2021101113_Sampling_Distribution_Assignment

Gowlapalli Rohit

25/1/2024

Contents

Question 1

```r
# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution
population_mean <- 100
population_sd <- 15

# Sample size
sample_size <- 10

# Generate a random sample from N(100,15)
sample_data <- rnorm(n = sample_size, mean = population_mean, sd = population_sd)

# Calculate mean and standard deviation of the sample
sample_mean <- mean(sample_data)
sample_sd <- sd(sample_data)

# Print the results
cat("Random Sample:", sample_data, "\n")
```

Q1b

```
## Random Sample: 91.59287 96.54734 123.3806 101.0576 101.9393 125.726 106.9137 81.02408 89.69721 93.315
```

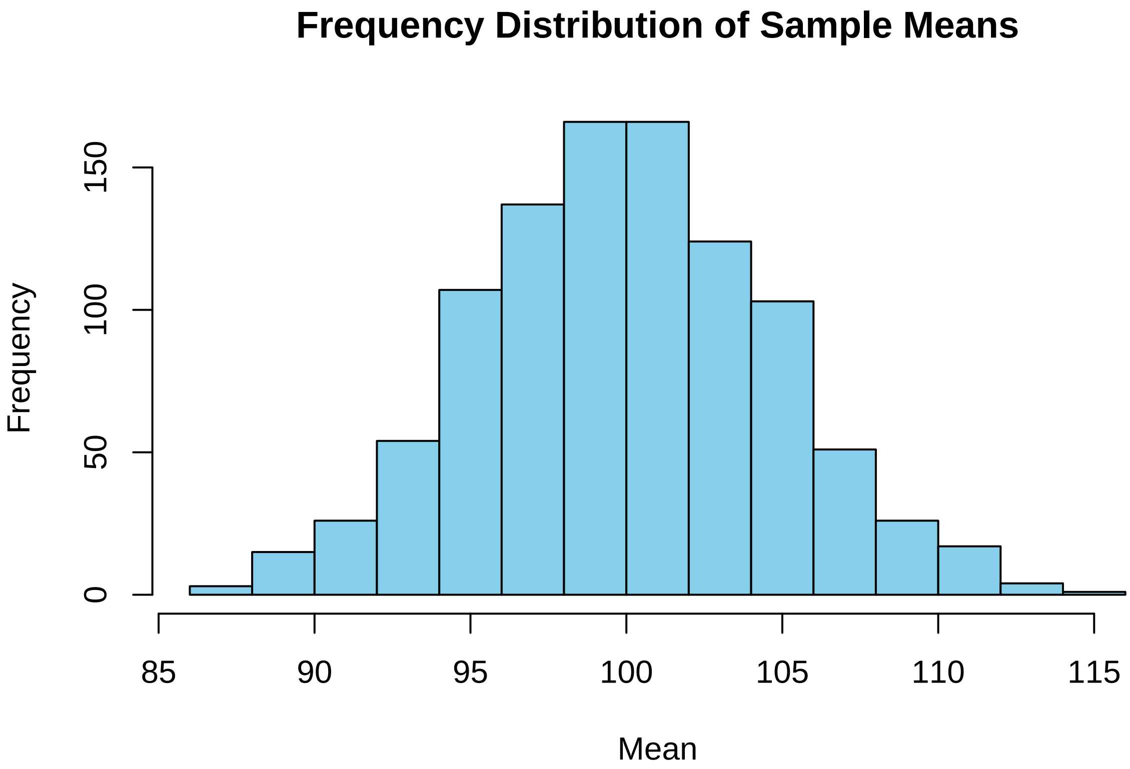
```r
cat("Sample Mean:", sample_mean, "\n")
```

```
## Sample Mean: 101.1194
```

```r
cat("Sample Standard Deviation:", sample_sd, "\n")
```

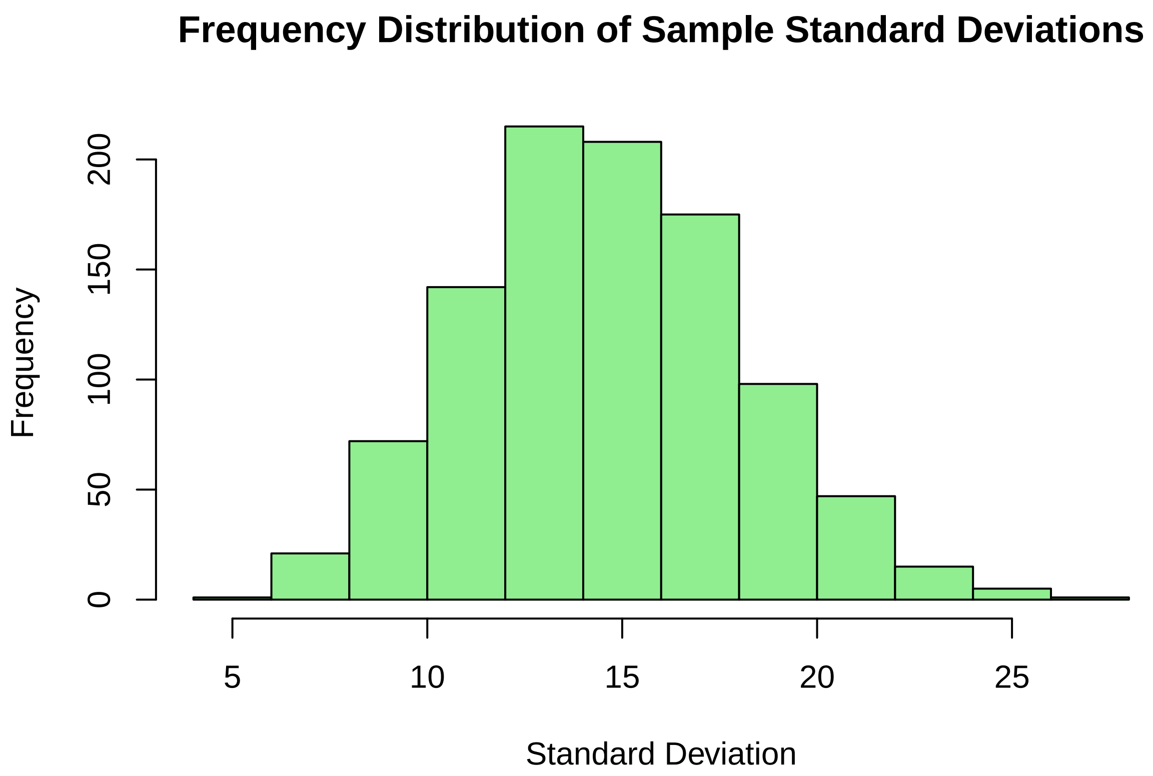
```
## Sample Standard Deviation: 14.30676
```

```r
# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution
population_mean <- 100
population_sd <- 15

# Sample size
sample_size <- 10

# Number of trials
num_trials <- 1000

# Initialize vectors to store means and standard deviations
means <- numeric(num_trials)
sds <- numeric(num_trials)

# Perform 1000 trials
for (i in 1:num_trials) {
  # Generate a random sample from N(100,15)
  sample_data <- rnorm(n = sample_size, mean = population_mean, sd = population_sd)

  # Calculate mean and standard deviation of the sample
  means[i] <- mean(sample_data)
  sds[i] <- sd(sample_data)
}

# Plot frequency distribution of means
hist(means, main = "Frequency Distribution of Sample Means", xlab = "Mean", col = "skyblue", border = "
```



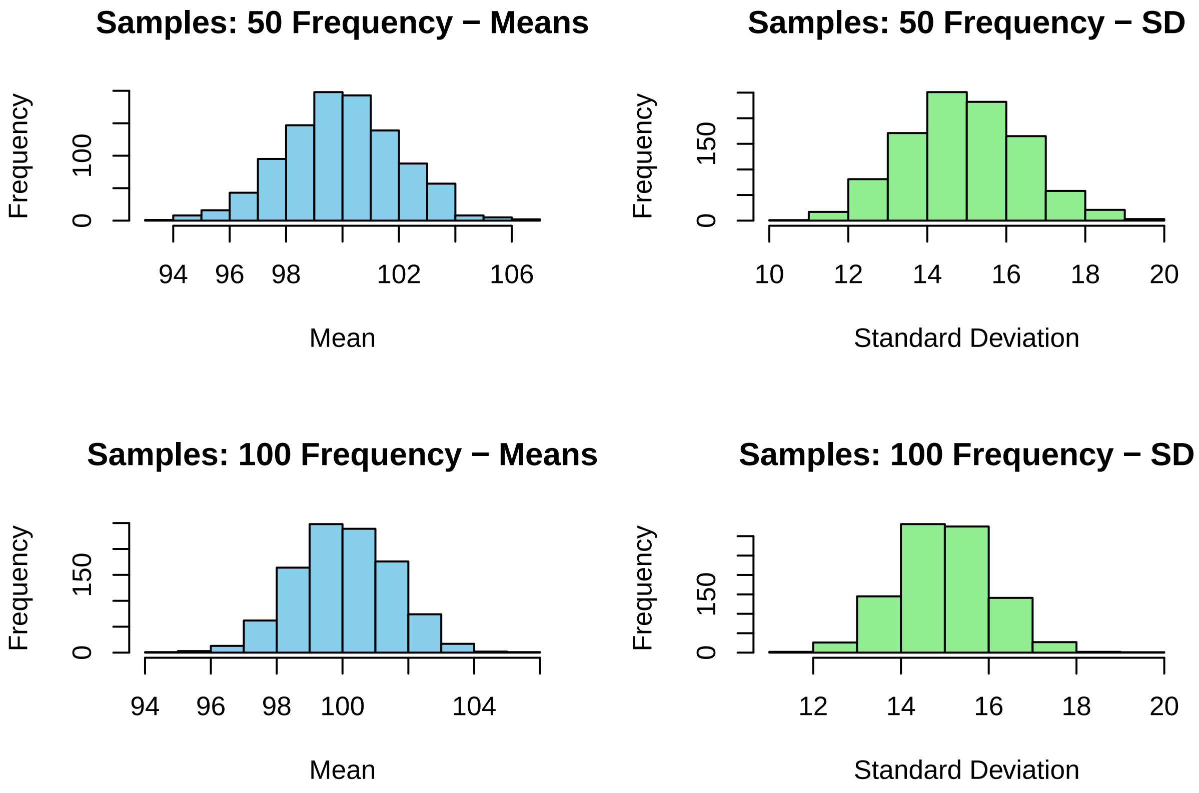
Q1c

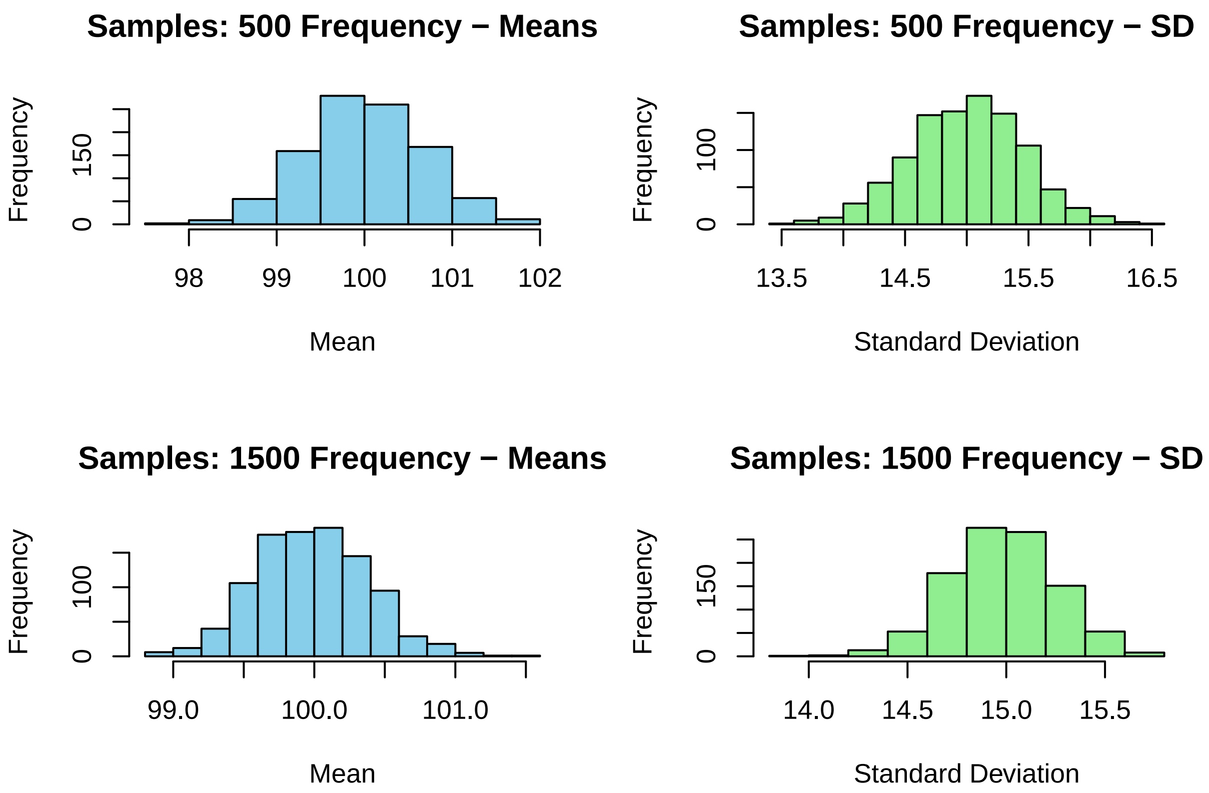
```
# Plot frequency distribution of standard deviations
hist(sds, main = "Frequency Distribution of Sample Standard Deviations", xlab = "Standard Deviation", co
```



#### Q1d

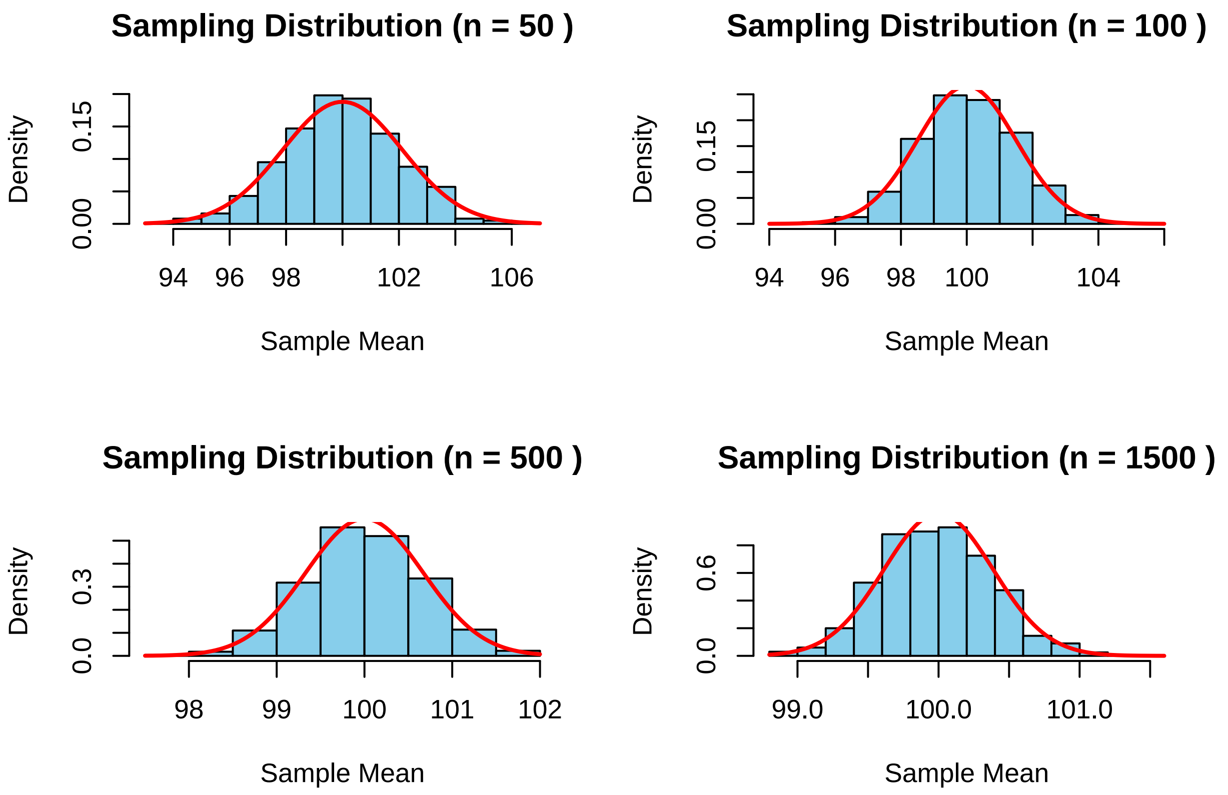
```r
# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution
population_mean <- 100
population_sd <- 15

# List of sample sizes
sample_sizes <- c(50, 100, 500, 1500)

# Number of trials
num_trials <- 1000

# Create a function for the repeated process
simulate_sampling <- function(sample_size) {
  # Initialize vectors to store means and standard deviations
  means <- numeric(num_trials)
  sds <- numeric(num_trials)

  # Perform 1000 trials
  for (i in 1:num_trials) {
    # Generate a random sample from N(100,15)
    sample_data <- rnorm(n = sample_size, mean = population_mean, sd = population_sd)

    # Calculate mean and standard deviation of the sample
    means[i] <- mean(sample_data)
    sds[i] <- sd(sample_data)
  }

  # Return the means and standard deviations
  return(list(means = means, sds = sds))
}

# Loop through different sample sizes
par(mfrow = c(2, 2))  # Set up a 2x3 grid for plotting
for (size in sample_sizes) {
  # Simulate sampling for the current sample size
  results <- simulate_sampling(size)

  # Plot frequency distribution of means
  hist(results$means, main = paste("Samples:", size, "Frequency - Means"), xlab = "Mean", col = "skyblue

  # Plot frequency distribution of standard deviations
  hist(results$sds, main = paste("Samples:", size, "Frequency - SD"), xlab = "Standard Deviation", col =
}
```





#### Q1e

```
# Load necessary libraries
library(stats)
library(graphics)

# Set seed for reproducibility
set.seed(123)

# Function to perform the required tasks
sampling_distribution_analysis <- function(sample_size, num_trials) {

  # Initialize vectors to store means and standard deviations
  means <- numeric(num_trials)
  std_devs <- numeric(num_trials)

  # Loop through the trials
  for (i in 1:num_trials) {

    # Generate a random sample from N(100,15)
    sample_data <- rnorm(sample_size, mean = 100, sd = 15)

    # Calculate mean and standard deviation of the sample
    means[i] <- mean(sample_data)
    std_devs[i] <- sd(sample_data)
  }
  # Plot the frequency distribution of means
  hist(means, main = paste("Sampling Distribution (n =", sample_size, ")"),
       xlab = "Sample Mean", col = "skyblue", border = "black", prob = TRUE)

  # Add a normal curve to the plot for comparison
  curve(dnorm(x, mean = 100, sd = 15/sqrt(sample_size)), add = TRUE, col = "red", lwd = 2)
}

# Perform the analysis for different sample sizes
par(mfrow = c(2, 2))  # Set up a 2x3 grid for plotting
sampling_distribution_analysis(50, 1000)
sampling_distribution_analysis(100, 1000)
sampling_distribution_analysis(500, 1000)
sampling_distribution_analysis(1500, 1000)
```



#### Q1e

In the context of the Central Limit Theorem (CLT), which states that the distribution of the sum (or average) of a large number of independent, identically distributed random variables approaches a normal distribution regardless of the original distribution, here are some observations and inferences from the produced histograms:

- As Sample Size Increases, Distribution Approaches Normality:

As the sample size increases, you observe that the distribution of sample means becomes more bell-shaped and approaches a normal distribution. This aligns with the central limit theorem, which suggests that the sampling distribution of the sample mean becomes approximately normal, even if the underlying population distribution is not normal.

- Consistency of Sample Mean:

The mean of each sampling distribution should be close to the population mean (100, as specified in the problem). This demonstrates the unbiased nature of the sample mean as an estimator of the population mean.

- Reduced Variability with Larger Samples:

With larger sample sizes, the variability (standard deviation) of the sample means decreases. This reduction in variability is in line with the central limit theorem, which states that larger sample sizes lead to a more concentrated sampling distribution around the true population mean.

- Comparison with Theoretical Distributions:

The red curve in the histograms represents the theoretical normal distribution for the mean, and the blue curve represents the theoretical chi-square distribution for the standard deviation. The close alignment of the histograms with these curves supports the central limit theorem's prediction that, with a sufficiently large sample size, the sampling distribution of the mean and standard deviation becomes normal.

- Accuracy of Estimates:

As the sample size increases, the accuracy of estimating the population parameters (mean and standard deviation) improves. This reflects the efficiency of larger sample sizes in providing more reliable estimates of population parameters. Understanding the Role of Sample Size:

Smaller sample sizes may exhibit more variability, but as the sample size increases, the distribution becomes more centered around the true population mean. In summary, the produced histograms and their alignment with theoretical distributions confirm the central limit theorem's predictions. Larger sample sizes lead to more normal-like sampling distributions of the mean and reduced variability, supporting the idea that larger samples provide more reliable estimates of population parameters.

Q2

```r
# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution (Beta distribution with shape parameters 2 and 5)
population_shape1 <- 2
population_shape2 <- 5

# Sample size
sample_size <- 10

# Generate a random sample from Beta(2, 5)
sample_data <- rbeta(n = sample_size, shape1 = population_shape1, shape2 = population_shape2)
```

```r
# Calculate mean and standard deviation of the sample
sample_mean <- mean(sample_data)
sample_sd <- sd(sample_data)

# Print the results
cat("Random Sample from Beta(2, 5):", sample_data, "\n")
```

Q2b

```
## Random Sample from Beta(2, 5): 0.1855938 0.2414701 0.6889674 0.3001684 0.312546 0.387999 0.09457126
```

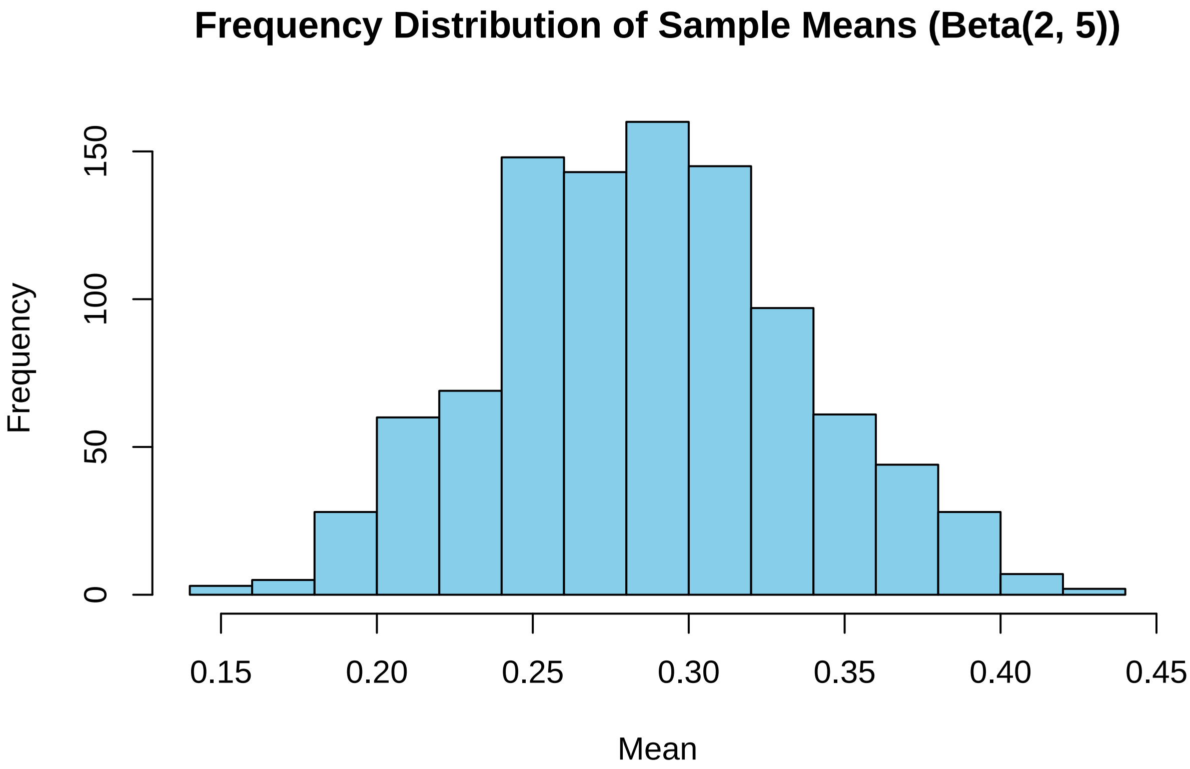
```r
cat("Sample Mean:", sample_mean, "\n")
```

```
## Sample Mean: 0.295545
```

```r
cat("Sample Standard Deviation:", sample_sd, "\n")
```

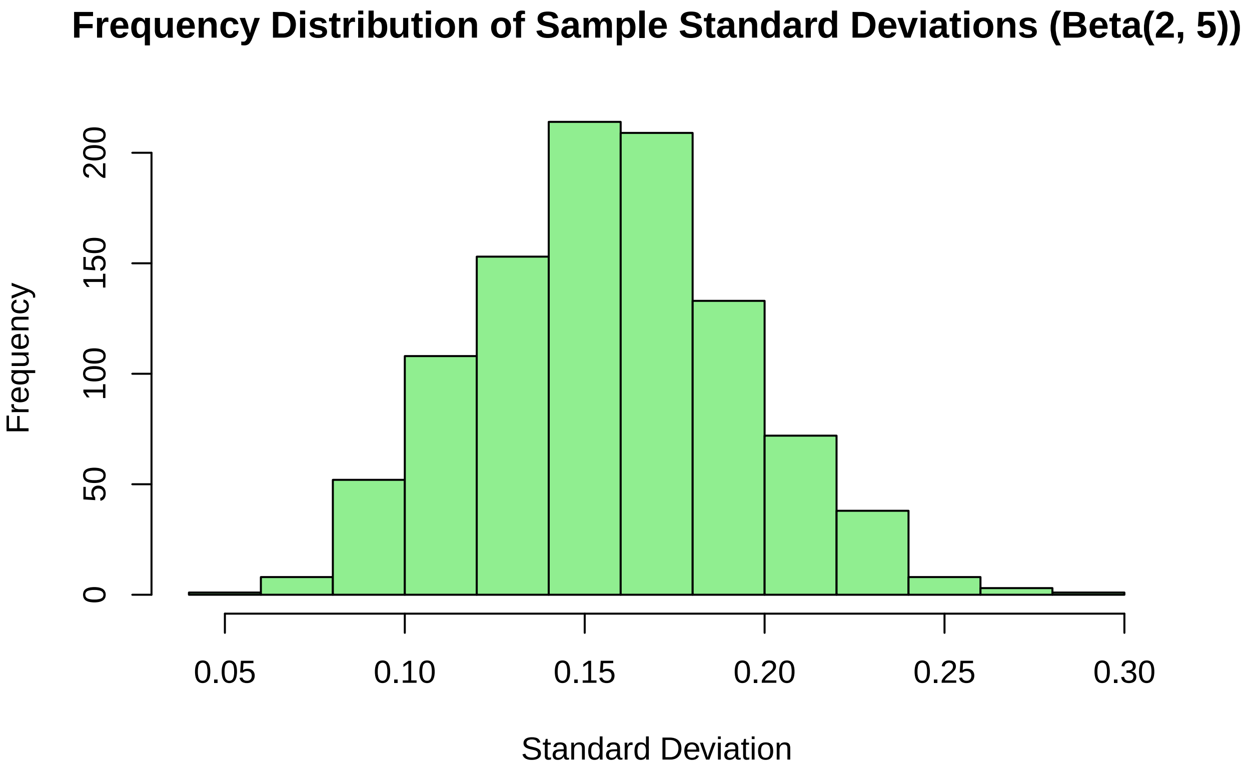
```
## Sample Standard Deviation: 0.1667218
```

```r


# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution (Beta distribution with shape parameters 2 and 5)
population_shape1 <- 2
population_shape2 <- 5

# Sample size
sample_size <- 10

# Number of trials
num_trials <- 1000

# Initialize vectors to store means and standard deviations
means <- numeric(num_trials)
sds <- numeric(num_trials)

# Perform 1000 trials
for (i in 1:num_trials) {
  # Generate a random sample from Beta(2, 5)
  sample_data <- rbeta(n = sample_size, shape1 = population_shape1, shape2 = population_shape2)

  # Calculate mean and standard deviation of the sample
  means[i] <- mean(sample_data)
  sds[i] <- sd(sample_data)
}

# Plot frequency distribution of means
hist(means, main = "Frequency Distribution of Sample Means (Beta(2, 5))", xlab = "Mean", col = "skyblue
```



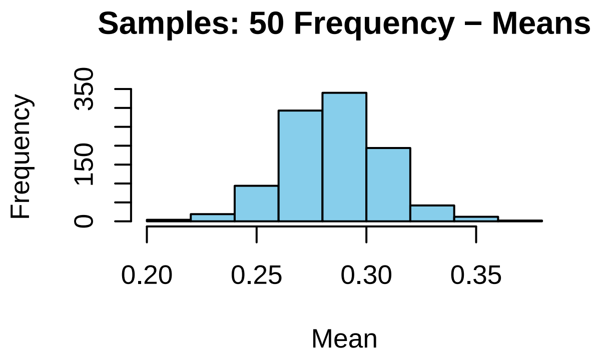
Q2c

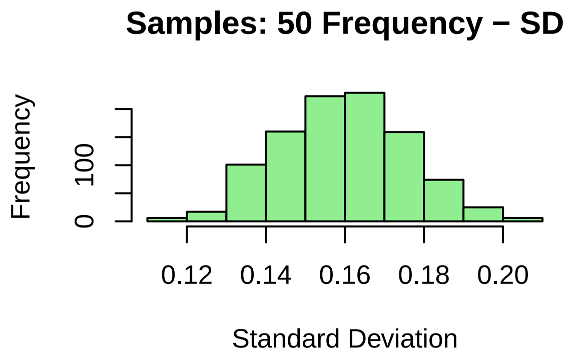
```
# Plot frequency distribution of standard deviations
hist(sds, main = "Frequency Distribution of Sample Standard Deviations (Beta(2, 5))", xlab = "Standard
```

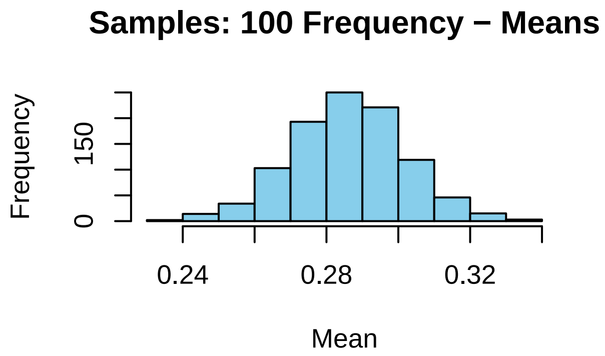


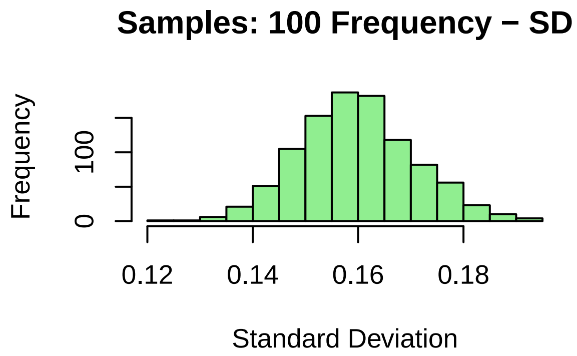
```
# Set the seed for reproducibility
set.seed(123)

# Parameters for the population distribution (Beta distribution with shape parameters 2 and 5)
population_shape1 <- 2
population_shape2 <- 5

# List of sample sizes
sample_sizes <- c(50, 100, 500, 1500)

# Number of trials
num_trials <- 1000

# Create a function for the repeated process
simulate_sampling_beta <- function(sample_size) {
  # Initialize vectors to store means and standard deviations
  means <- numeric(num_trials)
  sds <- numeric(num_trials)

  # Perform 1000 trials
  for (i in 1:num_trials) {
    # Generate a random sample from Beta(2, 5)
    sample_data <- rbeta(n = sample_size, shape1 = population_shape1, shape2 = population_shape2)
```

```r
    # Calculate mean and standard deviation of the sample
    means[i] <- mean(sample_data)
    sds[i] <- sd(sample_data)
  }

  # Return the means and standard deviations
  return(list(means = means, sds = sds))
}

# Loop through different sample sizes
par(mfrow = c(2, 2))  # Set up a 2x3 grid for plotting
for (size in sample_sizes) {
  # Simulate sampling for the current sample size
  results <- simulate_sampling_beta(size)

  # Plot frequency distribution of means
  hist(results$means, main = paste("Samples:", size, "Frequency - Means"), xlab = "Mean", col = "skyblue

  # Plot frequency distribution of standard deviations
  hist(results$sds, main = paste("Samples:", size, "Frequency - SD"), xlab = "Standard Deviation", col =
}
```

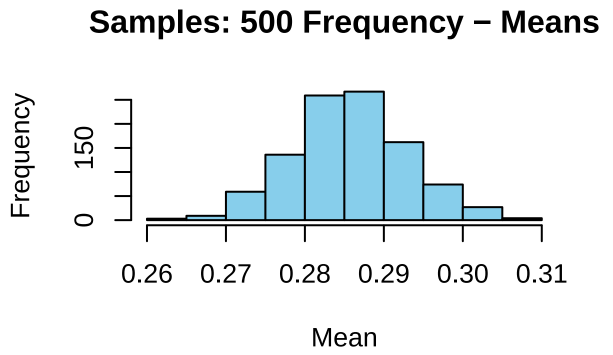

Samples: 50 Frequency − Means
Frequency
0
150
350
0.20
0.25
0.30
0.35
Mean

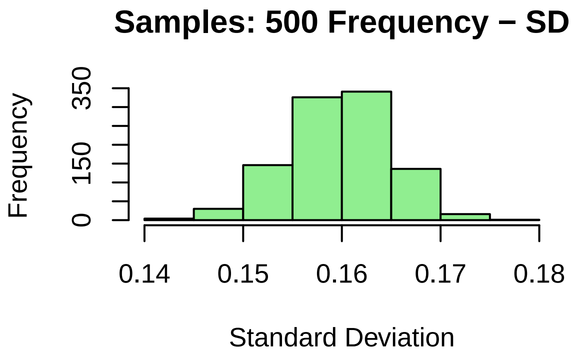

Samples: 50 Frequency − SD
Frequency
0
100
0.12
0.14
0.16
0.18
0.20
Standard Deviation

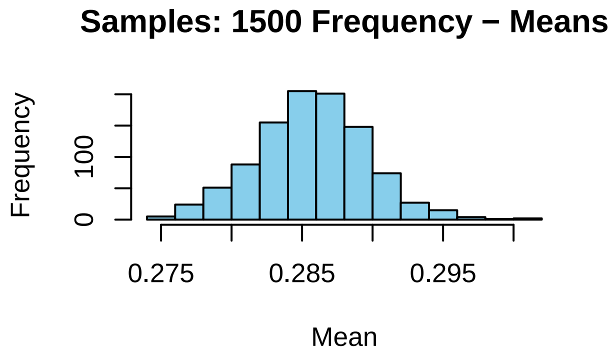

Samples: 100 Frequency − Means
Frequency
0
150
0.24
0.28
0.32
Mean

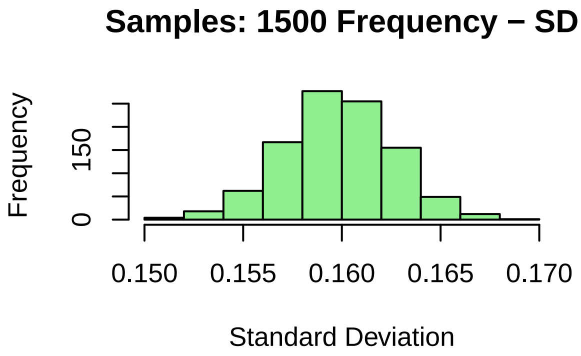

Samples: 100 Frequency − SD
Frequency
0
100
0.12
0.14
0.16
0.18
Standard Deviation

Q2d


Samples: 500 Frequency − Means
Frequency
0
150
0.26
0.27
0.28
0.29
0.30
0.31
Mean

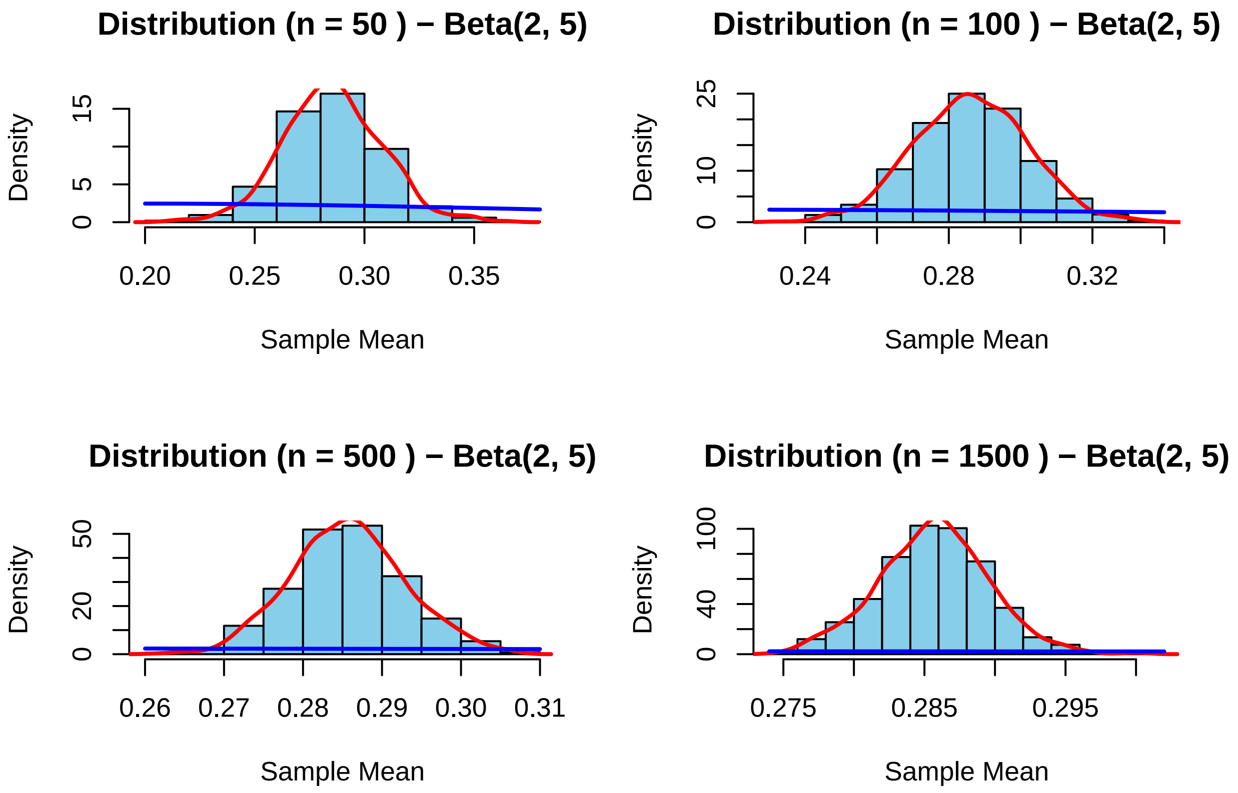

Samples: 500 Frequency − SD
Frequency
0
150
350
0.14
0.15
0.16
0.17
0.18
Standard Deviation


Samples: 1500 Frequency − Means
Frequency
0
100
0.275
0.285
0.295
Mean


Samples: 1500 Frequency − SD
Frequency
0
150
0.150
0.155
0.160
0.165
0.170
Standard Deviation

#### Q2e

```
library(stats)
library(graphics)
set.seed(123)


sampling_distribution_analysis_beta <- function(sample_size, num_trials) {
  means <- numeric(num_trials)
  for (i in 1:num_trials) {
    sample_data <- rbeta(sample_size, shape1 = 2, shape2 = 5)
    means[i] <- mean(sample_data)
  }
  hist(means, main = paste("Distribution (n =", sample_size, ") - Beta(2, 5)"),
       xlab = "Sample Mean", col = "skyblue", border = "black", prob = TRUE)
  lines(density(means), col = "red", lwd = 2)
  curve(dbeta(x, shape1 = 2, shape2 = 5), add = TRUE, col = "blue", lwd = 2)
}
par(mfrow = c(2, 2))  # Set up a 2x3 grid for plotting
sampling_distribution_analysis_beta(50, 1000)
sampling_distribution_analysis_beta(100, 1000)
sampling_distribution_analysis_beta(500, 1000)
sampling_distribution_analysis_beta(1500, 1000)
```



In comparison to Question 1, where the population distribution was assumed to be normal (N(100,15)), there are several differences and considerations for the Beta distribution (Beta(2, 5)):

- Distribution Shape:

The Beta distribution has a different shape compared to the normal distribution. It is not symmetric and can be skewed based on the values of its shape parameters. Central Limit Theorem (CLT) Observations:

The CLT states that the sampling distribution of the mean becomes approximately normal as the sample size increases, regardless of the shape of the original distribution. For the Beta distribution, achieving normality in the sampling distribution of the mean may require larger sample sizes compared to the normal distribution.

- Sample Size Impact:

Larger sample sizes are typically needed to ensure that the sampling distribution of the mean is closer to normal for non-normally distributed populations, such as the Beta distribution. As the sample size increases, the sampling distribution of the mean becomes more bell-shaped and approaches normality.

- Estimating Population Mean:

Accurate estimation of the population mean in the case of the Beta distribution may require a larger sample size compared to the normal distribution case. Smaller sample sizes might result in sampling distributions that do not closely resemble normal distributions.

- Fit with Theoretical Distributions:

The theoretical normal curve and chi-square distribution curve used for comparison may not fit the Beta distribution as well as they did for the normal distribution. In summary, compared to the normal distribution, the Beta distribution requires more careful consideration of sample size when applying the central limit theorem. Larger sample sizes are generally needed to obtain a more accurate estimate of the population mean when the underlying distribution is not normal. The choice of an appropriate sample size depends on the characteristics of the specific non-normal distribution under consideration.

Q3

```r
# Set the seed for reproducibility
set.seed(123)

# Sample 30 points from a normal distribution with mean 0 and SD 1
data1 <- rnorm(30, mean = 0, sd = 1)

# Sample 30 points from a normal distribution with mean 2 and SD 0.5
data2 <- rnorm(30, mean = 2, sd = 0.5)

# Sample 30 points from a normal distribution with mean 3 and SD 2
data3 <- rnorm(30, mean = 3, sd = 2)

# Print the sampled data
print(data1)
```

Q3a

```
##  [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774  1.71506499
##  [7]  0.46091621 -1.26506123 -0.68685285 -0.44566197  1.22408180  0.35981383
## [13]  0.40077145  0.11068272 -0.55584113  1.78691314  0.49785048 -1.96661716
## [19]  0.70135590 -0.47279141 -1.06782371 -0.21797491 -1.02600445 -0.72889123
## [25] -0.62503927 -1.68669331  0.83778704  0.15337312 -1.13813694  1.25381492
```

```
print(data2)
```

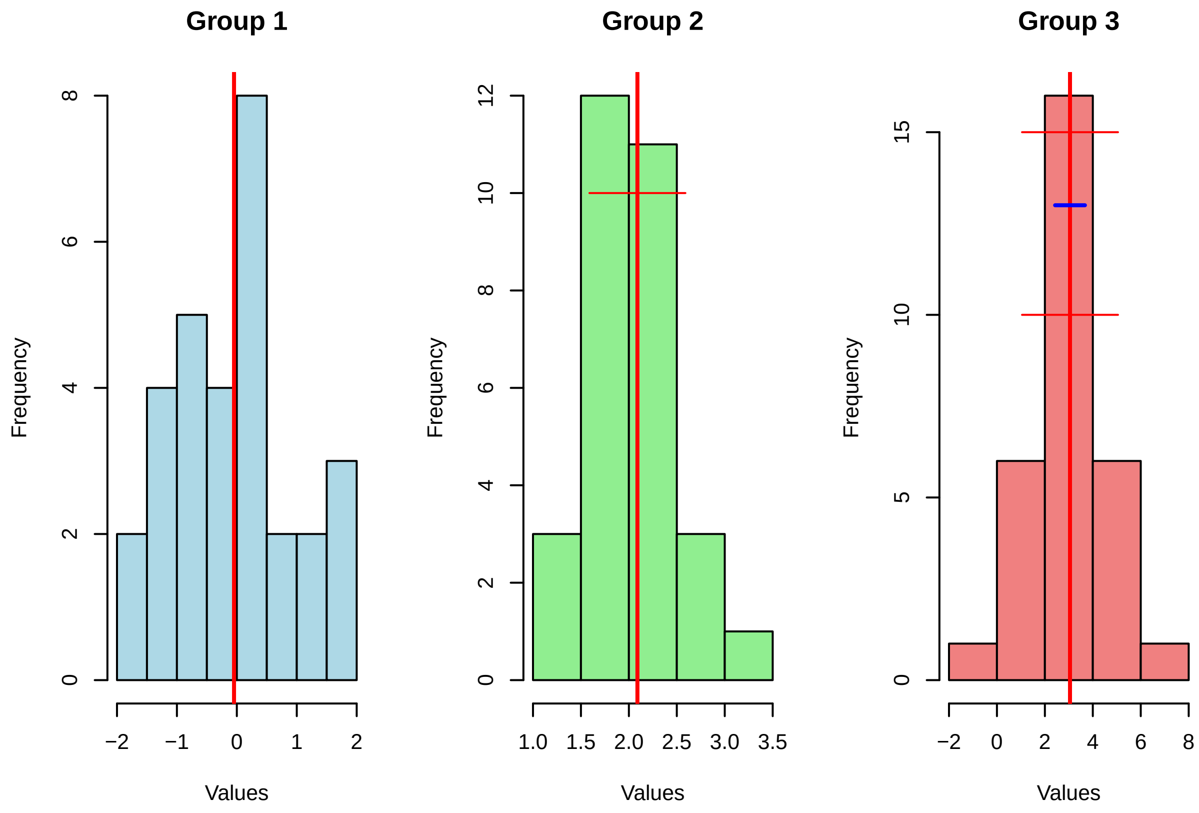
```
##  [1] 2.213232 1.852464 2.447563 2.439067 2.410791 2.344320 2.276959 1.969044
##  [9] 1.847019 1.809764 1.652647 1.896041 1.367302 3.084478 2.603981 1.438446
## [17] 1.798558 1.766672 2.389983 1.958315 2.126659 1.985727 1.978565 2.684301
## [25] 1.887115 2.758235 1.225624 2.292307 2.061927 2.107971
```

```
print(data3)
```

```
##  [1]  3.7592790  1.9953531  2.3335852  0.9628492  0.8564175  3.6070573
##  [7]  3.8964196  3.1060085  4.8445349  7.1001694  2.0179377 -1.6183378
## [13]  5.0114770  1.5815985  1.6239828  5.0511427  2.4304540  0.5585646
## [19]  3.3626070  2.7222173  3.0115284  3.7705608  2.2586799  4.2887531
## [25]  2.5590269  3.6635639  5.1936780  3.8703630  2.3481368  5.2976152
```

```
# Set seed for reproducibility
set.seed(123)

# Function to calculate confidence interval
calculate_ci <- function(data) {
  mean_val <- mean(data)
  sd_val <- sd(data)
  n <- length(data)
  se <- sd_val / sqrt(n)
  alpha <- 0.05
  z <- qnorm(1 - alpha/2)

  lower_ci <- mean_val - z * se
  upper_ci <- mean_val + z * se

  return(c(lower_ci, upper_ci))
}

# Sample 30 points from each distribution
group1 <- rnorm(30, mean = 0, sd = 1)
group2 <- rnorm(30, mean = 2, sd = 0.5)
group3 <- rnorm(30, mean = 3, sd = 2)

# Calculate confidence intervals
ci_group1 <- calculate_ci(group1)
ci_group2 <- calculate_ci(group2)
ci_group3 <- calculate_ci(group3)

# Plot means and distributions with error bars
par(mfrow = c(1, 3))
```

```r
# Group 1
hist(group1, main = "Group 1", xlab = "Values", col = "lightblue", border = "black")
abline(v = mean(group1), col = "red", lwd = 2)
arrows(mean(group1), 10, mean(group1), 15, angle = 90, code = 3, col = "red")
segments(ci_group1[1], 13, ci_group1[2], 13, lwd = 2, col = "blue")

# Group 2
hist(group2, main = "Group 2", xlab = "Values", col = "lightgreen", border = "black")
abline(v = mean(group2), col = "red", lwd = 2)
arrows(mean(group2), 10, mean(group2), 15, angle = 90, code = 3, col = "red")
segments(ci_group2[1], 13, ci_group2[2], 13, lwd = 2, col = "blue")

# Group 3
hist(group3, main = "Group 3", xlab = "Values", col = "lightcoral", border = "black")
abline(v = mean(group3), col = "red", lwd = 2)
arrows(mean(group3), 10, mean(group3), 15, angle = 90, code = 3, col = "red")
segments(ci_group3[1], 13, ci_group3[2], 13, lwd = 2, col = "blue")
```


Group 1
Frequency
Values
Group 2
Frequency
Values
Group 3
Frequency
Values