**Question 1  
Please run set.seed(6040) before running the analysis below.  
Implement a Monte Carlo simulation with M = 1, 000 repetitions to demonstrate that the number of data points selected in any bootstrap resample is about 63.2%. Consider a sample size N = 1, 000 for this Monte Carlo experiment.  
Note: no dataset is required to perform this analysis. Hint: R function unique() may be useful here.**

**Question 1: Bootstrap Resampling Demonstration**

This question aims to show that about 63.2% of data points are selected in any bootstrap resample. We will perform 1,000 repetitions with a sample size of 1,000.

# Set seed for reproducibility

set.seed(6040)

# Number of bootstrap samples

M <- 1000

# Sample size

N <- 1000

# Initialize a vector to store the number of unique points

unique\_counts <- numeric(M)

# Monte Carlo simulation

for (i in 1:M) {

# Generate a bootstrap sample with replacement

bootstrap\_sample <- sample(1:N, N, replace = TRUE)

# Count the number of unique points in the bootstrap sample

unique\_counts[i] <- length(unique(bootstrap\_sample))

}

# Calculate the proportion of unique points selected

proportion\_unique <- mean(unique\_counts) / N

# Print the result

cat("Proportion of unique data points in bootstrap samples:", round(proportion\_unique, 4), "\n")

**Question 2  
Please run set.seed(6015) before running the analysis below.  
Implement a Monte Carlo simulation of M = 1, 000 random samples of N = 100 realizations of the Gamma distribution G(a, b) with shape a = 3 and rate b = 2. Calculate and store the M corresponding sample means.  
(a) Quote the Monte Carlo estimate of the expected value of the sample mean for this probability model.  
(b) Is this value of the sample mean surprising? Briefly explain why (or why not).  
(c) Quote the Monte Carlo estimate of the standard error of the sample mean for this probability model.**

**Question 2: Monte Carlo Simulation of Gamma Distribution**

We will simulate random samples from a Gamma distribution and calculate the sample means.

# Set seed for reproducibility

set.seed(6015)

# Parameters for the Gamma distribution

a <- 3 # shape

b <- 2 # rate

# Number of samples and repetitions

N <- 100

M <- 1000

# Initialize a vector to store sample means

sample\_means <- numeric(M)

# Monte Carlo simulation

for (i in 1:M) {

# Generate random sample from the Gamma distribution

sample <- rgamma(N, shape = a, rate = b)

# Calculate and store the sample mean

sample\_means[i] <- mean(sample)

}

# (a) Estimate the expected value of the sample mean

expected\_value <- mean(sample\_means)

cat("Monte Carlo estimate of the expected value of the sample mean:", round(expected\_value, 4), "\n")

# (b) Check if this value is surprising

if (expected\_value > 1.5 \* (a / b) || expected\_value < 0.5 \* (a / b)) {

surprising <- TRUE

} else {

surprising <- FALSE

}

cat("Is this value of the sample mean surprising?", surprising, "\n")

# (c) Estimate the standard error of the sample mean

standard\_error <- sd(sample\_means)

cat("Monte Carlo estimate of the standard error of the sample mean:", round(standard\_error, 4), "\n")

**Question 3  
Please run set.seed(4060) before running the analysis below.  
Consider R’s dataset trees of 31 felled black cherry trees. Generate 100 bootstrap estimates of the slope estimate in the linear regression of tree height (Height, in feet) with respect to tree diameter (Girth, in inches), i.e. to explain tree height in function of tree girth.  
(a) Provide a boxplot of the sampling distribution of bootstrap estimates of the parameter of interest.  
(b) Quote the bootstrap estimate of the expected value of the least squares estimator of the regression slope parameter.  
(c) Quote the bootstrap estimate for the standard error associated with the least squares estimator of the regression slope parameter.  
(d) Provide an empirical bootstrap confidence interval for the true value of the regression slope parameter**.

**Question 3: Bootstrap Estimates of Linear Regression Slope**

We will use the trees dataset to perform linear regression and generate bootstrap estimates of the slope.

# Set seed for reproducibility

set.seed(4060)

# Load the trees dataset

data(trees)

# Number of bootstrap samples

B <- 100

# Initialize a vector to store bootstrap slope estimates

boot\_slope\_estimates <- numeric(B)

# Monte Carlo bootstrap

for (i in 1:B) {

# Generate a bootstrap sample

sample\_indices <- sample(1:nrow(trees), replace = TRUE)

boot\_sample <- trees[sample\_indices, ]

# Fit linear regression model to the bootstrap sample

boot\_model <- lm(Height ~ Girth, data = boot\_sample)

# Store the slope estimate

boot\_slope\_estimates[i] <- coef(boot\_model)[2] # Slope corresponds to coef[2]

}

# (a) Boxplot of bootstrap estimates

boxplot(boot\_slope\_estimates, main = "Bootstrap Estimates of Slope",

ylab = "Slope Estimate", col = "lightblue")

# (b) Bootstrap estimate of expected value of the slope

boot\_slope\_expected <- mean(boot\_slope\_estimates)

cat("Bootstrap estimate of expected value of the slope:", round(boot\_slope\_expected, 4), "\n")

# (c) Bootstrap estimate for the standard error of the slope

boot\_slope\_se <- sd(boot\_slope\_estimates)

cat("Bootstrap estimate for standard error of the slope:", round(boot\_slope\_se, 4), "\n")

# (d) Empirical bootstrap confidence interval for the slope

ci <- quantile(boot\_slope\_estimates, probs = c(0.025, 0.975))

cat("Empirical bootstrap 95% confidence interval for the slope:", round(ci, 4), "\n")  
  
A white paper with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A white paper with black text

Description automatically generated

A screenshot of a paper

Description automatically generated

A white page with black text

Description automatically generated

A white text with black text

Description automatically generated

A graph of a number of numbers

Description automatically generated with medium confidence