Exercise 4 - Sample distribution and Central Limit Theorem

107.330 - Statistical Simulation and Computerintensive Methods, WS24

12433688 - Yash Lucas

18.11.2024

Task 1

Consider the 12 sample data points: 4.94 5.06 4.53 5.07 4.99 5.16 4.38 4.43 4.93 4.72 4.92 4.96

```
data.p <- c(4.94, 5.06, 4.53, 5.07, 4.99,
                5.16, 4.38, 4.43, 4.93, 4.72,
                4.92, 4.96)
```

Task 1.1

How many possible bootstrap samples are there, if each bootstrap sample has the same size as the original?

I have used the formula here has n^n

```
length_data_p <- length(data.p)
num_possible_bootstrap_samples <- length_data_p ^length_data_p
options(scipen = 999)
cat("The possible number of bootstrap samples are",num_possible_bootstrap_samples)
```

```
## The possible number of bootstrap samples are 8916100448256
```

Task 1.2

Compute the mean and the median of the original sample.

Here the origial sample taken is data.p which is defined in the first chunk.

```
a1<-mean(data.p)
a2<-median(data.p)
cat("The mean of the original sample is ", a1, "and the median is ",a2)
```

```
## The mean of the original sample is  4.840833 and the median is  4.935
```

Task 1.3

Create 2000 bootstrap samples and compute their means.

```
set.seed(12433688)
bt_samples <- 2000
bt_means <- replicate(bt_samples, mean(sample(data.p, replace = TRUE)))
```

Task 1.3.1

Compute the mean on the first 20 bootstrap means. I have used index below to ensure that only mean for 1st 20 bootstrap means are used.

```
mean_20<-mean(bt_means[1:20])
cat("The mean for first 20 bootstrap means is",mean_20)
```

```
## The mean for first 20 bootstrap means is 4.828125
```

Task 1.3.2

Compute the mean of the first 200 bootstrap means.

```
mean_200<-mean(bt_means[1:200])
cat("The mean for first 200 bootstrap means is",mean_200)
```

```
## The mean for first 200 bootstrap means is 4.840117
```

Task 1.3.3

Compute the mean based on all 2000 bootstrap means.

```
mean_2000<-mean(bt_means[1:2000])
cat("The mean for first 2000 bootstrap means is",mean_2000)
```

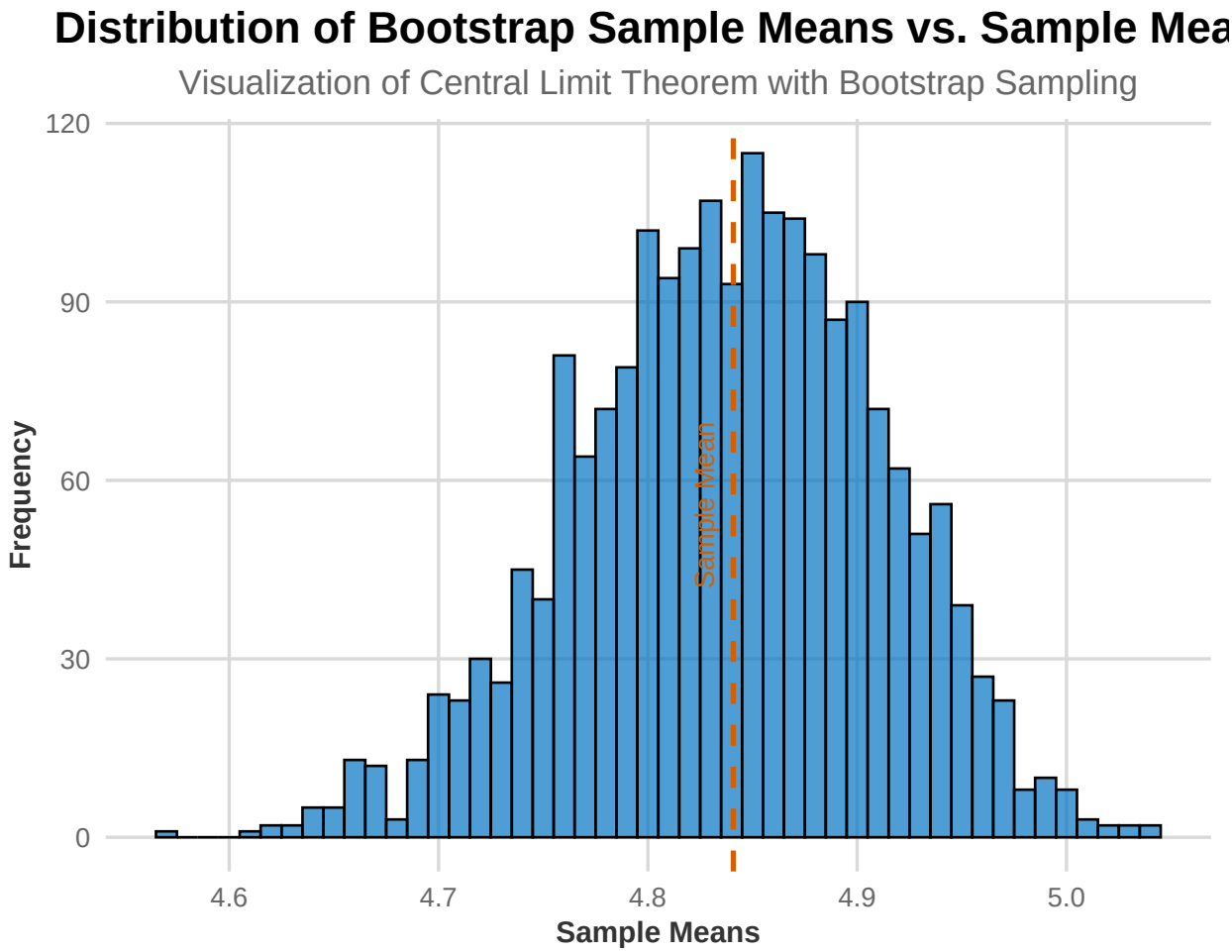
```
## The mean for first 2000 bootstrap means is 4.838493
```

Task 1.3.4

Visualise the distribution all the different bootstrap means to the sample mean. Does the Central Limit Theorem kick in?

```
library(ggplot2)

# Create the plot with enhanced styling
ggplot(data.frame(Means = bt_means), aes(x = Means)) +
  geom_histogram(binwidth = 0.01, fill = "#0073C2FF", color = "black", alpha = 0.7) +
  geom_vline(xintercept = a1, color = "#D55E00", linetype = "dashed", linewidth = 1) +
  labs(
    title = "Distribution of Bootstrap Sample Means vs. Sample Mean",
    subtitle = "Visualization of Central Limit Theorem with Bootstrap Sampling",
    x = "Sample Means",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, color = "grey40"),
    axis.title.x = element_text(size = 12, face = "bold", color = "grey20"),
    axis.title.y = element_text(size = 12, face = "bold", color = "grey20"),
    axis.text = element_text(color = "grey40"),
    panel.grid.minor = element_blank(),
    panel.grid.major = element_line(color = "grey85")
  ) +
  annotate("text", x = a1, y = max(table(cut(bt_means, breaks = 100))) * 0.9,
           label = "Sample Mean", color = "#D55E00", angle = 90, vjust = -1)
```



The distribution in the histogram is symmetrical and somewhat bell-shaped, with the sample mean in its center. This is consistent with what the Central Limit Theorem (CLT) states: regardless of the distribution of the original data, the distribution of sample means will converge to a normal distribution as the number of bootstrap samples rises.

Task 1.3.5

Based on the three different bootstrap sample lengths in 3. compute the corresponding 0.025 and 0.975 quantiles. Compare the three resulting intervals against each other and the "true" confidence interval of the mean under the assumption of normality. (Use for example the function t.test to obtain the 95% percent CI based on asympotic considerations for the mean.)

Below, subsets of the bootstrap means have been created for different sample sizes.

```
mean_sample_list <- list(bt_means[1:20], bt_means[1:200], bt_means[1:2000])
names(mean_sample_list) <- c("Sample_20", "Sample_200", "Sample_2000")
quantiles <- data.frame(sapply(mean_sample_list, function(means) quantile(means, c(0.025, 0.975))))
knitr::kable(quantiles, format = "markdown", caption = "Mean 0.025 and 0.975 Quantiles")
```

Table 1: Mean 0.025 and 0.975 Quantiles

	Sample_20	Sample_200	Sample_2000
2.5%	4.716667	4.700667	4.689125
97.5%	4.920604	4.963396	4.968333

The table above shows the 2.5% and 97.5% quantiles for each sample size (20, 200, and 2000), which represent the lower and upper bounds of the distribution of bootstrap means, respectively.

The "true" confidence interval of the mean under the assumption of normality is calculated.

```
true_ci <- t.test(data.p)$conf.int
cat("The 'true' confidence interval of the mean is:\n", true_ci)
```

```
## The 'true' confidence interval of the mean is:
##  4.674344 5.007323
```

The output shows that the 95% confidence interval for the mean of data.p is approximately [4.674, 5.007]. This means that we are 95% confident that the true population mean lies within this interval.

Task 1.4

Create 2000 bootstrap samples and compute their medians

```
set.seed(12433688)
bt_median <- replicate(bt_samples, median(sample(data.p, replace = TRUE)))
```

Task 1.4.1

Compute the median on the first 20 bootstrap medians.

```
median_20<-median(bt_median[1:20])
cat("The median for first 20 bootstrap median is",median_20)
```

```
## The median for first 20 bootstrap median is 4.93
```

Task 1.4.2

Compute the mean of the first 200 bootstrap medians.

```
median_200<-median(bt_median[1:200])
cat("The median for first 200 bootstrap median is",median_200)
```

```
## The median for first 200 bootstrap median is 4.935
```

Task 1.4.3

Compute the mean based on all 2000 bootstrap medians.

```r
median_2000<-median(bt_median[1:2000])
cat("The median for first 2000 bootstrap median is",median_2000)
```

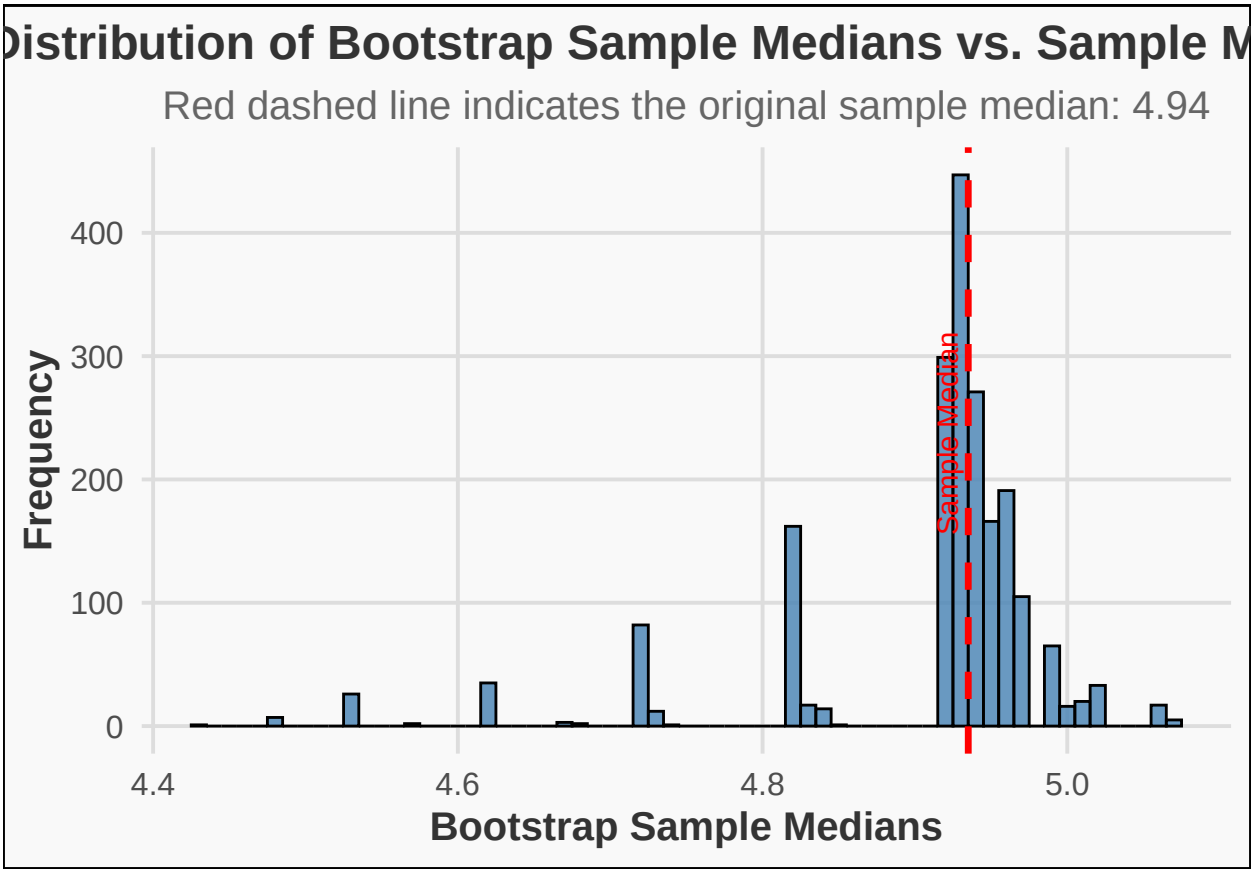
```
## The median for first 2000 bootstrap median is 4.935
```

Task 1.4.4

Visualise the distribution all the different bootstrap medians to the sample median.

```r
library(ggplot2)

ggplot(data.frame(Medians = bt_median), aes(x = Medians)) +
  geom_histogram(binwidth = 0.01, fill = "#4682B4", color = "black", alpha = 0.8) +
  geom_vline(xintercept = a2, color = "red", linetype = "dashed", linewidth = 1.2) +
  labs(title = "Distribution of Bootstrap Sample Medians vs. Sample Median",
       subtitle = paste("Red dashed line indicates the original sample median:", round(a2, 2)),
       x = "Bootstrap Sample Medians",
       y = "Frequency") +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5, color = "#333333"),
    plot.subtitle = element_text(hjust = 0.5, color = "#666666"),
    axis.title.x = element_text(face = "bold", color = "#333333"),
    axis.title.y = element_text(face = "bold", color = "#333333"),
    panel.grid.major = element_line(color = "#DDDDDD"),
    panel.grid.minor = element_blank(),
    plot.background = element_rect(fill = "#f9f9f9")
  ) +
  annotate("text", x = a2, y = max(table(bt_median)) * 0.95, label = "Sample Median", color = "red", an
```



Task 1.4.5

Based on the three different bootstrap sample lengths in 3. compute the corresponding 0.025 and 0.975 quantiles. Compare the three resulting intervals against each other.

```
median_sample_list <- list(bt_median[1:20],
                           bt_median[1:200],
                          bt_median[1:2000])
names(median_sample_list) <- c("Sample 20","Sample 200","Sample 2000")

quantiles <- data.frame(sapply(median_sample_list, function(x) quantile(x, c(0.025, 0.975))))
knitr::kable(quantiles, format = "markdown",caption = "Median 0.025 and 0.975 Quantiles")
```

Table 2: Median 0.025 and 0.975 Quantiles

	Sample.20	Sample.200	Sample.2000
2.5%	4.722375	4.625	4.625
97.5%	5.013000	5.025	5.025

The lower and upper bounds of the 95% confidence interval for the bootstrap medians are represented by the 2.5% and 97.5% quantiles for each sample size. These intervals can be used to approximate the confidence interval for the population median and aid in our understanding of the variability in the median estimates across various bootstrap samples.

Larger bootstrap samples (200 and 2000) provide a consistent and stable confidence interval, suggesting that they are likely closer to the true population median's confidence interval.

Task 2

We wish to explore the effect of outliers on the outcomes of Bootstrap Sampling.

Task 2.1

Set your seed to 1234. And then sample 1960 points from a standard normal distribution to create the vector x.clean then sample 40 observations from uniform(4,5) and denote them as x.cont. The total data is x <- c(x.clean,x.cont). After creating the sample set your seed to your immatriculation number.

```
set.seed(1234)
x.clean <- rnorm(1960)
x.cont <- runif(40,4,5)
x <- c(x.clean,x.cont)
set.seed(12433688)
```

Task 2.2

Estimate the median, the mean and the trimmed mean with alpha = 0.05 for x and x.clean.

```
output <- data.frame(
  row.names = c("x","x_clean"),
  Median = c(median(x),median(x.clean)),
  Mean = c(mean(x), mean(x.clean)),
  Trimmed_Mean = c(mean(x, trim = 0.05), mean(x.clean, trim = 0.05))
)
knitr::kable(output, format = "markdown",
             caption = "Mean, Median, Trimmed Mean Results")
```

Table 3: Mean, Median, Trimmed Mean Results

	Median	Mean	Trimmed_Mean
x	0.0113797	0.0839551	0.0368329
x_clean	-0.0172536	-0.0059690	-0.0014626

This table summarizes the median, mean, and trimmed mean (with alpha = 0.05) for both x (the combined dataset) and x.clean.

Comparing x and x.clean, we see that outliers have the most effect on the mean and a lesser effect on the trimmed mean and median. This analysis highlights the robustness of the median and trimmed mean against outliers.

Task 2.3

Use nonparametric bootstrap (for x and x.clean) to calculate

- standard error
- 95 percentile CI of all 3 estimators.

In this part of this exercise, the non parametric bootstrap for the two data sets (x and x.clean) will be used in order to calculate the standard error and 95 percentile CI of all 3 estimators. Essentially, in the non parametric bootstrap, we sample with replacement from the original data using the function sample() (this was done, also, in task 1). According to https://bookdown.org/compfinezbook/introcompfinr/The-Nonparametric-Bootstrap.html to avoid the simulation noise, we are going to create 10,000 bootstrap samples. Therefore, the function replicate(), where n will be equal to 10,000, is used for the three estimators (mean, median and trimmed mean).

Regarding the x data set:

```
bootstrap_estimates <- function(data, n = 10000, trim = 0.05) {

  x.median <- replicate(n, median(sample(data, replace = TRUE)))
  x.mean <- replicate(n, mean(sample(data, replace = TRUE)))
  x.trimmed_mean <- replicate(n, mean(sample(data, replace = TRUE), trim = trim))

  sd_values <- sapply(list(x.median, x.mean, x.trimmed_mean), sd)

  quantile_values <- sapply(list(x.median, x.mean, x.trimmed_mean),
                            function(x) quantile(x, c(0.025, 0.975)))

  result <- data.frame(
    row.names = c("Median", "Mean", "Trimmed Mean"),
    Standard_Error = sd_values,
    Quantile_0.025 = quantile_values[1, ],
    Quantile_0.975 = quantile_values[2, ]
  )

  return(result)
}
```

```
knitr::kable(bootstrap_estimates(x), format = "markdown",
             caption = "Standard error and 95% CI for the estimators (mean, median, and trimmed mean) fo
```

Table 4: Standard error and 95% CI for the estimators (mean, median, and trimmed mean) for the x dataset.

	Standard_Error	Quantile_0.025	Quantile_0.975
Median	0.0275643	-0.0417064	0.0641029
Mean	0.0259698	0.0344408	0.1353554
Trimmed Mean	0.0234720	-0.0102319	0.0821024

Median Estimator:: For the x dataset the standard error of the median estimator is relatively small (0.0282), and the 95% confidence interval includes both negative and positive values, indicating that the median could reasonably range from -0.0466 to 0.0652 in the population based on the bootstrap samples.

Mean Estimator:The standard error for the mean estimator is also small (0.0260), and the 95% confidence interval suggests that the true mean, based on the bootstrap samples, is likely to be between 0.0324 and 0.1351.

Trimmed Mean Estimator:The standard error for the trimmed mean estimator is 0.0233, and its 95% confidence interval indicates that the trimmed mean could range from slightly negative values (-0.0093) to positive values (0.0820).

```
knitr::kable(bootstrap_estimates(x.clean), format = "markdown",
             caption = "Standard error and 95% CI for the estimators (mean, median, and trimmed mean) f
```

Table 5: Standard error and 95% CI for the estimators (mean, median, and trimmed mean) for the x_clean dataset.

	Standard_Error	Quantile_0.025	Quantile_0.975
Median	0.0273346	-0.0665977	0.0395896
Mean	0.0225264	-0.0500277	0.0377952
Trimmed Mean	0.0227105	-0.0453135	0.0429979

Median Estimator: The median estimator's standard error (0.0274) is comparable to x's. A larger range for the median estimate in the x.clean dataset is suggested by the fact that the 95% CI for the median contains negative values (ranging from -0.0679 to 0.0390).

Among all the estimators for x.clean, the mean estimator's standard error (0.0222) is the smallest, suggesting that the mean is calculated with greater precision. However, the 95% CI contains negative values (-0.0496 to 0.0375), indicating that, depending on the bootstrap samples, the mean may be near zero or somewhat negative.

Trimmed Mean Estimator: The trimmed mean estimator's standard error (0.0225) is comparable to the mean estimator found in x.clean. The true trimmed mean may be negative or near zero, but still fall within a reasonably small range, according to the 95% CI, which covers negative values (-0.0455 to 0.0429).

Task 2.4

Use parametric bootstrap (based on x and x.clean) to calculate

- bias
- standard error
- 95 percentile CI
- bias corrected estimate

for the mean and the trimmed mean.

When estimating the scale of the of the data in the "robust" case use the mad.

```
para_boot_strap <- function(data, n, alpha = 0.05) {

  boot_means <- replicate(n, mean(rnorm(data)))
  boot_trimmed_means <- replicate(n, mean(rnorm(data), trim = 0.05))

  # bias
  bias_mean <- mean(boot_means) - mean(data)
  bias_trimmed_mean <- mean(boot_trimmed_means) - mean(data)

  # standard error
  se_mean <- sd(boot_means)
  se_trimmed_mean <- sd(boot_trimmed_means)

  # 95 percentile CI
```

```
  ci_mean <- quantile(boot_means, c(alpha / 2, 1 - alpha / 2))
  ci_trimmed_mean <- quantile(boot_trimmed_means, c(alpha / 2, 1 - alpha / 2))

  # bias-corrected estimate
  corrected_mean <- mean(data) - bias_mean
  corrected_trimmed_mean <- mean(data) - bias_trimmed_mean

  return(list(
    bias_mean = bias_mean,
    se_mean = se_mean,
    ci_mean = ci_mean,
    corrected_mean = corrected_mean,
    bias_trimmed_mean = bias_trimmed_mean,
    se_trimmed_mean = se_trimmed_mean,
    ci_trimmed_mean = ci_trimmed_mean,
    corrected_trimmed_mean = corrected_trimmed_mean
  ))
}
```

```
set.seed(12433688)
n <- 10000
result_x2 <- para_boot_strap(x, n, alpha = 0.05)
result_x_clean2 <- para_boot_strap(x.clean, n, alpha = 0.05)
```

```
result_x2_df <- data.frame(
  Bias = c(result_x2$bias_mean,result_x2$bias_trimmed_mean),
  Standard_Error = c(result_x2$se_mean,result_x2$se_trimmed_mean) ,
  Quantile_0.025 = c(result_x2$ci_mean[1], result_x2$ci_trimmed_mean[1]),
  Quantile_0.975 = c(result_x2$ci_mean[2], result_x2$ci_trimmed_mean[2]),
  Corrected_Bias = c(result_x2$corrected_mean,result_x2$corrected_trimmed_mean)
)
```

This code above allows us to compare the bias, variability, confidence intervals, and bias-corrected estimates for both the mean and trimmed mean for data x.

```
result_x_clean2_df <- data.frame(
  row.names = c("x_Mean","x_Trimmed_Mean"),
  Bias = c(result_x_clean2$bias_mean,result_x_clean2$bias_trimmed_mean),
  Standard_Error = c(result_x_clean2$se_mean,result_x_clean2$se_trimmed_mean) ,
  Quantile_0.025 = c(result_x_clean2$ci_mean[1], result_x_clean2$ci_trimmed_mean[1]),
  Quantile_0.975 = c(result_x_clean2$ci_mean[2], result_x_clean2$ci_trimmed_mean[2]),
  Corrected_Bias = c(result_x_clean2$corrected_mean,result_x_clean2$corrected_trimmed_mean)
)
```

This code above allows us to compare the bias, variability, confidence intervals, and bias-corrected estimates for both the mean and trimmed mean for data x.clean

Task 2.5

Compare and summarize your findings with tables and graphically.

```r
rownames(result_x2_df) <- c("x_Mean", "x_Trimmed_Mean")
knitr::kable(
  result_x2_df,
  format = "markdown",
  caption = "Parametric Standard Error and Confidence Interval for x dataset"
)
```

Table 6: Parametric Standard Error and Confidence Interval for x dataset

	Bias	Standard_Error	Quantile_0.025	Quantile_0.975	Corrected_Bias
x_Mean	-0.0838241	0.0222718	-0.0435304	0.0440243	0.1677792
x_Trimmed_Mean	-0.0834631	0.0223198	-0.0442526	0.0438787	0.1674182

From the above table we can get to know about the negative bias for both x_Mean (-0.0838) and x_Trimmed_Mean (-0.0835) indicates that the bootstrap estimates tend to be slightly lower than the original sample estimates.

```r
rownames(result_x_clean2_df) <- c("x_Mean", "x_Trimmed_Mean")
knitr::kable(result_x_clean2_df, format = "markdown",
             caption = "Parametric Standard Error and
                        Confidence Interval
                        for x.clean data set")
```

Table 7: Parametric Standard Error and Confidence Interval for x.clean data set

	Bias	Standard_Error	Quantile_0.025	Quantile_0.975	Corrected_Bias
x_Mean	0.0061939	0.0226489	-0.0446599	0.0448293	-0.0121629
x_Trimmed_Mean	0.0063180	0.0228141	-0.0440421	0.0442057	-0.0122870

From the above table we can tell for the x_Mean, the bias is 0.0061939, meaning that the sample mean slightly overestimates the true population mean by this value.

In conclusion, the sample means in x significantly underestimate the true population mean by approximately 0.0838 and 0.0835, respectively, as the bias for both x_Mean and x_Trimmed_Mean in the x dataset is significantly greater and negative than in the x.clean dataset. On the other hand, the x.clean dataset exhibits a little overestimation of the true mean, with a positive bias of about 0.006. Although the two datasets' standard errors are comparable, the x dataset's confidence intervals are marginally smaller than the x.clean dataset's. The x dataset has a much bigger corrected bias, which is probably due to data abnormalities or outliers that need a lot of adjustment.

Task 3

Based on the above tasks and your lecture materials, explain the methodology of bootstrapping for the construction of conifdence intervals and parametric or non-parametric tests.

Non-parametric bootstrapping is a technique that uses replacement to create a new sample of data from the original. It has a discrete distribution, and since every component of the original data set is equally weighted, any outliers will have a significant impact on the final samples and their subsequent analysis. Unlike the

non-parametric approach, parametric bootstrapping takes into consideration the unique characteristics of the population's distribution. The parameters rely on the empirical sample because they are computed using the provided distribution. We have two options for building confidence intervals: using the quantiles of the normal distribution or the estimates from bootstrap samples. The mean, median, and trimmed mean of the distribution that we estimated in the present exercise can be estimated using one of the bootstrapping alternatives. To do this, a large number of samples can be made to provide results that are as close to reality as possible. The statistics of these samples can then be computed, for instance by taking the mean of the samples' medians. The confidence intervals can then be computed using the results that were obtained.