MCexample2

March 13, 2025

[1]:
```r
# Monte Carlo Integration with Importance Sampling (compare to MCexample1)

# Define the original function f(x)
forig <- function(x) {
  return(x^(-1/3) + x/10)
}
```

[2]:
```r
# Define the importance sampling distribution p(x)
p <- function(x) {
  (2/3) * x^(-1/3)
}
```

[3]:
```r
# Define the transformed function g(x) = f(x) / p(x)
# you can simplify g(x) = 1.5 * (1 + 0.1 * x^(4/3))
ftrans <- function(x) {
  forig(x) / p(x)
}
```

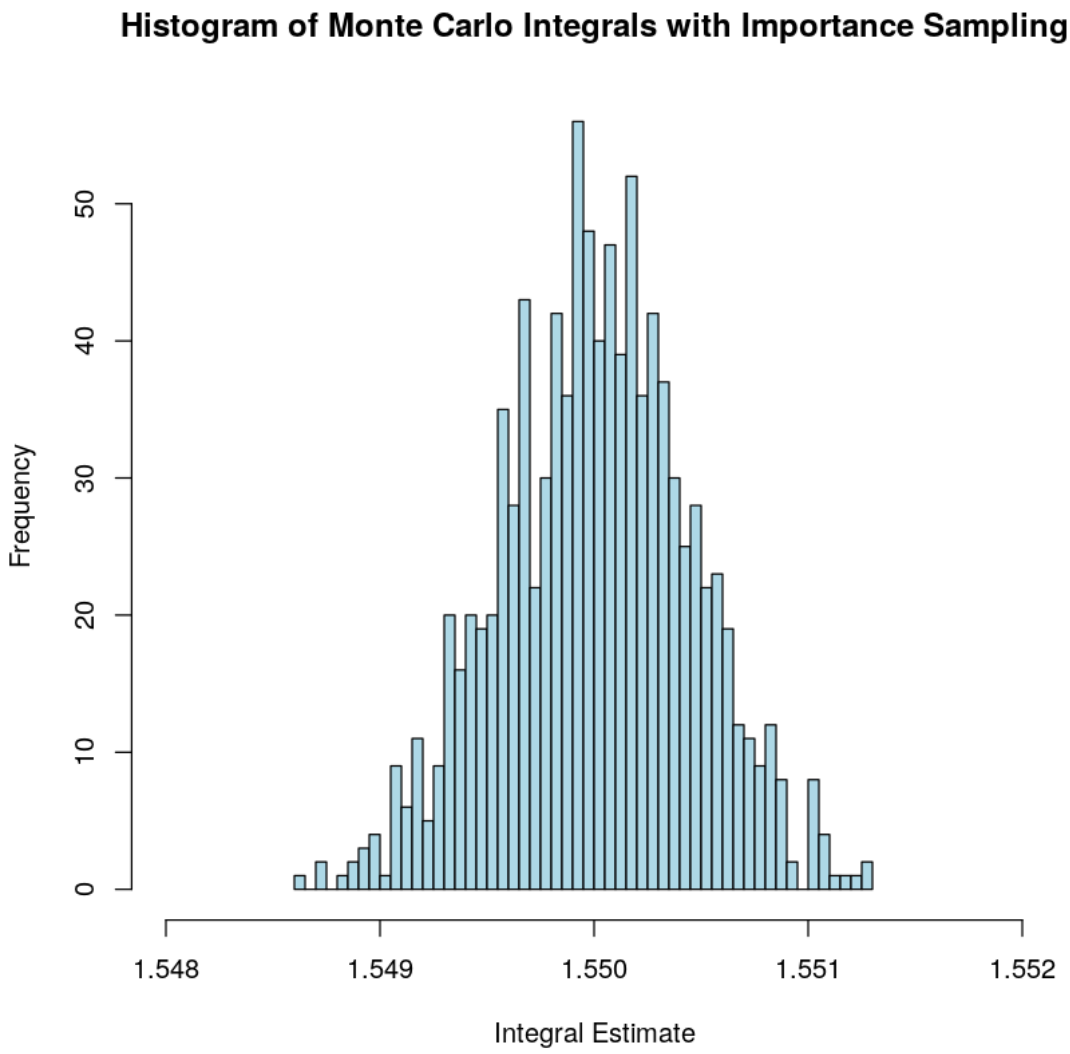
[4]:
```r
# Parameters:
N <- 10000     # Number of samples per Monte Carlo run
M <- 1000      # Number of Monte Carlo runs
set.seed(12345)  # if reproducibility is required

# Set up some arrays to store the result of the runs
integral_estimates <- numeric(M)
g_squared_means <- numeric(M)
```

[5]:
```r
# Monte Carlo Integration Loop
for (i in 1:M) {
  # Step 1: Generate uniform random samples u
  u <- runif(N)

  # Step 2: Transform the samples using the inverse CDF of p(x)
  x <- u^(3/2)

  # Step 3: Compute the transformed function g(x)
  g <- ftrans(x)
```

```
  # Step 4: Store the mean and mean squared of g(x)
  integral_estimates[i] <- mean(g)
  g_squared_means[i] <- mean(g^2)
}
```

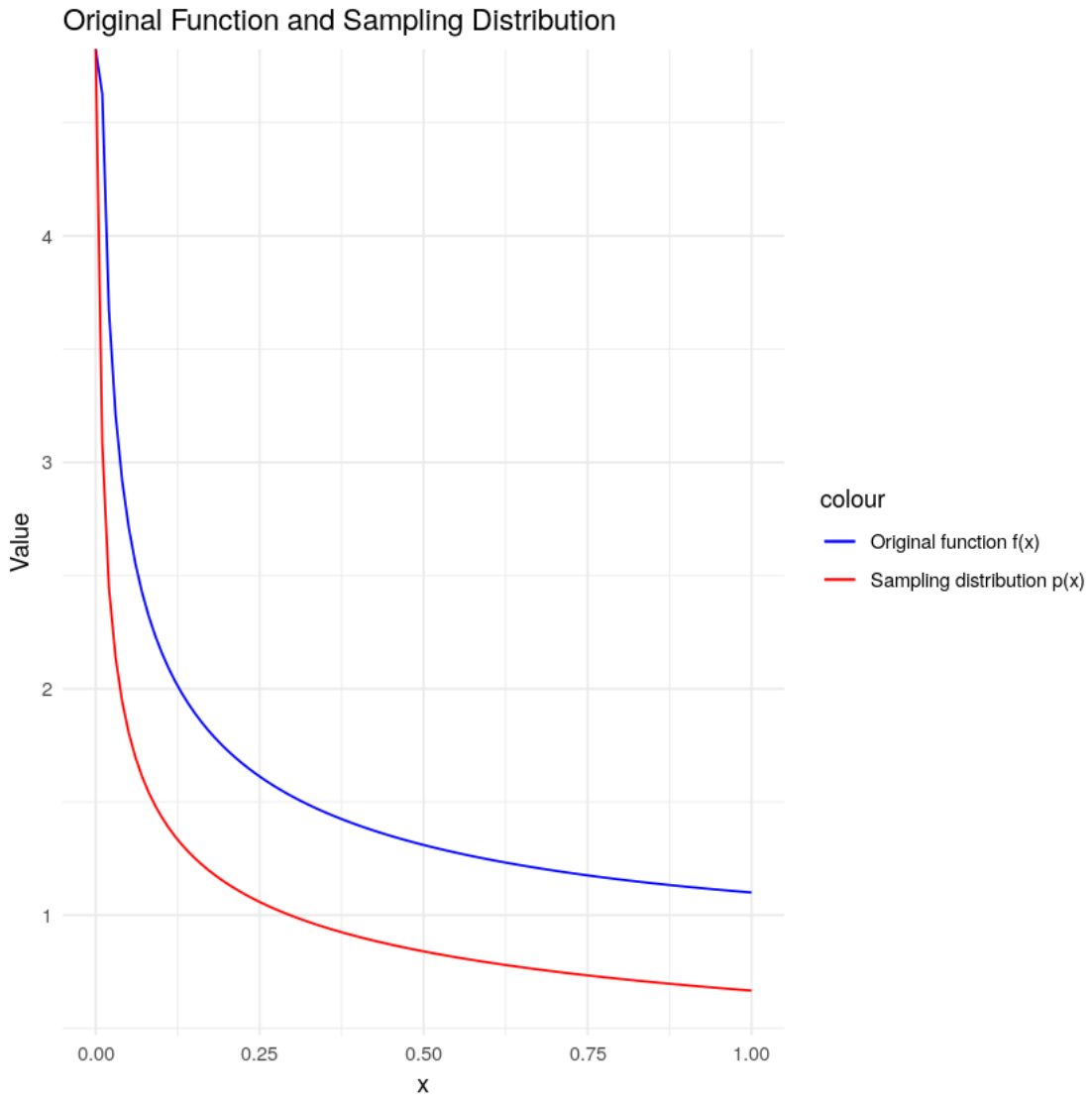
[6]:
```
# Plot Histogram of the Monte Carlo Integrals
hist(integral_estimates, breaks = 50, col = "lightblue", border = "black",
     main = "Histogram of Monte Carlo Integrals with Importance Sampling",
     xlab = "Integral Estimate", xlim=c(1.548,1.552))

# Print Results
cat("Mean of MC integrals:                                    ",␣
 ↪mean(integral_estimates), "\n")
cat("Standard deviation of MC integrals:                      ",␣
 ↪sd(integral_estimates), "\n")
cat("Analytical estimate of standard deviation of MC integrals:", 0.0448 /␣
 ↪sqrt(N), "\n")
```

```
Mean of MC integrals:                                    1.55001
Standard deviation of MC integrals:                      0.0004448448
Analytical estimate of standard deviation of MC integrals: 0.000448
```



[7]:
```r
# Plot the original function f(x) and the importance sampling distribution p(x)
x_vals <- seq(0, 1, length.out = 100)
df <- data.frame(
  x = x_vals,
  f_x = forig(x_vals),
  p_x = p(x_vals)
)

# Plot f(x) and p(x)
library(ggplot2)
p1 <- ggplot(df, aes(x)) +
  geom_line(aes(y = f_x, color = "Original function f(x)")) +
  geom_line(aes(y = p_x, color = "Sampling distribution p(x)")) +
```

```
  labs(title = "Original Function and Sampling Distribution", y = "Value") +
  scale_color_manual(values = c("Original function f(x)" = "blue", "Sampling␣
 ↪distribution p(x)" = "red")) +
  theme_minimal()

# Display the plot
print(p1)
```



[ ]: