Final Assignment NP

Arthur WEHBE

2024-05-14

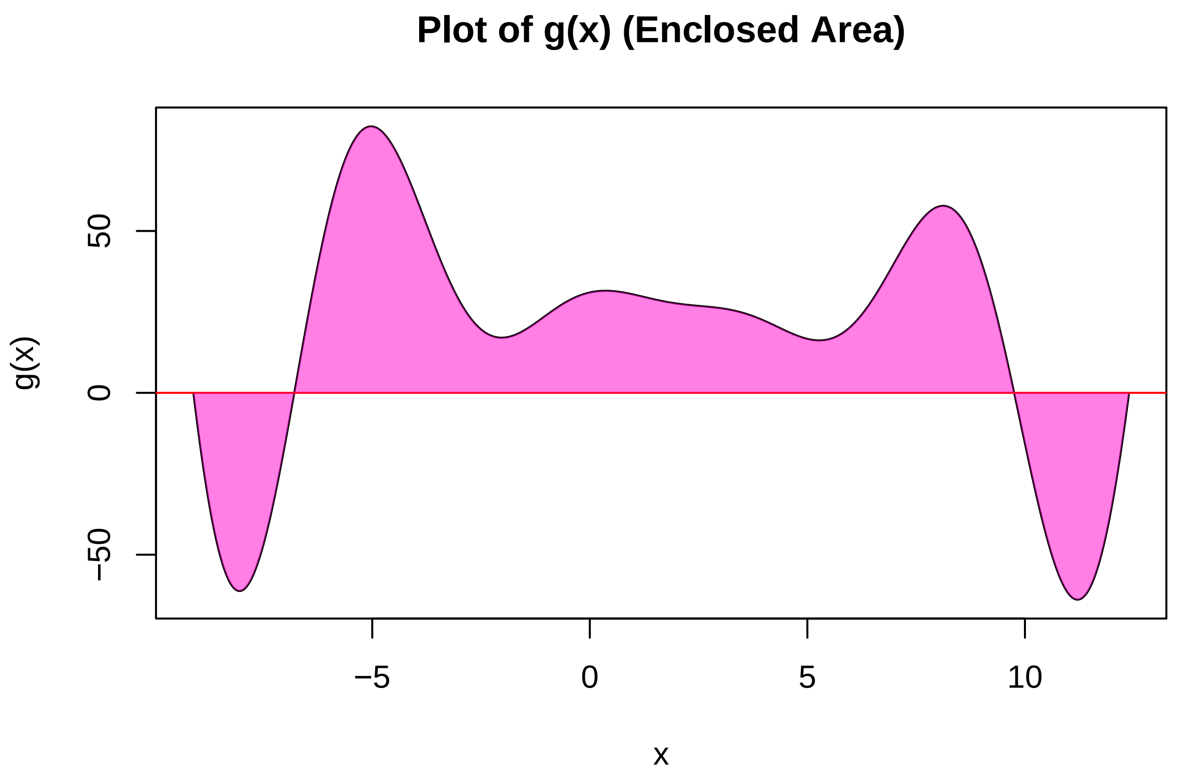
Question 1 : Maths and Stats in R

Here we define g and plot its enclosed area in pink and calculate the roots and enclosed area with nulerical integration

```r
library(pracma)
```

```
## Warning: le package 'pracma' a été compilé avec la version R 4.3.3
```

```r
g <- function(x) {
  sin(x) * (x - 2)^2 - x + cos(x) + 30
}

#those roots are the limit of the enclosed area
root1 <- -9.11
root2 <- 12.39

x_values_enclosed <- seq(root1, root2, by = 0.01)

abs_g_enclosed <- function(x) abs(g(x))

total_area_enclosed <- integrate(abs_g_enclosed, root1, root2)$value

plot(x_values_enclosed, g(x_values_enclosed), type = "l", main = "Plot of g(x) (Enclosed Area)", xlab =
abline(h = 0, col = "red")

polygon(c(x_values_enclosed, rev(x_values_enclosed)),
        c(pmin(g(x_values_enclosed), 0), rep(0, length(x_values_enclosed))),
        col = rgb(1, 0, 0.8, alpha = 0.5), border = NA)

polygon(c(x_values_enclosed, rev(x_values_enclosed)),
        c(pmax(g(x_values_enclosed), 0), rep(0, length(x_values_enclosed))),
        col = rgb(1, 0, 0.8, alpha = 0.5), border = NA)
```



```
cat("Total enclosed area:", total_area_enclosed, "\n")
```

```
## Total enclosed area: 778.8835
```

Question 2 : Inversion Method

We define lambda and simulate 100000 samples with the inversion method then calculate the expected time with monte carclo then finally calculate the probability all 5 machine breakdown before 20 days.

```
# 1.
lambda <- 1/15
n <- 5
set.seed(123)
time_to_breakdown <- function(x) {
  (1 - exp(-lambda * x))^n
}

# 2.
u <- runif(100000)
samples <- -log(1 - u^(1/n)) / lambda

# 3.
expected_time <- mean(samples)
expected_time
```

```
## [1] 34.21228
```

```
# 4.
probability_20_days <- mean(samples >= 20)
probability_20_days
```

```
## [1] 0.78294
```

Question 3 : Rejection sampling

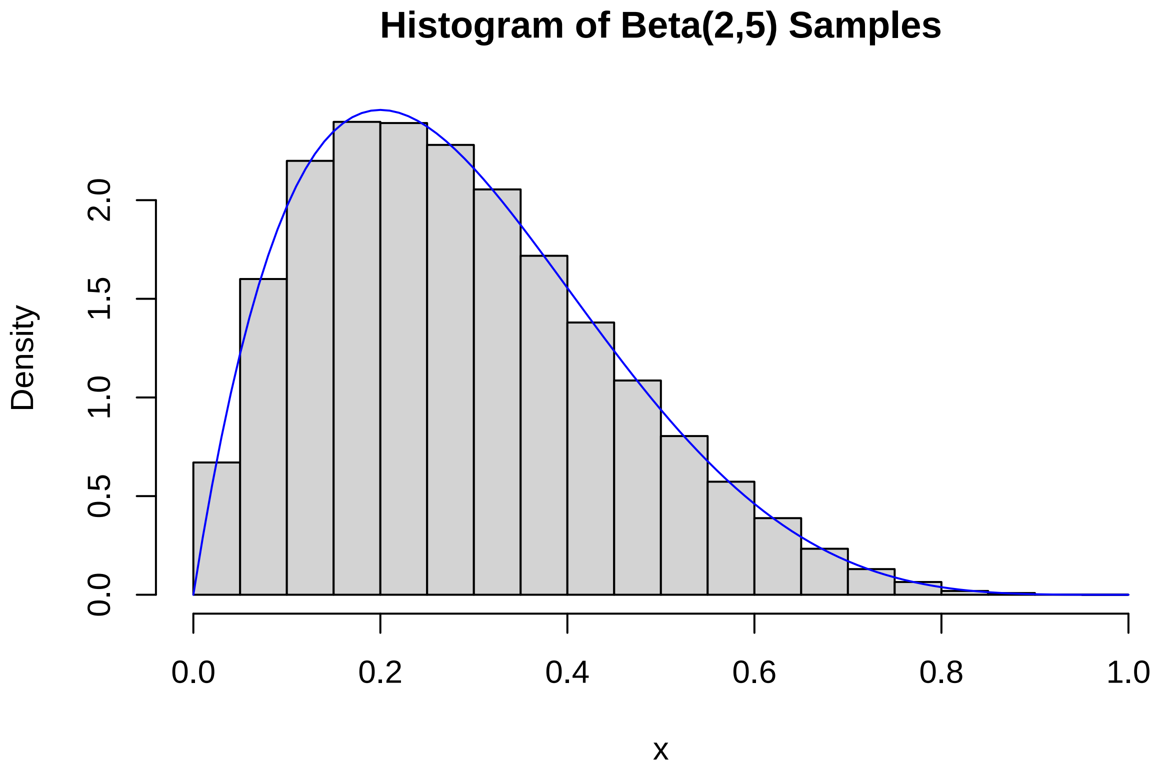
We first find the value of the max M then use it to produce 100000 samples from the beta distribution. We then plot the beta density overlaid by the curve to confirm how accurate the sample are. We estimate the mean of the samples with monte carlo.

The estimated mean is very close to the true mean, and the true mean falls within the 95% confidence interval. This indicates that our sampling method is accurate and the results are consistent with the theoretical properties of the Beta(2, 5) distribution. The small difference between the estimated mean and the true mean can be attributed to some randomness.

```
# 1.
beta_density <- function(x) dbeta(x, 2, 5)
x_seq <- seq(0, 1, by = 0.001)
M <- max(beta_density(x_seq))
cat("Maximum value of the Beta(2, 5) density (M):", M, "\n")
```

```
## Maximum value of the Beta(2, 5) density (M): 2.4576
```

```
# 2.
set.seed(123)
rejection_sampling <- function(n, beta_density, M) {
  samples <- numeric(n)
  count <- 0
  i <- 1
  while (i <= n) {
    y <- runif(1)
    u <- runif(1)
    if (u <= beta_density(y) / M) {
      samples[i] <- y
      i <- i + 1
    }
    count <- count + 1
  }
  list(samples = samples, acceptance_rate = n / count)
}

result <- rejection_sampling(100000, beta_density, M)
samples <- result$samples
acceptance_rate <- result$acceptance_rate

hist(samples, probability = TRUE, main = "Histogram of Beta(2,5) Samples", xlab = "x")
curve(dbeta(x, 2, 5), add = TRUE, col = "blue")
```



```
# 3.
estimated_mean <- mean(samples)
se <- sd(samples) / sqrt(length(samples))
ci <- estimated_mean + c(-1.96, 1.96) * se

cat("Estimated mean:", estimated_mean, "\n")
```

```
## Estimated mean: 0.2865136
```

```
cat("95% confidence interval for the mean:", ci, "\n")
```

```
## 95% confidence interval for the mean: 0.2855205 0.2875066
```

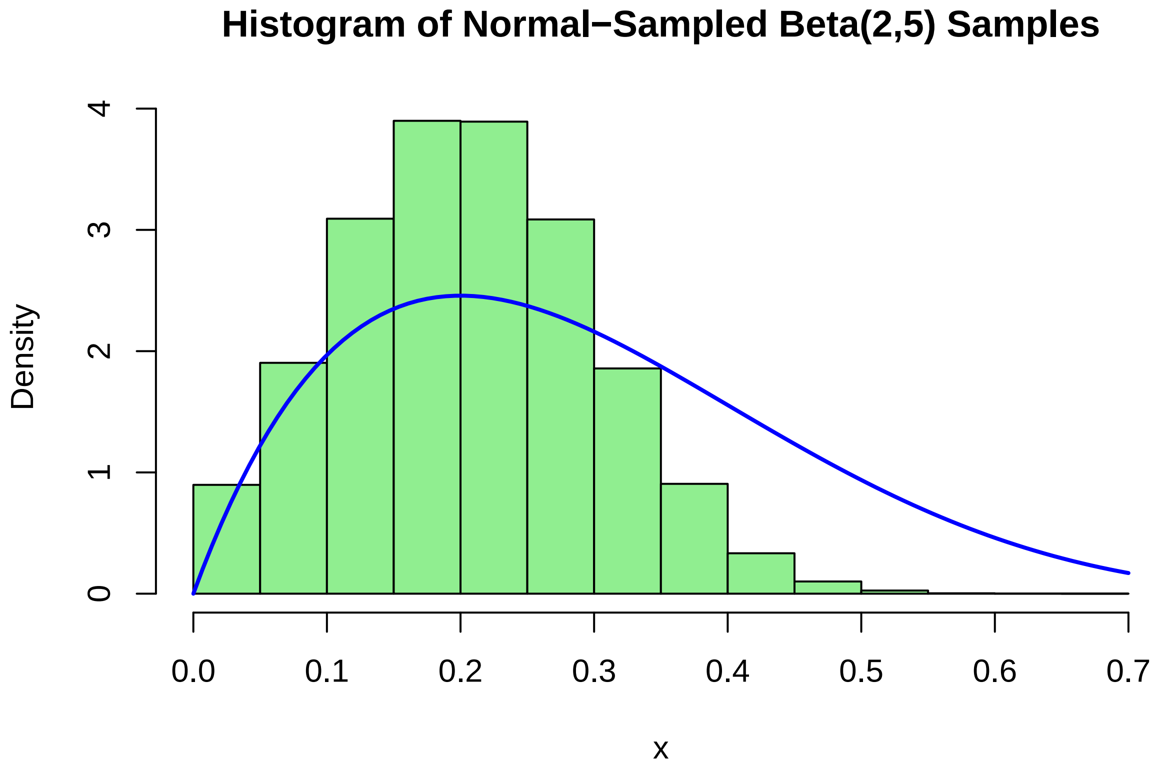
```
true_mean <- 2 / (2 + 5)
cat("True mean of Beta(2, 5):", true_mean, "\n")
```

```
## True mean of Beta(2, 5): 0.2857143
```

```
# 4.
normal_sampling <- function(n, mu, sigma) {
  samples <- numeric(0)
  while (length(samples) < n) {
    proposals <- rnorm(n, mean = mu, sd = sigma)
    proposals <- proposals[proposals >= 0 & proposals <= 1]
    samples <- c(samples, proposals)
  }
  samples <- samples[1:n]
  samples
}

normal_samples <- normal_sampling(100000, 0.2, 0.1)

normal_acceptance <- function(samples, beta_density, mu, sigma) {
  accepted <- mean(beta_density(samples) / dnorm(samples, mean = mu, sd = sigma))
  accepted
}

normal_acceptance_rate <- normal_acceptance(normal_samples, beta_density, 0.2, 0.1)
cat("Acceptance rate using normal distribution:", normal_acceptance_rate, "\n")
```

```
## Acceptance rate using normal distribution: 1.072673
```

```
hist(normal_samples, probability = TRUE, main = "Histogram of Normal-Sampled Beta(2,5) Samples", xlab =
curve(dbeta(x, 2, 5), add = TRUE, col = "blue", lwd = 2)
```



```
cat("Acceptance rate using uniform distribution:", acceptance_rate, "\n")
```

```
## Acceptance rate using uniform distribution: 0.4064842
```

```
cat("Acceptance rate using normal distribution:", normal_acceptance_rate, "\n")
```

```
## Acceptance rate using normal distribution: 1.072673
```