Session 2: Point Estimation and Confidence Interval

Tingfang Wang

June 2024

Contents

Point Estimation

Point estimation is a method used to estimate a population parameter based on a sample statistic. For a population percentage, we use the sample proportion, while for a population average, we use the sample mean.

Population Percentage

Suppose we want to estimate the percentage of people who prefer cats over dogs in a population. We take a random sample of 200 individuals and count how many of them prefer cats.

The point estimate for the population percentage is the sample proportion, which is calculated as:

```
set.seed(1)

# Generate a random sample of size 200 from Bernoulli distribution
# 1: prefer cats over dogs, 0: otherwise
sample_pet <- rbinom(n = 200, size = 1, prob = 0.4)

# Show how many of the sample observations are cats lovers
cat_lovers <- sum(sample_pet == 1)

# Calculate the sample proportion
sample_proportion <- cat_lovers / 200

# Print the sample proportion which is the point estimate
# for the population percentage and should be close to 0.4
sample_proportion
```

```
## [1] 0.415
```

Polulation Average

Now let's consider estimating the average income of a population. We take a random sample of 100 individuals and find their incomes.

```
set.seed(1)

# Generate a random sample of size 100 from the uniform distribution
income <- runif(n = 100, min = 25000, max = 45000)

# Show the first 5 sample observations
income[1:5]
```

```
## [1] 30310.17 32442.48 36457.07 43164.16 29033.64
```

```
# Calculate the sample mean
sample_mean <- mean(income)

# Print the sample mean which is the point estimate of population average
sample_mean
```

```
## [1] 35356.94
```

Confidence Interval for One Population

A confidence interval is a range of values that provides an estimate of the unknown population parameter with a certain level of confidence. It indicates the precision or uncertainty associated with a sample estimate.

Typically, a confidence interval is expressed as a range with an associated confidence level, such as 95% or 99%.

Confidence Interval for Population Proportion

The confidence interval for population proportion can be calculated using the following formula:

$$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

In R, we simply do the following

```
set.seed(1665)
# Generate a random sample of size 200 from Bernoulli distribution
# 1: prefer cats over dogs, 0: otherwise
pets <- rbinom(n = 200, size = 1, prob = 0.4)

# Print out the first 5 observations
pets[1:5]
```

```
## [1] 0 1 1 1 1
```

```
# How many of the sample observations are cats lovers
cat_lovers <- sum(pets == 1)

# The sample size
sample_size <- length(pets)

# Define the sample proportion
sample_proportion <- cat_lovers/sample_size

# Define the confidence level to be 95%
confidence_level <- 0.95

# Calculate the margin of error
margin_of_error <- qnorm((1 + confidence_level) / 2) *
  sqrt(sample_proportion * (1 - sample_proportion) / sample_size)

# Construct the confidence interval
confidence_interval <- c(sample_proportion - margin_of_error,
                         sample_proportion + margin_of_error)

# Print out the confidence interval
confidence_interval
```

```
## [1] 0.32725 0.46275
```

```
# Or, we can write a function to calculate the CI
CI_1Sample_P <- function(data, confidence_level){
  # How many of the sample observations are "successes"
  success_num <- sum(data == 1)
  # The sample size
```

```
  sample_size <- length(data)
  # Define the sample proportion
  sample_proportion <- success_num/sample_size
  # Calculate the margin of error
  margin_of_error <- qnorm((1 + confidence_level) / 2) *
    sqrt(sample_proportion * (1 - sample_proportion) / sample_size)
  # Construct the confidence interval
  confidence_interval <- c(sample_proportion - margin_of_error,
                           sample_proportion + margin_of_error)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
CI_1Sample_P(data = pets,
             confidence_level = 0.95)
```

```
## [1] 0.32725 0.46275
```

```
by_test <- prop.test(cat_lovers, sample_size, conf.level = 0.95, correct = FALSE)
by_test$conf.int
```

```
## [1] 0.3298410 0.4641165
## attr(,"conf.level")
## [1] 0.95
```

Therefore, the sample proportion is 0.395 and we estimate that the population percentage of cat lovers is between 32.7249996% and 46.2750004% with 95% confidence.

Confidence Interval for Population Average

To construct a confidence interval for the population average, we use the z-score when the sample size is large and t-score when the sample size is small.

When the sample size is large.

The confidence interval for population average can be calculated using the following formulas:

$$\bar{X} \pm z^* \frac{s}{\sqrt{n}}$$

Let's use a 90% confidence level to calculate the confidence interval for the previous example.

```
set.seed(46835)
# Generate a random sample of size 100 from the uniform distribution
income <- runif(n = 100, min = 25000, max = 45000)

# Show the first 5 sample observations
income[1:5]
```

```
## [1] 34006.08 28411.59 30305.89 31460.07 27845.83
```

```r
# Define a function to calculate CI
CI_1SampleZ_Avg <- function(data, confidence_level){
  # Calculate the sample mean
  sample_mean <- mean(data)
  # Calculate the sample standard deviation
  sample_sd <- sd(data)
  # The sample size
  sample_size <- length(data)
  # Calculate the margin of error
  margin_of_error <- qnorm(p = (1 + confidence_level) / 2, mean = 0, sd = 1) *
    (sample_sd / sqrt(sample_size))
  # Construct the confidence interval
  confidence_interval <- c(sample_mean - margin_of_error,
                           sample_mean + margin_of_error)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_1SampleZ_Avg(data = income,
                                       confidence_level = 0.90)
confidence_interval
```

```
## [1] 32796.34 34607.62
```

The sample average income is 3.3701983×10^4 and we estimate that the average income of the population is between 3.2796342×10^4 and 3.4607623×10^4 with 90% confidence.

When the sample size is small.

The confidence interval for population average can be calculated using the following formulas:

$$\bar{X} \pm t^*_{n-1,\alpha/2} \frac{s}{\sqrt{n}}$$

Let's use a 90% confidence level to calculate the confidence interval for the previous example but this time with a samll sample size(25).

```r
set.seed(46835)
# Generate a random sample of size 100 from the uniform distribution
income <- runif(n = 25, min = 25000, max = 45000)

# Show the first 5 sample observations
income[1:5]
```

```
## [1] 34006.08 28411.59 30305.89 31460.07 27845.83
```

```r
# Define a function to calculate CI
CI_1SampleT_Avg <- function(data, confidence_level){
  # Calculate the sample mean
  sample_mean <- mean(data)
```

```r
  # Calculate the sample standard deviation
  sample_sd <- sd(data)
  # The sample size
  sample_size <- length(data)
  # Calculate the margin of error
  margin_of_error <- qt( p = (1 + confidence_level) / 2, df = sample_size-1) *
    (sd(income) / sqrt(sample_size))
  # Construct the confidence interval
  confidence_interval <- c(sample_mean - margin_of_error,
                           sample_mean + margin_of_error)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_1SampleT_Avg(data = income,
                                       confidence_level = 0.90)
confidence_interval
```

```
## [1] 31534.49 35135.20
```

The sample average income is 3.3334847×10^4 and we estimate that the average income of the population is between 3.1534493×10^4 and 3.5135202×10^4 with 90% confidence.

Confidence Interval for Two Populations

Confidence Interval for Difference of Population Proportions

The confidence interval for the difference of population proportions is used to estimate the range of values within which the true difference between two population proportions is likely to fall. It provides a measure of uncertainty around the estimated difference.

$$(\hat{p_1} - \hat{p_2}) \pm z^* \sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}$$

Let's consider an example where we have two samples, sample1 and sample2, and we want to calculate the confidence interval for the difference in proportions between two populations.

```r
set.seed(11681)

# Generate two random samples of size 100 and 120 from Bernoulli distribution
# 1: success, 0: otherwise
sample1 <- rbinom(n = 100, size = 1, prob = 0.6)
sample1[1:5]
```

```
## [1] 0 0 1 0 0
```

```r
sample2 <- rbinom(n = 120, size = 1, prob = 0.5)
sample2[1:5]
```

```
## [1] 0 0 1 1 0
```

```
# Write a function to calculate the CI
CI_2SampleZ_P <- function(data.x, data.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
  # The sample size of data.y
  n2 <- length(data.y)
  # The number of successes in data.x
  x1 <- sum(data.x==1)
  # The number of successes in data.y
  x2 <- sum(data.y==1)
  # The sample proportion of data.x
  prop1 <- x1/n1
  # The sample proportion of data.y
  prop2 <- x2/n2
  # Calculate the standard error
  se <- sqrt((prop1 * (1 - prop1) / n1) + (prop2 * (1 - prop2) / n2))
  # Calculate the margin of error (multiply by the appropriate z-value)
  z <- qnorm(1 - (1 - confidence_level) / 2)  # Calculate the z-value
  margin_of_error <- z * se
  # Construct the confidence interval
  lower_bound <- (prop1 - prop2) - margin_of_error
  upper_bound <- (prop1 - prop2) + margin_of_error
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleZ_P(data.x = sample1,
                                     data.y = sample2,
                                     confidence_level = 0.95)

cat("The sample proportions difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample proportions difference is -0.01166667
```

```
cat("The confidence interval of the population proportions difference is",
    confidence_interval)
```

```
## The confidence interval of the population proportions difference is -0.1442906 0.1209572
```

In the above code, `prop1` and `prop2` represent the sample proportions of the two populations you are comparing, while `n1` and `n2` represent the corresponding sample sizes. The `confidence_level` variable represents the desired confidence level, such as 0.95 for a 95% confidence interval. Adjust the values accordingly in your code. The `qnorm()` function is used to calculate the z-value based on the desired confidence level.

After calculating the standard error, margin of error, and confidence interval bounds, the code displays the confidence interval for the difference of population proportions.

Confidence Interval for Difference of Population Averages

A confidence interval for the difference of population averages is a range of values within which we estimate the true difference between the means of two populations to lie, with a certain level of confidence. It provides a measure of uncertainty around the estimated difference.

When the population variances are known

To calculate the confidence interval for the difference of population averages when the population variances are known, we use the z-distribution.

$$(\bar{X}_1 - \bar{X}_2) \pm z^* \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

Here's how you can calculate the confidence interval for the difference of population means with known variances:

```r
# Sample data
sample1 <- c(12.5, 13.1, 14.2, 11.9, 12.8)
sample2 <- c(11.2, 12.1, 13.5, 11.8, 12.6)

# Known population standard deviations
sigma1 <- 1.5
sigma2 <- 1.8

# Write a function to calculate the CI
CI_2SampleZ_Avg <- function(data.x, data.y, sigma.x, sigma.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
  # The sample size of data.y
  n2 <- length(data.y)
  # Calculate the sample means
  mean1 <- mean(data.x)
  mean2 <- mean(data.y)
  # Calculate the standard error
  se <- sqrt((sigma.x^2 / n1) + (sigma.y^2 / n2))
  # Calculate the z-value for the desired confidence level
  z_value <- qnorm((1 + confidence_level) / 2)
  # Calculate the margin of error
  margin_of_error <- z_value * se
  # Calculate the lower and upper bounds of the confidence interval
  lower_bound <- (mean1 - mean2) - margin_of_error
  upper_bound <- (mean1 - mean2) + margin_of_error
  # Display the confidence interval
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleZ_Avg(data.x = sample1,
                                       data.y = sample2,
                                       sigma.x = 1.5,
```

```
                                        sigma.y = 1.8,
                                        confidence_level = 0.95)

cat("The sample means difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample means difference is 0.66
```

```
cat("The confidence interval of the population means difference is",
    confidence_interval)
```

```
## The confidence interval of the population means difference is -1.393758 2.713758
```

In the above code, `sample1` and `sample2` represent the two samples for which you want to calculate the confidence interval. The `sigma1` and `sigma2` variables represent the known standard deviations of the respective populations. The `confidence_level` variable represents the desired confidence level, such as 0.95 for a 95% confidence interval. Adjust these values accordingly in your code. The `qnorm()` function is used to calculate the z-value based on the desired confidence level.

After calculating the standard error, margin of error, and confidence interval bounds, the code displays the confidence interval for the difference of population means.

When the population variances are the same but unknown

To calculate the confidence interval for the difference of population averages when the population variances are the same but unknown, we use pooled variance and the z-distribution when the sample sizes are large. When the sample sizes are small, we use the t-distribution instead.

$$S_p = \sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}}$$

If $n_1 > 30$ and $n_2 > 30$, we can use the z-distribution:

$$(\bar{X}_1 - \bar{X}_2) \pm z^* S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

If $n_1 < 30$ or $n_2 < 30$, we can use the t-distribution:

$$(\bar{X}_1 - \bar{X}_2) \pm t^*_{(n_1+n_2-2)} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Here's how you can calculate the confidence interval for the difference of population means with unknown but equal variances and large sample sizes:

```
set.seed(2546)
# Generate two samples from normal distribution
sample1 <- rnorm(n = 100, mean = 3, sd = 1)
sample2 <- rnorm(n = 100, mean = 0, sd = 1)

# Write a function to calculate the CI
CI_2SampleZ_Avg <- function(data.x, data.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
```

```r
  # The sample size of data.y
  n2 <- length(data.y)
  # Calculate the sample means
  mean1 <- mean(data.x)
  mean2 <- mean(data.y)
  # Calculate the sample standard deviations
  sd1 <- sd(data.x)
  sd2 <- sd(data.y)
  # Calculate the pooled standard deviation
  pooled_sd <- sqrt(((n1 - 1) * sd1^2 + (n2 - 1) * sd2^2) / (n1 + n2 - 2))
  # Calculate the standard error
  se <- pooled_sd * sqrt((1 / n1) + (1 / n2))
  # Calculate the z-value for the desired confidence level
  z_value <- qnorm((1 + confidence_level) / 2)
  # Calculate the margin of error
  margin_of_error <- z_value * se
  # Calculate the lower and upper bounds of the confidence interval
  lower_bound <- (mean1 - mean2) - margin_of_error
  upper_bound <- (mean1 - mean2) + margin_of_error
  # Display the confidence interval
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleZ_Avg(data.x = sample1,
                                       data.y = sample2,
                                       confidence_level = 0.95)

cat("The sample means difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample means difference is 2.870202
```

```r
cat("The confidence interval of the population means difference is",
    confidence_interval)
```

```
## The confidence interval of the population means difference is 2.604862 3.135541
```

And, here's how you can calculate the confidence interval for the difference of population means with unknown but equal variances and small sample sizes:

```r
set.seed(13861)
# Generate two samples from normal distribution
sample1 <- rnorm(n = 25, mean = 3, sd = 1)
sample2 <- rnorm(n = 25, mean = 0, sd = 1)

# Write a function to calculate the CI
CI_2SampleT_Avg <- function(data.x, data.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
  # The sample size of data.y
```

```
  n2 <- length(data.y)
  # Calculate the sample means
  mean1 <- mean(data.x)
  mean2 <- mean(data.y)
  # Calculate the sample standard deviations
  sd1 <- sd(data.x)
  sd2 <- sd(data.y)
  # Calculate the pooled standard deviation
  pooled_sd <- sqrt(((n1 - 1) * sd1^2 + (n2 - 1) * sd2^2) / (n1 + n2 - 2))
  # Calculate the standard error
  se <- pooled_sd * sqrt((1 / n1) + (1 / n2))
  # Calculate the degrees of freedom for the t-distribution
  df <- n1 + n2 - 2
  # Calculate the t-value for the desired confidence level
  t_value <- qt((1 + confidence_level) / 2, df)
  # Calculate the margin of error
  margin_of_error <- t_value * se
  # Calculate the lower and upper bounds of the confidence interval
  lower_bound <- (mean1 - mean2) - margin_of_error
  upper_bound <- (mean1 - mean2) + margin_of_error
  # Display the confidence interval
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleT_Avg(data.x = sample1,
                                       data.y = sample2,
                                       confidence_level = 0.95)

cat("The sample means difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample means difference is 3.154277
```

```
cat("The confidence interval of the population means difference is",
    confidence_interval)
```

```
## The confidence interval of the population means difference is 2.677519 3.631035
```

In the above code, `sample1` and `sample2` represent the two samples for which you want to calculate the confidence interval. The `confidence_level` variable represents the desired confidence level, such as 0.95 for a 95% confidence interval. Adjust the sample data accordingly in your code. The `qnorm()` function is used to calculate the z-value based on the desired confidence level. The qt() function is used to calculate the t-value based on the desired confidence level and the degrees of freedom.

After calculating the pooled standard deviation, standard error, margin of error, and confidence interval bounds, the code displays the confidence interval for the difference of population means.

When population variances are different and unknown

To calculate the confidence interval for the difference of population averages when the population variances are different and unknown, we use the t-distribution the sample sizes are relatively small. When the sample sizes are large, we can use the normal distribution instead.

If $n_1 > 30$ and $n_2 > 30$, we can use the z-distribution:

$$(\bar{X}_1 - \bar{X}_2) \pm z^* \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

If $n_1 < 30$ or $n_2 < 30$, we can use the t-distribution:

$$(\bar{X}_1 - \bar{X}_2) \pm t_\nu^* \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}$$

where

$$\nu = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{S_1^4}{n_1^2(n_1-1)} + \frac{S_2^4}{n_2^2(n_2-1)}}$$

Here's how you can calculate the confidence interval for the difference of population means with unknown different variances and large sample sizes:

```r
# Generate two samples from normal distribution
sample1 <- rnorm(n = 100, mean = 3, sd = 1)
sample2 <- rnorm(n = 100, mean = 0, sd = 1)

# Write a function to calculate the CI
CI_2SampleZ_Avg <- function(data.x, data.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
  # The sample size of data.y
  n2 <- length(data.y)
  # Calculate the sample means
  mean1 <- mean(data.x)
  mean2 <- mean(data.y)
  # Calculate the sample standard deviations
  sd1 <- sd(data.x)
  sd2 <- sd(data.y)
  # Calculate the standard error
  se <- sqrt((sd1^2 / n1) + (sd2^2 / n2))
  # Calculate the z-value for the desired confidence level
  z_value <- qnorm((1 + confidence_level) / 2)
  # Calculate the margin of error
  margin_of_error <- z_value * se
  # Calculate the lower and upper bounds of the confidence interval
  lower_bound <- (mean1 - mean2) - margin_of_error
  upper_bound <- (mean1 - mean2) + margin_of_error
  # Display the confidence interval
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleZ_Avg(data.x = sample1,
                                       data.y = sample2,
```

```
                                        confidence_level = 0.95)

cat("The sample means difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample means difference is 2.931505
```

```
cat("The confidence interval of the population means difference is",
    confidence_interval)
```

```
## The confidence interval of the population means difference is 2.672974 3.190037
```

Here's how you can calculate the confidence interval for the difference of population means with unknown different variances and small sample sizes:

```
set.seed(57424)
# Generate two samples from normal distribution
sample1 <- rnorm(n = 25, mean = 3, sd = 1)
sample2 <- rnorm(n = 25, mean = 0, sd = 1)

# Write a function to calculate the CI
CI_2SampleT_Avg <- function(data.x, data.y, confidence_level){
  # The sample size of data.x
  n1 <- length(data.x)
  # The sample size of data.y
  n2 <- length(data.y)
  # Calculate the sample means
  mean1 <- mean(data.x)
  mean2 <- mean(data.y)
  # Calculate the sample standard deviations
  sd1 <- sd(data.x)
  sd2 <- sd(data.y)
  # Calculate the degrees of freedom
  df <- ((sd1^2 / n1 + sd2^2 / n2)^2) /
    ((sd1^2 / n1)^2 / (n1 - 1) + (sd2^2 / n2)^2 / (n2 - 1))
  # Calculate the standard error
  se <- sqrt(sd1^2 / n1 + sd2^2 / n2)
  # Calculate the t-value for the desired confidence level and degrees of freedom
  t_value <- qt((1 + confidence_level) / 2, df)
  # Calculate the margin of error
  margin_of_error <- t_value * se
  # Calculate the lower and upper bounds of the confidence interval
  lower_bound <- (mean1 - mean2) - margin_of_error
  upper_bound <- (mean1 - mean2) + margin_of_error
  # Display the confidence interval
  confidence_interval <- c(lower_bound, upper_bound)
  # Return the CI
  return(confidence_interval)
}

# Calculate the CI by calling the function
confidence_interval <- CI_2SampleT_Avg(data.x = sample1,
                                       data.y = sample2,
```

```
                                        confidence_level = 0.95)

cat("The sample means difference is", mean(sample1) - mean(sample2), "\n")
```

```
## The sample means difference is 2.956444
```

```
cat("The confidence interval of the population means difference is",
    confidence_interval)
```

```
## The confidence interval of the population means difference is 2.361496 3.551392
```

In the above code, `sample1` and `sample2` represent the two samples for which you want to calculate the confidence interval. The `confidence_level` variable represents the desired confidence level, such as 0.95 for a 95% confidence interval. Adjust the sample data accordingly in your code. The qnorm() function is used to calculate the z-value based on the desired confidence level. The `qt()` function is used to calculate the t-value based on the desired confidence level and the degrees of freedom.

After calculating the degrees of freedom, standard error, margin of error, and confidence interval bounds, the code displays the confidence interval for the difference of population means.

Bootstrap Confidence Interval

Bootstrap is a re-sampling technique used to estimate the sampling distribution of a statistic when the theoretical distribution is unknown or difficult to derive. It involves repeatedly sampling with replacement from the original sample to create multiple bootstrap samples. By analyzing the distribution of the bootstrap statistic, we can estimate the standard error, confidence intervals, and perform hypothesis tests.

Bootstrap CI for Population Average

Here's an example of using the bootstrap method to estimate the 95% confidence interval for a population mean:

```
set.seed(12)

# Suppose this is the data we have and we don't the distribution of this data
data <- c(3, 5, 1, 6, 4, 8, 2, 9, 7)

# Sample mean
sample_mean <- mean(data)

# Define how many bootstrap samples we want to construct
n_bootstrap <- 1000

# Generate 1000 samples by randomly take values form the data with replacement,
# then calculate the mean of those 1000 samples.
bootstrap_means <- replicate(n_bootstrap, mean(sample(data, replace = TRUE)))

# Lower bound of the bootstrap confidence interval
lower_ci <- quantile(bootstrap_means, 0.025)
```

```r
# Lower bound of the bootstrap confidence interval
upper_ci <- quantile(bootstrap_means, 0.975)

# Print out the result
cat("The sample mean is", sample_mean, '\n')
```

```
## The sample mean is 5
```

```r
cat("95% Confidence Interval is:", '[', lower_ci, upper_ci, ']')
```

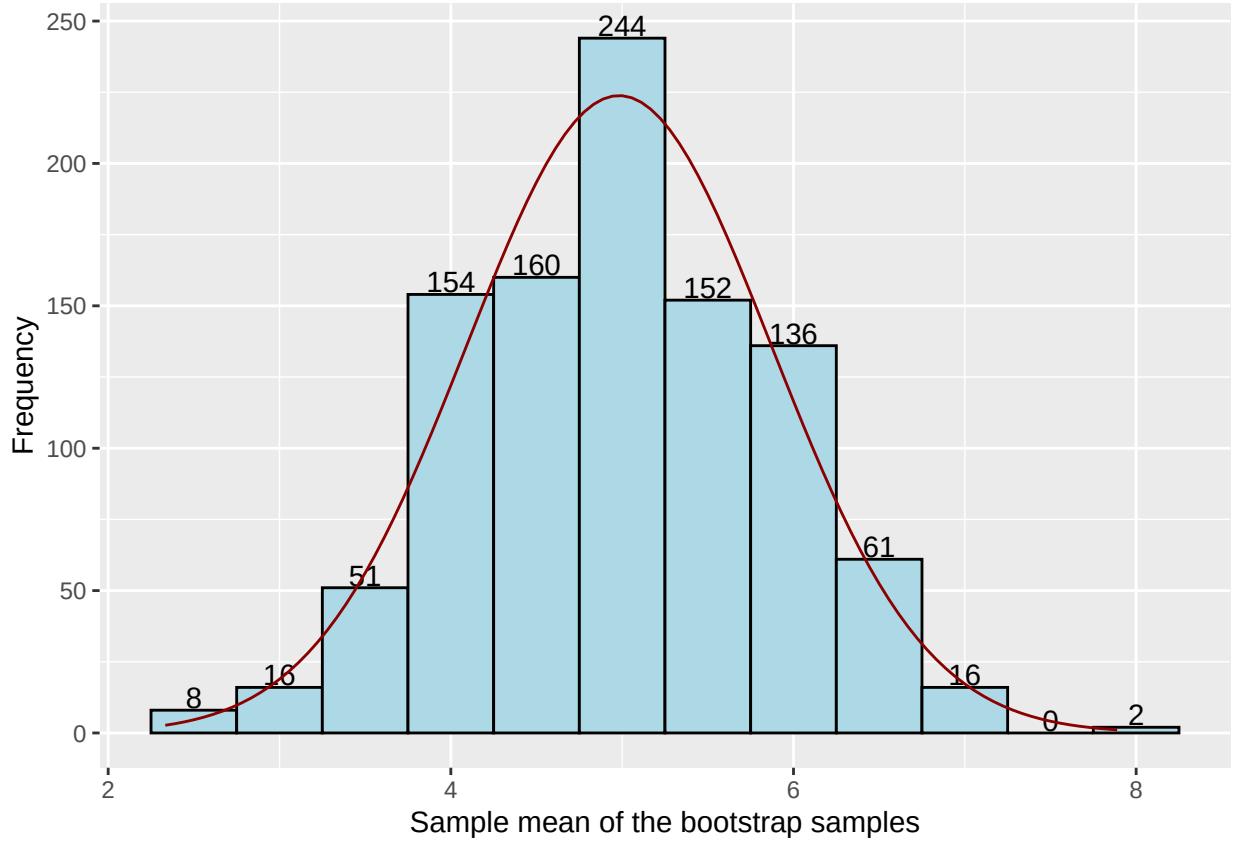
```
## 95% Confidence Interval is: [ 3.441667 6.777778 ]
```

Bootstrap CI for Population Average using `boot()` function

Another way to do this in `R` is to use the `boot()` function in `boot` package. For example

```r
set.seed(165)

# Install and load the boot package
#install.packages("boot")
library(boot)

# Suppose this is the data we have and we don't the distribution of this data
data <- c(3, 5, 1, 6, 4, 8, 2, 9, 7)

# Sample mean
sample_mean <- mean(data)

# Define a function to get the statistic of interest (mean in this example)
mean_func <- function(data, index) {
  mean(data[index])
}

# Use boot function to get the bootstrap means (1000 samples)
boot_results <- boot(data = data, statistic = mean_func, R = 1000)

# Get the generated bootstrap samples
bootstrap_samples <- boot_results$t
bootstrap_samples <- as.data.frame(bootstrap_samples)

# Create a histogram of the bootstrap sample means
histogram <- ggplot(data = bootstrap_samples,
                    aes(x = V1)) +
  geom_histogram(binwidth = 0.5,
                 fill = "lightblue",
                 color = "black") +
  # Add text
  stat_bin(geom = "text",
           aes(label = ..count..),
           vjust = -0.1,
           binwidth = 0.5) +
```

```
  labs(x = "Sample mean of the bootstrap samples",
       y = "Frequency") +
  # Add normal curve
  stat_function(fun = function(x) dnorm(x,
                                        mean = mean(bootstrap_samples$V1),
                                        sd = sd(bootstrap_samples$V1)) * 500,
                color = "darkred", linewidth = 0.5)
# Display the histogram
print(histogram)
```




```
# Use boot.ci to get the bootstrap CI
boot_conf_interval <- boot.ci(boot_results, type = "basic")

# Print out the result
cat("The sample mean is", sample_mean, '\n')
```

```
## The sample mean is 5
```

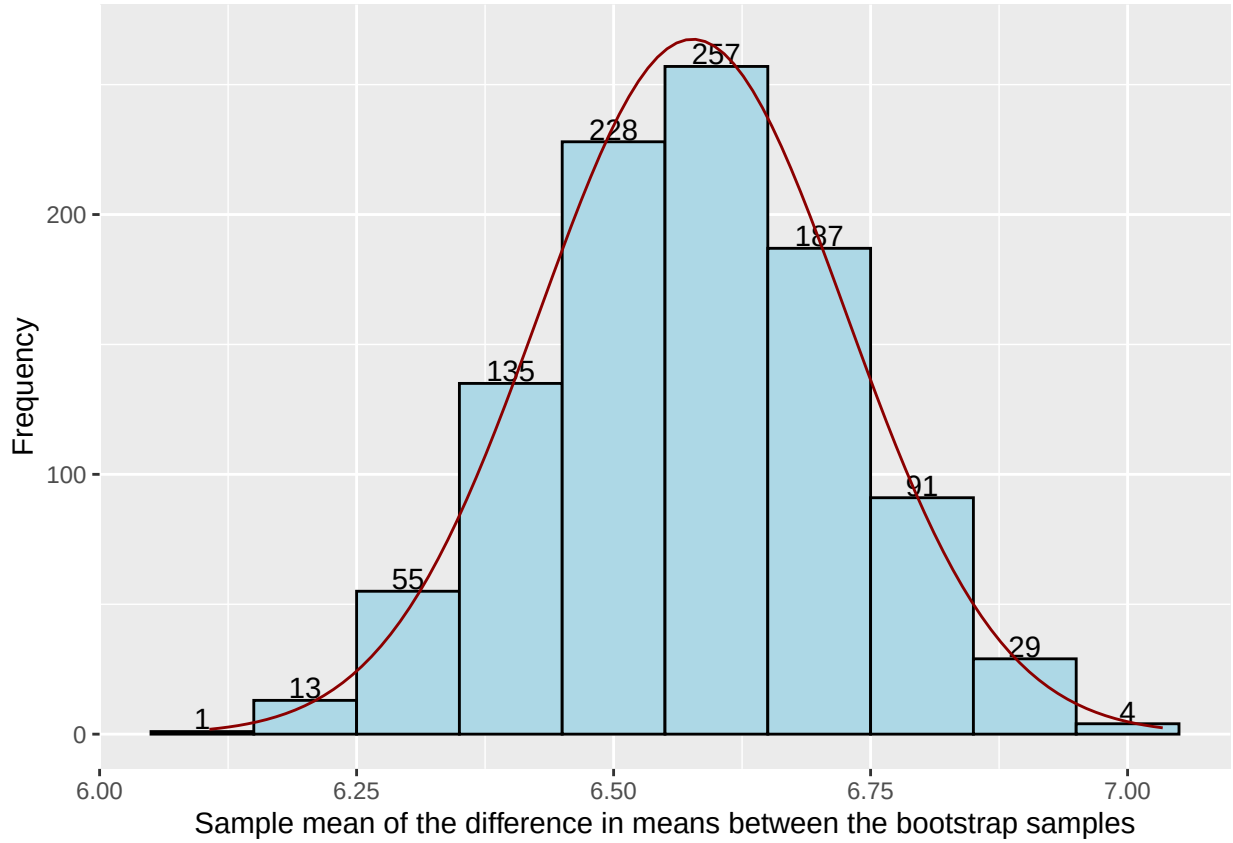
```
cat("95% Confidence Interval is", '[', boot_conf_interval$basic[4:5], ']')
```

```
## 95% Confidence Interval is [ 3.333333 6.666667 ]
```

Bootstrap CI for the Difference of Two Population Averages

To calculate a bootstrap confidence interval (CI) for the difference of the means of highway mpg and city mpg using bootstrapping in R, we can follow these steps:

```
# Load data
data(mpg)

# Create a data frame combining the two samples (highway and city mpg)
data <- as.data.frame(mpg[, c("hwy", "cty")])

# Define a function that computes the statistic of interest, which in this case
#is the difference in means between the two populations:
mean_diff <- function(data, indices) {
  mean(data[indices, "hwy"]) - mean(data[indices, "cty"])
}

# Perform the bootstrapping (1000 samples)
library(boot)
set.seed(123)  # For reproducibility
boot_result <- boot(data, mean_diff, R = 1000)

# Get the generated bootstrap samples
bootstrap_samples <- boot_result$t
bootstrap_samples <- as.data.frame(bootstrap_samples)

# Create a histogram of the bootstrap sample means
histogram <- ggplot(data = bootstrap_samples,
                    aes(x = V1)) +
  geom_histogram(binwidth = 0.1,
                 fill = "lightblue",
                 color = "black") +
  # Add text
  stat_bin(geom = "text",
           aes(label = ..count..),
           vjust = -0.1,
           binwidth = 0.1) +
  labs(x = "Sample mean of the difference in means between the bootstrap samples",
       y = "Frequency") +
  # Add normal curve
  stat_function(fun = function(x) dnorm(x,
                                        mean = mean(bootstrap_samples$V1),
                                        sd = sd(bootstrap_samples$V1)) * 100,
                color = "darkred", linewidth = 0.5)
# Display the histogram
print(histogram)
```




```r
# Obtain the bootstrap confidence interval
ci <- boot.ci(boot_result, type = "basic")

cat("The difference in sample means is", mean(data$hwy) - mean(data$cty), "\n")
```

```
## The difference in sample means is 6.581197
```

```r
# Print the confidence interval
print(ci)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_result, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   ( 6.291,  6.880 )
## Calculations and Intervals on Original Scale
```

Conclusion

In this document, we covered various statistical concepts and provided examples with R code. We discussed point estimation, confidence intervals, and bootstrap. These techniques allow us to make inferences about population parameters. Remember to choose appropriate methods based on the nature of your data and the specific research question.