IMPORTANCE SAMPLING in MONTE CARLO INTEGRATION

Aim :- To compute integral of $f(x) = e^{-x}$ from $[0,1]$ using standard Monte Carlo integration & Importance Sampling.

Theory :-

- Monte Carlo uses random uniform sampling to estimate the value of integration.
- This is inefficient if function has large variation.
- Importance Sampling improves efficiency by sampling from a distribution $p(x)$ similar to $f(x)$ and weighing samples as $f(x)/p(x)$.

Method :-

1. Define $f(x)$
2. For standard monte carlo, sample $x_i \sim U[0,1]$ & compute the average $I = \frac{1}{N} \sum f(x_i)$
3. For importance sampling choose a suitable PDF $p(x)$ & sample $x_i \sim p(x)$.
4. Compute $I = \frac{1}{N} \sum f(x_i)/p(x_i)$
5. Compare the results.

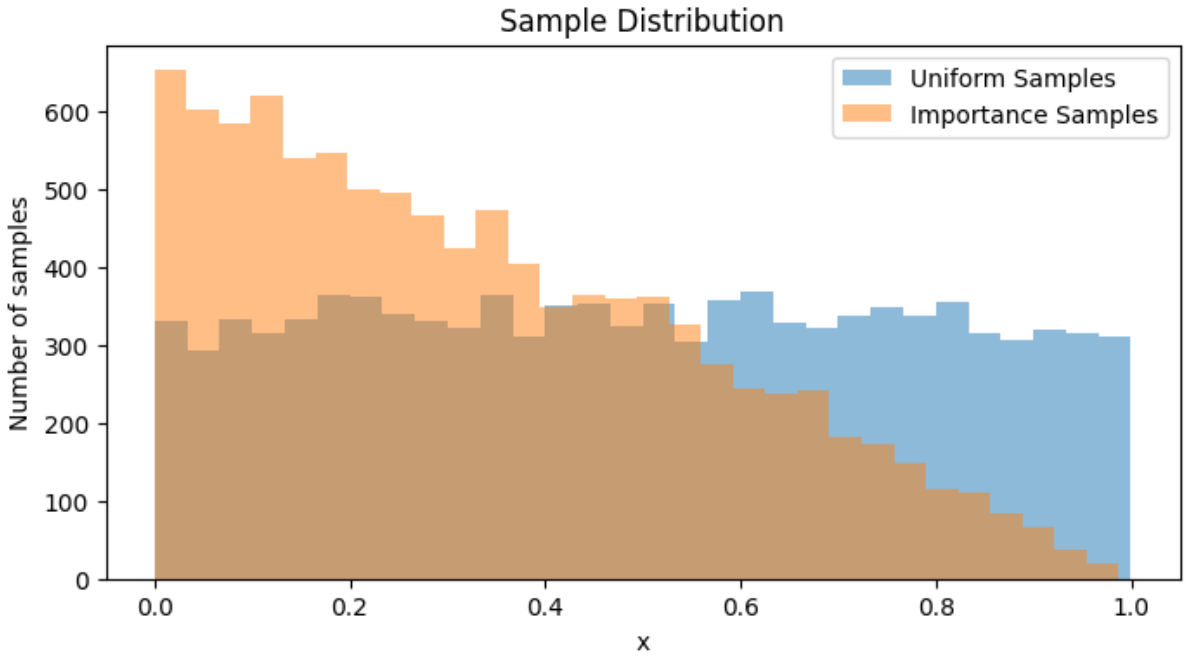
Results :- Choosen $p(x) = \dfrac{e^{-x}}{(1-\frac{1}{e})}$ → Highly Similar to e^{-x}

Error in Integration (i) by Standard Monte Carlo = 6.8×10^{-4}

(ii) by Importance Sampling = 2.22×10^{-16}

Code:

```
1. import numpy as np
2. import matplotlib.pyplot as plt
3.
4. # Function to integrate
5. def f(x):
6. return np.exp(-x)
7.
8. true_value = 1 - np.exp(-1)
9. N = 10000
10.
11.# Standard Monte Carlo
12.x_uniform = np.random.uniform(0, 1, N)
13.I_uniform = np.mean(f(x_uniform))
14.print("Standard Monte Carlo Estimate:", I_uniform)
15.print("Error:", abs(I_uniform - true_value))
16.
17.# Importance Sampling
18.def p(x):
19.return np.exp(-x) / (1 - np.exp(-1))
20.
21.# Inverse CDF sampling for p(x)
22.u = np.random.uniform(0, 1, N)
23.x_importance = 1 - np.sqrt(1 - u)
24.
25.# Weighted estimate
26.weights = f(x_importance) / p(x_importance)
27.I_importance = np.mean(weights)
28.print("Importance Sampling Estimate:", I_importance)
29.print("Error:", abs(I_importance - true_value))
30.
31.# Visualization
32.plt.figure(figsize=(8,4))
33.plt.hist(x_uniform, bins=30, alpha=0.5, label='Uniform Samples')
34.plt.hist(x_importance, bins=30, alpha=0.5, label='Importance Samples')
35.plt.xlabel('x')
36.plt.ylabel('Number of samples')
37.plt.title('Sample Distribution')
38.plt.legend()
39.plt.show()
```


Sample Distribution
Uniform Samples
Importance Samples
Number of samples
0
100
200
300
400
500
600
0.0
0.2
0.4
0.6
0.8
1.0
x