Lab 5 Solution

2025-02-05

Problem 1

To generate samples from $N(0,1)$ distribution using the method of ratio of uniforms.

So, to generate samples using the ratio of uniforms , we need to find

$C = \{(u,v) \mid 0 \leq u \leq \sqrt(f(\frac{v}{u}))\}$ and try to enclose C in a rectangle $D[0,a] \times [b,c]$ such that $0 \leq u \leq a$ and $b \leq v \leq c$. To find a, b and c we have $a = \sup_x f^{1/2}(x)$, $b = \inf_{x \leq 0} x f^{1/2}(x)$ and $c = \sup_{x \geq 0} x f^{1/2}(x)$. Here the given distribution is $N(0,1)$ with density $f(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$. So, on substituting the values of $f^{1/2}(x)$ and calculating a, b and c we get, $a = (\frac{1}{2\pi})^{1/4}$, $b = -(\frac{2}{\pi e^2})^{1/4}$ and $c = (\frac{2}{\pi e^2})^{1/4}$.

So, by using the above values of a, b and c, we can draw samples from $N(0,1)$ using the method of ratio of uniforms.

```r
Normal_ROU <- function(a, b, c) {
  while (TRUE) {
    # Generating samples from Uniform(0,1)
    u1 <- runif(1)
    u2 <- runif(1)

    # Converting the generated samples into required u and v
    u <- a * u1
    v <- (c - b) * u2 + b

    # Checking the condition for ROU
    sqrt_f <- (1 / (2 * pi))^(1 / 4) * exp(-(v / u)^2 / 4)

    # Acceptance condition
    if (u <= sqrt_f) {
      return(v / u)
    }
  }
}
```

The above function generates samples from $N(0, 1)$ for appropriate values of a, b and c .

```
# Number of samples to be generated
n <- 1e4

# Array to store the generated samples
norm_arr <- numeric(n)

# Assigning the proper values for a, b and c
a <- (1/(2*pi))^(1/4)
b <- -(2/(pi * exp(1)^2))^(1/4)
c <- (2/(pi * exp(1)^2))^(1/4)

# Generating required number of samples
for (i in 1:1e4){
  norm_arr[i] <- Normal_ROU(a, b, c)
}

# Mean and variance of the generated sample
mean(norm_arr)
```

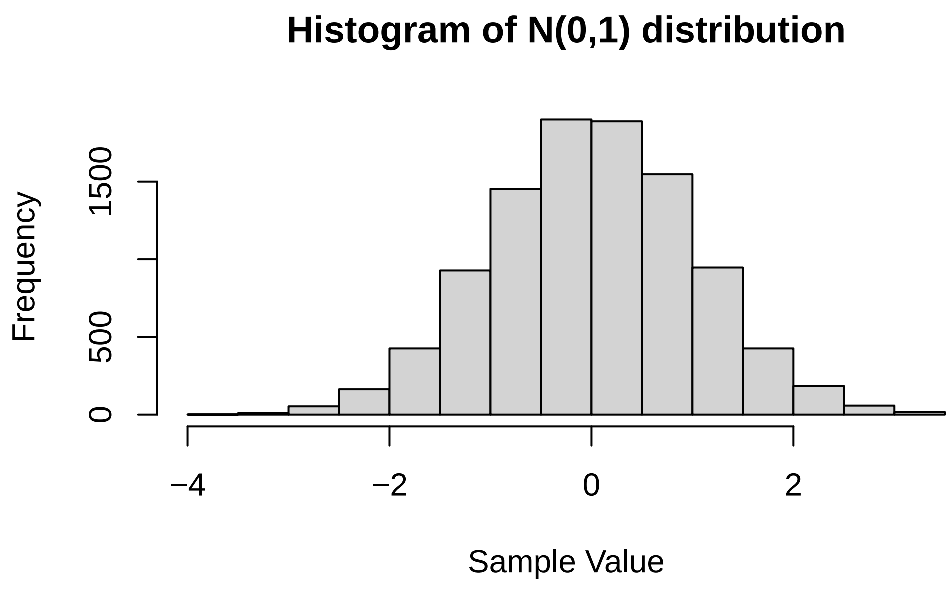
```
[1] 0.01804426
```

```
var(norm_arr)
```

```
[1] 1.022246
```

From the above code, we can see that the sample mean is quite close to the population mean (0 in this case) and the sample variance is quite close to the population variance (1 in this case).

```
# Histogram of the generated values
hist(norm_arr, main = "Histogram of N(0,1) distribution",
     xlab = "Sample Value", ylab = "Frequency", )
```



Problem 2

Using simple Monte Carlo Sampling to evaluate the given integrals.

a)

```r
# Number of samples to be generated
n <- 1e5

# Generating n samples from Uniform(0,1)
x <- runif(n, min = 0, max = 1)

# Calculating h(xi) for all xi element of x
h_x <- exp(x)

# Finding the mean of h(xi), which is in turn the value of the integral
calculated_expectation <- mean(h_x)

# Analytical value of the integral
h_analytical <- exp(1) - 1
```

```
# Output results
cat("Monte Carlo estimate for Integral :", calculated_expectation, "\n")
```

```
Monte Carlo estimate for Integral : 1.718143
```

```
cat("Analytical result for Integral :", h_analytical, "\n")
```

```
Analytical result for Integral : 1.718282
```

```
cat("Absolute error:", abs(calculated_expectation - h_analytical), "\n")
```

```
Absolute error: 0.0001392642
```

The above code uses Monte Carlo sampling technique to evaluate the integral $\int_0^1 e^x$. From the results obtained in the above code, we can say that the value obtained in the Monte Carlo sampling technique is quite close to that of the analytical value of the integral as the absolute error tends to 0.

b)

```
# Number of samples to be generated
n <- 100000

# Generating samples from Uniform(0,1)
y <- runif(n, 0, 1)

# Correctly scaling the generated sample to get the required sample
x <- pi*y

# Define the function to integrate
f_x <- sqrt(x^3 + sqrt(x)) - x^2 * sin(4*x)

# Monte Carlo estimate
I_mc <- mean(f_x) * pi

# Compute the correct analytical result
I_analytical <- (pi^2)/4 + (2/5) * (pi^(5/4) * sqrt(1 + pi^(5/2)) + asinh(pi^(5/4)))

# Output results
cat("Monte Carlo estimate for Integral 2:", I_mc, "\n")
```

```
Monte Carlo estimate for Integral 2: 10.46948
```

```r
cat("Analytical result for Integral 2:", I_analytical, "\n")
```

```
Analytical result for Integral 2: 10.51721
```

```r
cat("Absolute error:", abs(I_mc - I_analytical), "\n")
```

```
Absolute error: 0.04772875
```

The above code uses Monte Carlo sampling technique to evaluate the integral $\int_0^{\pi}(\sqrt(x^3 + \sqrt{x}) - x^2 sin(4x))dx$. From the results obtained in the above code, we can say that the value obtained in the Monte Carlo sampling technique is quite close to that of the analytical value of the integral as the absolute error tends to 0.