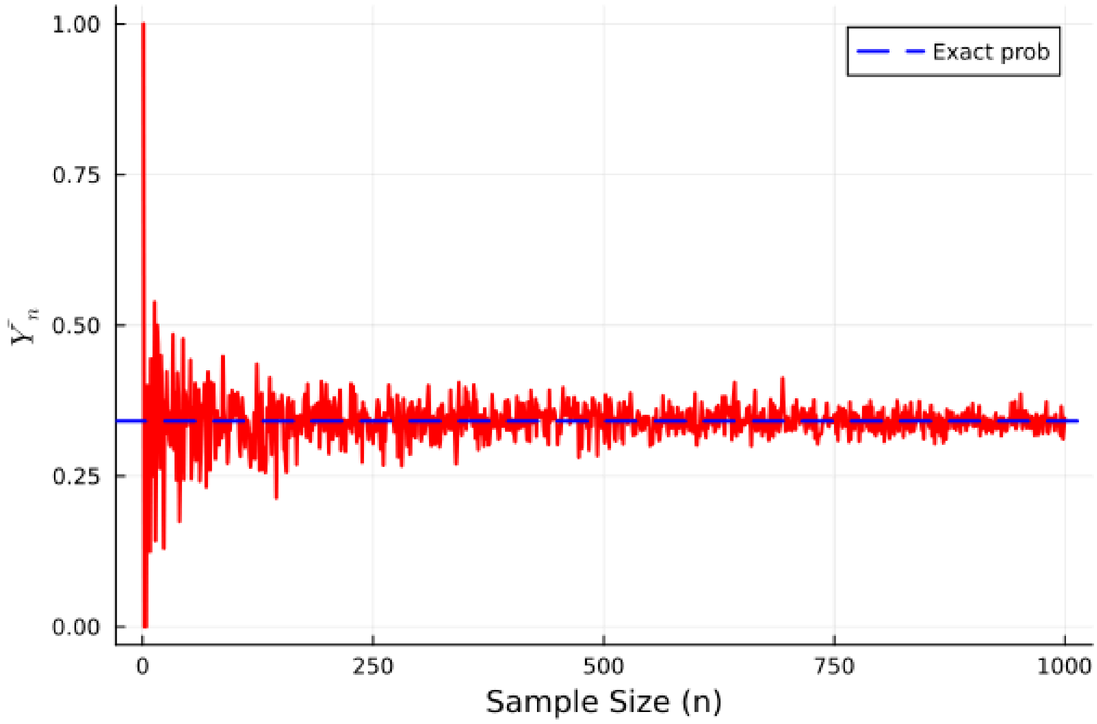

Figure 9: As the number of simulation increases, the sample proportions of success converges to the true proportion. This is a demonstration of WLLN when we are interested in approximating probabilities of some event

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

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

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

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

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

In [40]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [41]:
```
g(x) = exp.(-x.^3)
exact_integral, er = quadgk(g,0,Inf)
println(exact_integral)
```

```
0.8929795115692495
```

In [42]:
```
psi(x) = exp.(x.-x.^3)
M = 1000
x = rand(Exponential(1), M)
approx_integral = mean(psi(x))
println(approx_integral)
```

```
0.8768311765964493
```

In [43]:
```
se = sqrt(var(psi(x))/M)
println("Standard error of the approximation is : ", se)
```

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

In [44]:
```
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 [45]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings, StatsBase, QuadGK
```

In [46]:
```
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 [47]:
```
println(first(approx_integral, 5))
```

```
[0.8914633500517565, 0.893945266454704, 0.8889845432970686, 0.8922001808693331,
0.885100534978075]
```

In [48]:
```
histogram(approx_integral, normalize = true, label = "",
xlabel = L"\widehat{I_M}", ylabel = "Density",title = "M = $M")
scatter!([exact_integral], [0], color = "red", marker = :circle,
       markersize = 10, label = "")
```

Out[48]:

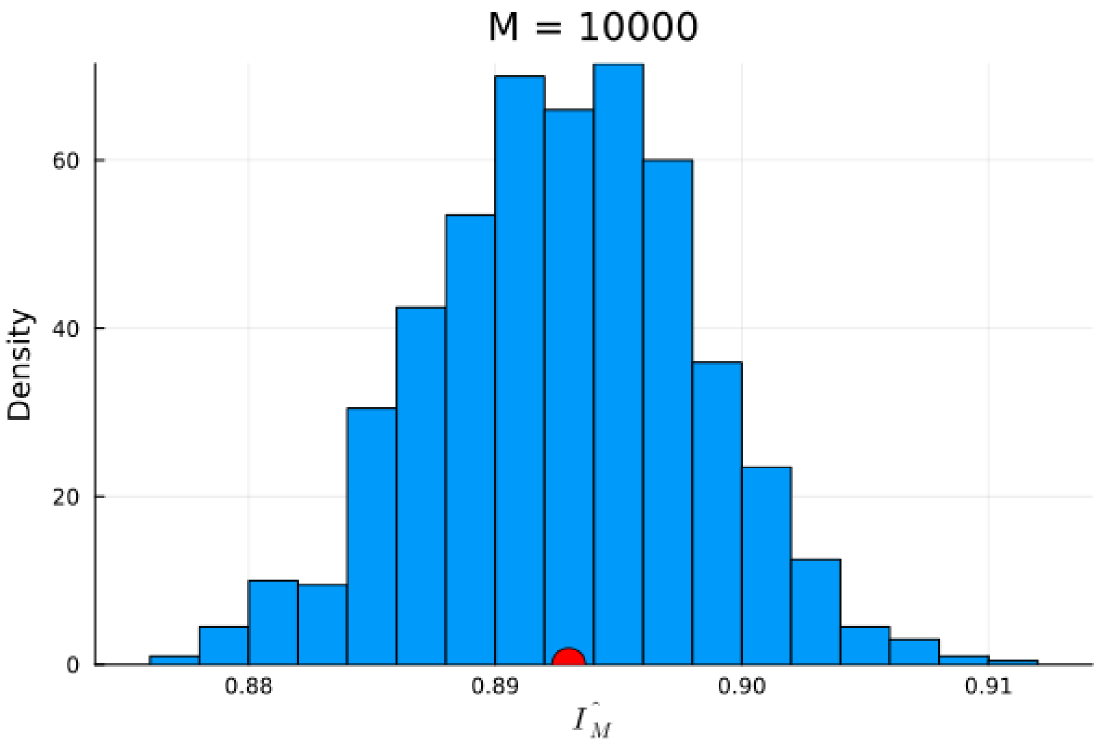

Figure 10: Approximate sampling distribution of the Monte Carlo estimator for the integral. It is clear that as M the histogram will be shrinked to the true value shown using red dot

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

In [49]:
```
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

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[49]:

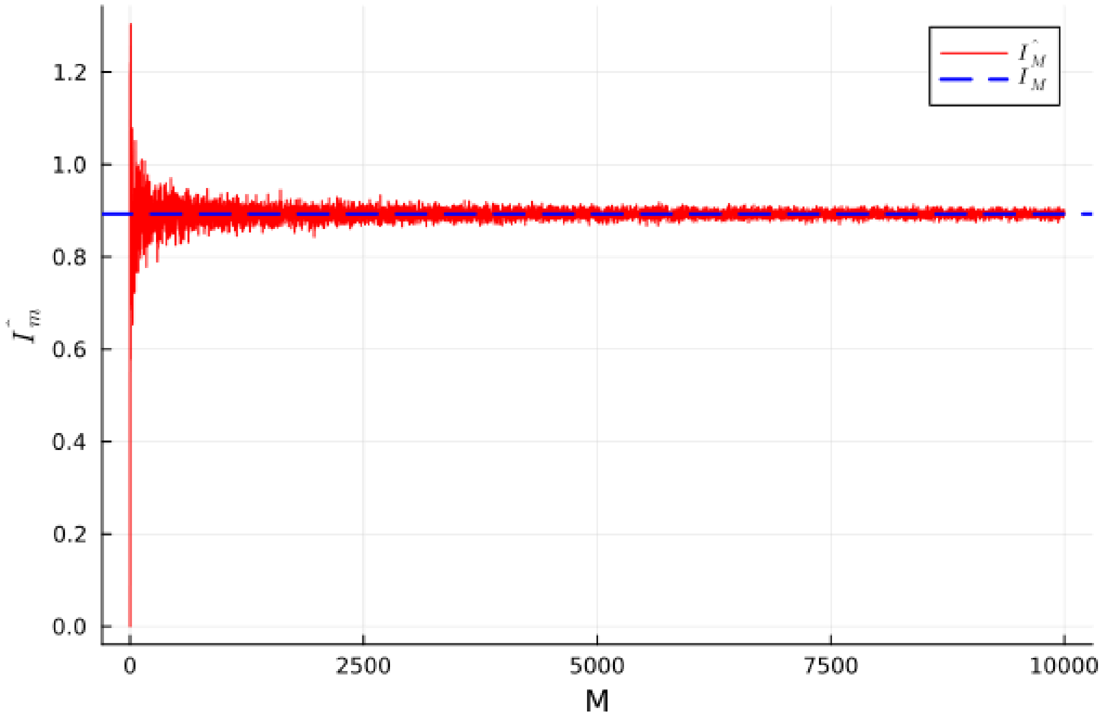

Figure 11: Shape of the sampling distribution of the Monte Carlo approximator of the integral. This gives us a visuals about the possible deviations of the Monte Carlo estimate of the integral. As M increases, the Monte Carlo integral conerges to the true integral. The idea is to show that the variance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

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

$$=$$

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

$$=$$

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

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

In [50]:

```
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

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[50]:

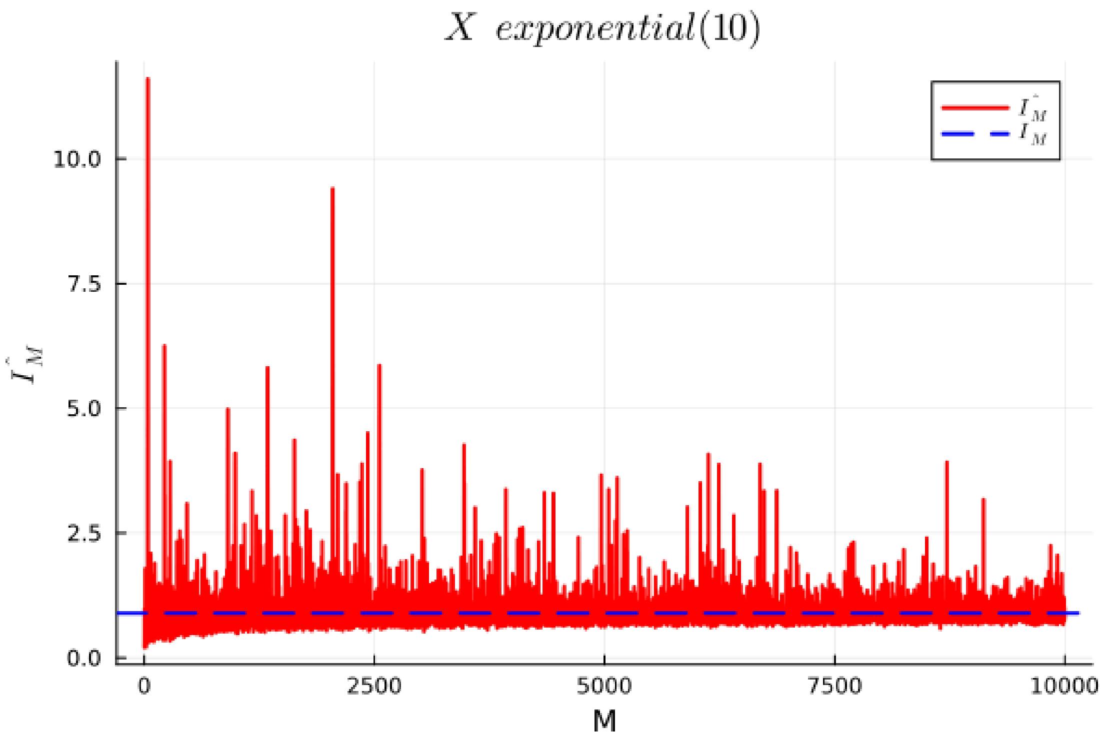

Figure 12: It is important to consider with respect to which density function we are writing the integral as an expected value. The same exercise, we have performed by writing the expected with respect to exponential density with large λ (mean) value. From the plot, it is clear that the convergence of 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 [51]:

```
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,
0.8860465709190601, 0.8954578033416376]
```

In [52]:

```
p1 = histogram(I_M_10, normalize=true, label=false,
xlabel=L"\widehat{I_M}", ylabel = "Density" ,title= "Exponential(10)")

scatter!([exact_integral], [0], markershape=:circle,
       color="red", markersize=10, label="I")

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

scatter!([exact_integral], [0], markershape=:circle, color="red",
         markersize=10, label="I")

plot(p1,p2, layout = (1,2), size = (800, 500))
```

Out[52]:

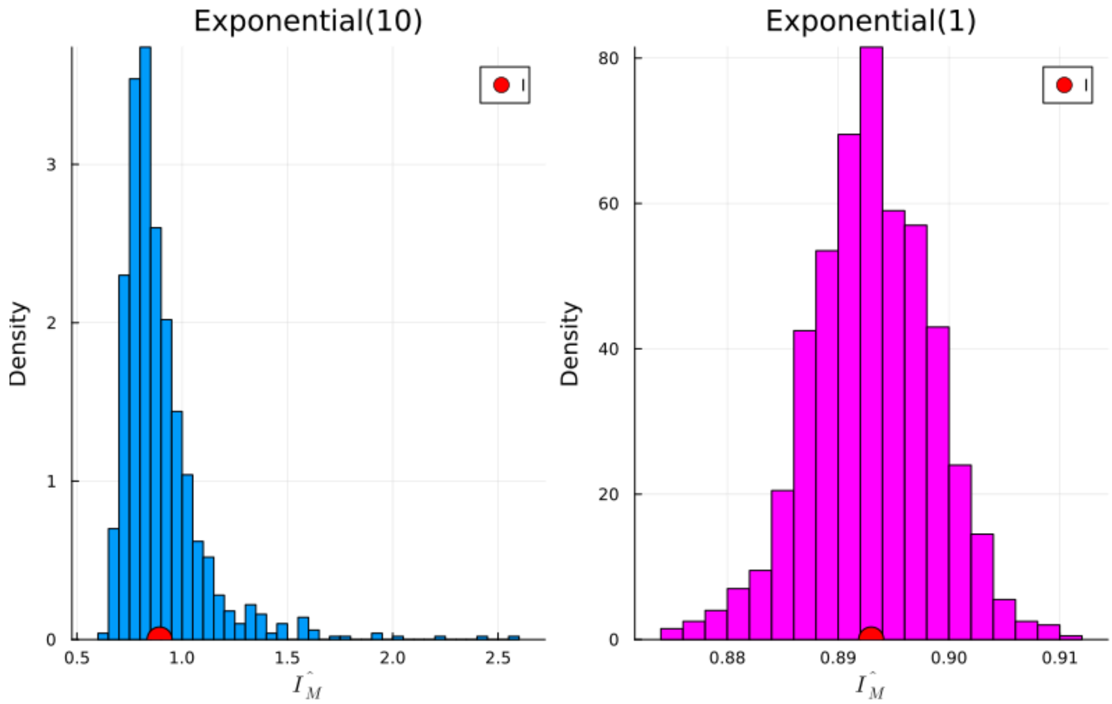

Figure 13: The integral has been evaluated based $M = 1000$ monte carlo simulation. It is important to note that the Monte carlo approximate using $\exp(1)$ as the support density gives more closeness to the true value as compared to the $\exp(10)$ density function. We shall see later what could be a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

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

In [54]:
```
g(x) = (1 + x.^2 + x.^3).^(-1) .* (x .> 0) # function
plot(g, -1, 6, color="red", lw=2, label="")
```

Out[54]:

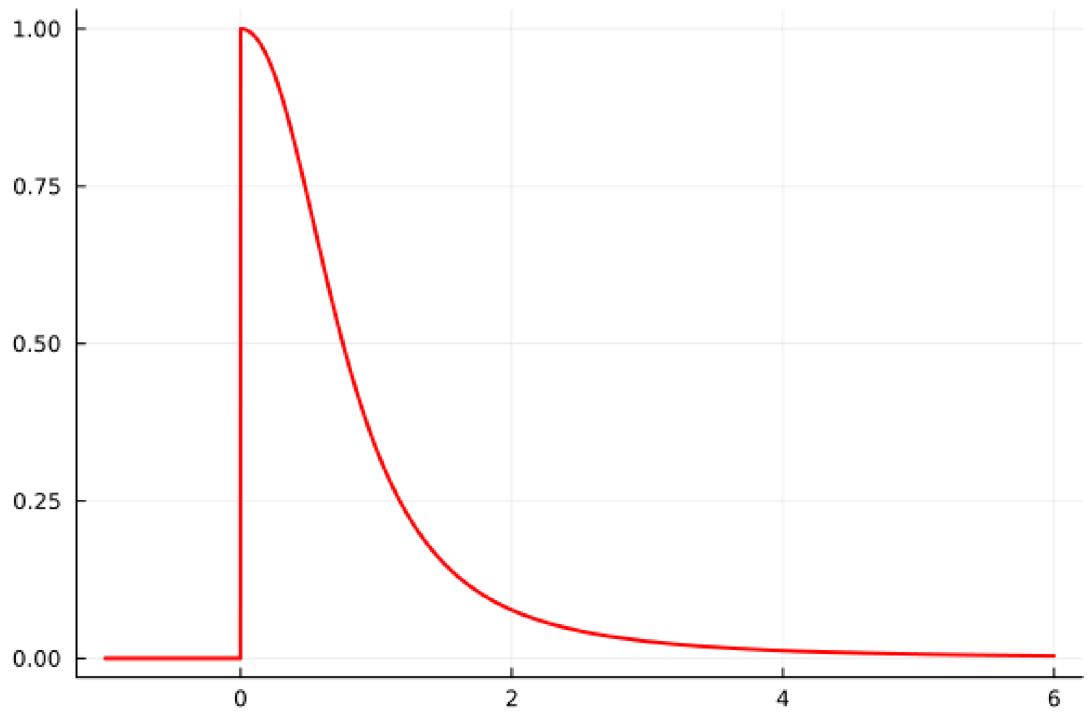

Figure 14 : The plot represents the function $g(x) = \frac{1}{1+x^2+x^3}$ over the range $[-1, 6]$. The red curve illustrates the behavior of $g(x)$, which is used in the integral approximation via Monte Carlo methods with an exponential sampling distribution.

In [55]:
```
I , er =  quadgk(g, 0, Inf) # er = numerical error
println("exact integral: ", I)
```

```
exact integral: 0.9693439633068099
```

In [56]:
```
lambda = 0.05
psi(x) = exp.(lambda .* x) .* (g(x) / lambda)

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

```julia
rep = 1000

fig = plot(layout=(3, 4), size=(800, 500))

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", ylabel = "Density",legend=false,
    xlims = extrema(I_n),subplot=idx)
    scatter!([mean(I_n)], [0], color= "red", marker=:circle,
             markersize = 10 ,subplot=idx)
end

display(fig)
```

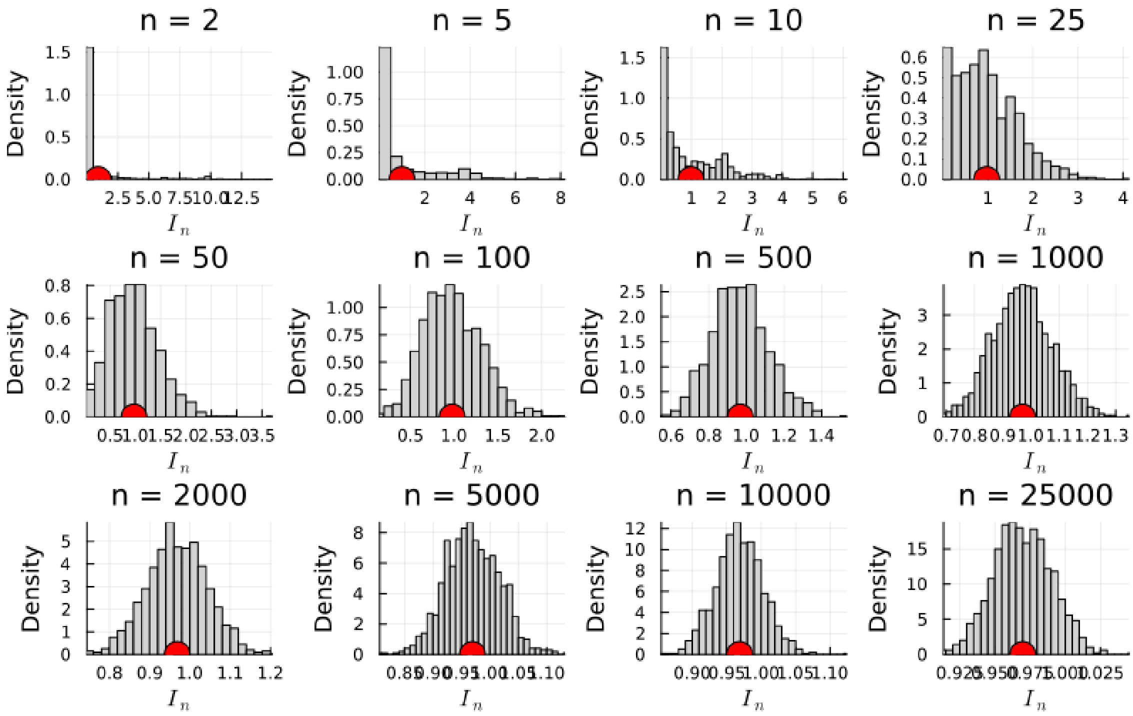

Figure 15: Histogram of the Monte Carlo estimate I_n for various sample sizes n. As $n \to \infty$, the distribution of I_n converges to a normal distribution centered at the expected value $\mathbb{E}[I_n]$, demonstrating the law of large numbers and the central limit theorem.

Exercises

Q.1. Using Monte Carlo integration, compute the value of the integral

$$\int_0^1 \sin(x^5)\, dx$$

Compare the result obtained using numerical integration in **Julia**.

Q.2. Consider the definite integral

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

- Perform the integration on the interval $[0, 1]$ using Monte Carlo method by simulating uniform random numbers from the interval $[0, 1]$. Explicitly write down the steps involved in it.
- Perform the integration on the interval $[0, 1]$ using Monte Carlo method by simulating Exponential(1) random numbers. Explicitly write down the steps involved in the method.
- Let I_1 and I_2 be the values of the integral by using strategy 1(a) and 1(b), respectively. Using 1000 replications, estimate the variance associated with your approximation. Which method performs better? Consider $n = 25$.
- Vary $n = 1, 2, \ldots, 100$. Based on 1000 replications, compute the mean and variance of I_1 and I_2 for all n. Plot the variance of I_1 and I_2 against n in a single plot. For $n = 100$, draw histogram based on these I_1 values and overlay the appropriate normal density function plot.
- Without using multiple replications, approximate the variance of I_1 and I_2 using the usual formula for the variance:

  $$\mathbb{E}(g(X)^2) - (\mathbb{E}(g(X)))^2$$

  For I_1, $X \sim \text{Uniform}(0, 1)$ and for I_2, $X \sim \text{Exponential}(1)$. Here, the integral (expectation) $\mathbb{E}(g(X)^2)$ can be evaluated again using Monte Carlo integration (provided the variance exists). Do the computation for $n = 1, 2, \ldots, 100$ and plot them against n. Use different color to plot them and add appropriate legend.

Q.3. Let $X \sim \text{Exp}(\lambda)$ with mean $\lambda = 2$. Estimate the probability $\mathbb{P}(X > 3)$ by simulating random numbers from the exponential density function. Compare the result obtained using numerical integration in **Julia**.

Q.4. Suppose X has the following probability density function:

$$f(x) = \begin{cases} 3e^{-3x}, & 0 < x < \infty \\ 0, & \text{otherwise} \end{cases}$$

Using simulation, based on a random sample of size $n = 100$, approximate the probability
$\mathbb{P}(0 < X < 2)$.
For each $n = 1, 2, \ldots, 1000$, approximate the probability and plot them against different values of n.
Write your observations and state the relevant theorem(s).

Q.5.

- Suppose $X_1, X_2, \ldots, X_n \sim \text{Exp}(1)$.
  Obtain the exact sampling distribution of the sample mean $\bar{X}_n$ and identify the family.

- Using $m = 1000$ replications, obtain the sampling distribution of $\bar{X}_n$ for $n = 10$.
- Using Julia, show that $\bar{X}_n$ converges in probability to 1.
- Based on a sample of size n from the $\mathrm{Exp}(1)$ density function, compute the probability $\mathbb{P}(X > 1)$.
- Obtain the variance associated with the approximation. Give an estimate of the variance based on sample of size n. Plot the estimated variance as a function of n. As $n \to \infty$, does the variance converge?

Q.6. Obtain the sampling distribution of

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \quad \text{where} \quad h(x) = \exp(x - x^3)$$

and $X_1, X_2, \ldots, X_n \sim \mathrm{Exp}(1)$.
Comment on its shape for large n. Is it close to normal distribution? If so, what are the means and variance?

- In the above problem, we have considered $f(x) = \exp(-x)$, that is, we have taken the expectation with respect to the exponential(1) density function.
  We could actually write down the integral as an expected value in several other ways as well:

  $$\int_0^\infty \frac{1}{2} e^{2x - x^3} \cdot 2e^{-2x}\, dx = \mathbb{E}(h(X)), \quad X \sim \text{Exponential}(2)$$

  where $h(x) = \frac{1}{2} e^{2x - x^3}$.
  Do the same exercise using exponential(2) density function and compare the two approximations.
- For $n = 4, 5, 10, 20$, approximate the sampling distribution of $\frac{1}{n}\sum_{i=1}^{n} h(X_i)$ using both ways and draw these two histograms in a single plot.
- Vary $n = 1:1000$ and show the values of $\frac{1}{n}\sum_{i=1}^{n} h(X_i)$ for each n. Create two different plots and identify the difference.
- What do you expect if you do same experiment with exponential(10)?

Q.8. Suppose that we are interested in approximating the integral

$$I = \int_0^\infty \frac{1}{\pi(1 + x^2)} dx$$

Note that this is the probability $\mathbb{P}(X > 2)$ where $X \sim \mathcal{C}(0, 1)$, standard Cauchy distribution.

We consider four different approaches to estimate the integral, given as:

- $I_1 = \frac{1}{n}\sum_{j=1}^{n} \mathbb{I}_{X_j > 2}$

- $I_2 = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{\pi(1+X_j^2)}$

Express the integral as

$$I = \frac{1}{2} - \int_0^2 \frac{1}{\pi(1+x^2)}dx$$

and consider the approximation

$$I_3 = \frac{1}{2} - \frac{1}{n}\sum_{j=1}^{n}\frac{1}{\pi(1+U_j^2)},$$

where $U_1, U_2, \ldots, U_n \sim \text{Uniform}(0,2)$.

Show that the required integral can also be written as

$$I = \mathbb{E}\left[\frac{2}{\pi(1+(1-U_j)^2)}\right]$$

and consider the approximation

$$I_4 = \frac{1}{n}\sum_{j=1}^{n}\frac{2}{\pi(1+(1-U_j)^2)},$$

where $U_1, U_2, \ldots, U_n \sim \text{Uniform}(0,1)$.

- Compute the variance of the approximations for I_j, $1 \leq j \leq 4$, and comment which approximation is the best among these four.
- For $n = 100$, obtain the sampling distributions of I_j, $1 \leq j \leq 4$ using $M = 10^5$ replications.
- Interpret the histograms in terms of their accuracy.
- Justify if the evaluation of the integral by I_4 requires approximately 32 times fewer simulations than I_1 to achieve the same precision.

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.

- Rubinstein, R. (1981). Simulation and the Monte Carlo Method. John Wiley, New York.