Sample distribution and Central Limit Theorem

Mahtab Nahayati

2024-11-19

Contents

Bootstrap Sampling

Number of Possible Bootstrap Samples and the mean and the median of the original sample

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 The number of possible bootstrap samples, if each sample has the same size as the original, is calculated as:

$$n^n = 12^{12}$$

```
n <- 12
n_boot <- 12
boot_samples <- n^n_boot
boot_samples
```

```
## [1] 8.9161e+12
```

```
# Original sample data
data <- 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)

original_mean <- mean(data)
original_median <- median(data)
cat("Original Mean: ", original_mean, "\n")
```

```
## Original Mean:  4.840833
```

```
cat("Original Median: ", original_median, "\n")
```

```
## Original Median:  4.935
```

Genarate 2000 bootstap samples and compute their mean

```
# Number of bootstrap samples
n_bootstrap <- 2000

# Original sample data
data <- 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)

# Generate 2000 bootstrap samples and compute their means
set.seed(123)  # For reproducibility
bootstrap_means <- replicate(n_bootstrap, mean(sample(data, replace = TRUE)))

# 1. Compute the mean on the first 20 bootstrap means
mean_first_20 <- mean(bootstrap_means[1:20])
cat("Mean of the first 20 bootstrap means:", mean_first_20, "\n")
```

```
## Mean of the first 20 bootstrap means: 4.82525
```

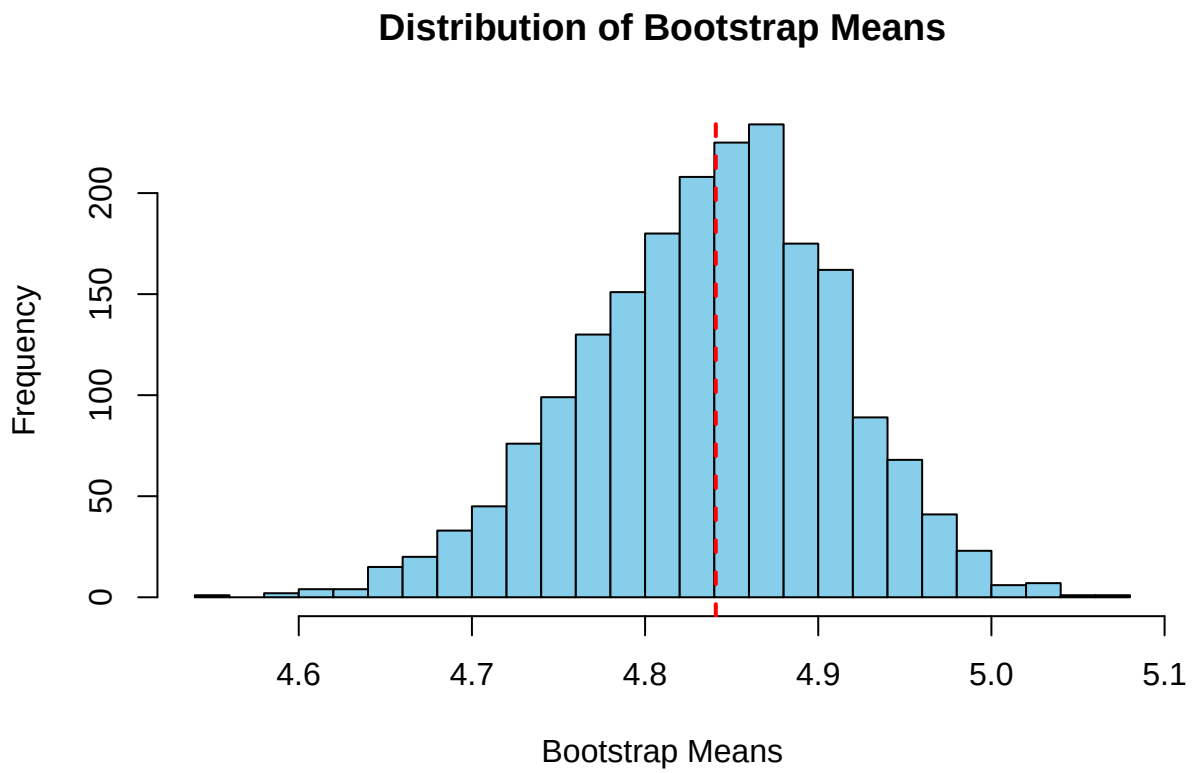
```
# 2. Compute the mean of the first 200 bootstrap means
mean_first_200 <- mean(bootstrap_means[1:200])
cat("Mean of the first 200 bootstrap means:", mean_first_200, "\n")
```

```
## Mean of the first 200 bootstrap means: 4.827187
```

```
# 3. Compute the mean based on all 2000 bootstrap means
mean_all_2000 <- mean(bootstrap_means)
cat("Mean of all 2000 bootstrap means:", mean_all_2000, "\n")
```

```
## Mean of all 2000 bootstrap means: 4.838049
```

```r
# 4. Visualise the distribution of bootstrap means
hist(
  bootstrap_means,
  breaks = 30,
  main = "Distribution of Bootstrap Means",
  xlab = "Bootstrap Means",
  col = "skyblue"
)
abline(v = mean(data), col = "red", lwd = 2, lty = 2)  # Add the sample mean
```




```r
# 5. Compute 0.025 and 0.975 quantiles for different bootstrap lengths
quantiles_20 <- quantile(bootstrap_means[1:20], c(0.025, 0.975))
quantiles_200 <- quantile(bootstrap_means[1:200], c(0.025, 0.975))
quantiles_2000 <- quantile(bootstrap_means, c(0.025, 0.975))

# Print quantiles
cat("Quantiles for first 20 bootstrap means: ", quantiles_20, "\n")
```

```
## Quantiles for first 20 bootstrap means:  4.668833 4.943833
```

```r
cat("Quantiles for first 200 bootstrap means: ", quantiles_200, "\n")
```

```
## Quantiles for first 200 bootstrap means:  4.684042 4.953437
```

```r
cat("Quantiles for all 2000 bootstrap means: ", quantiles_2000, "\n")
```

```
## Quantiles for all 2000 bootstrap means:  4.683313 4.974188
```

```r
# Compare with t-test confidence interval for the mean
t_test_ci <- t.test(data)$conf.int
cat("95% CI based on t-test:", t_test_ci, "\n")
```

```
## 95% CI based on t-test: 4.674344 5.007323
```

Results

1. Means of Bootstrap Samples

- **First 20 bootstrap means**: 4.825
- **First 200 bootstrap means**: 4.827
- **All 2000 bootstrap means**: 4.838

The bootstrap means converge as the number of samples increases, demonstrating the consistency of the bootstrapping approach.

---

2. Quantiles (95% Confidence Intervals)

Bootstrap Samples	0.025 Quantile	0.975 Quantile
First 20	4.669	4.944
First 200	4.684	4.953
All 2000	4.683	4.974

- **Theoretical CI (t-test)**: $[4.674, 5.007]$

---

3. Observations

1. The quantiles converge with increasing bootstrap sample sizes, becoming close to the theoretical CI.
2. The distribution of bootstrap means appears normal, supporting the Central Limit Theorem.

Generate 2000 bootstrap samples and compute their medians

```r
set.seed(123)  # For reproducibility
bootstrap_medians <- replicate(n_bootstrap, median(sample(data, replace = TRUE)))

# 1. Compute the mean on the first 20 bootstrap medians
mean_first_20_medians <- mean(bootstrap_medians[1:20])
cat("Mean of the first 20 bootstrap medians:", mean_first_20_medians, "\n")
```

```
## Mean of the first 20 bootstrap medians: 4.88225
```

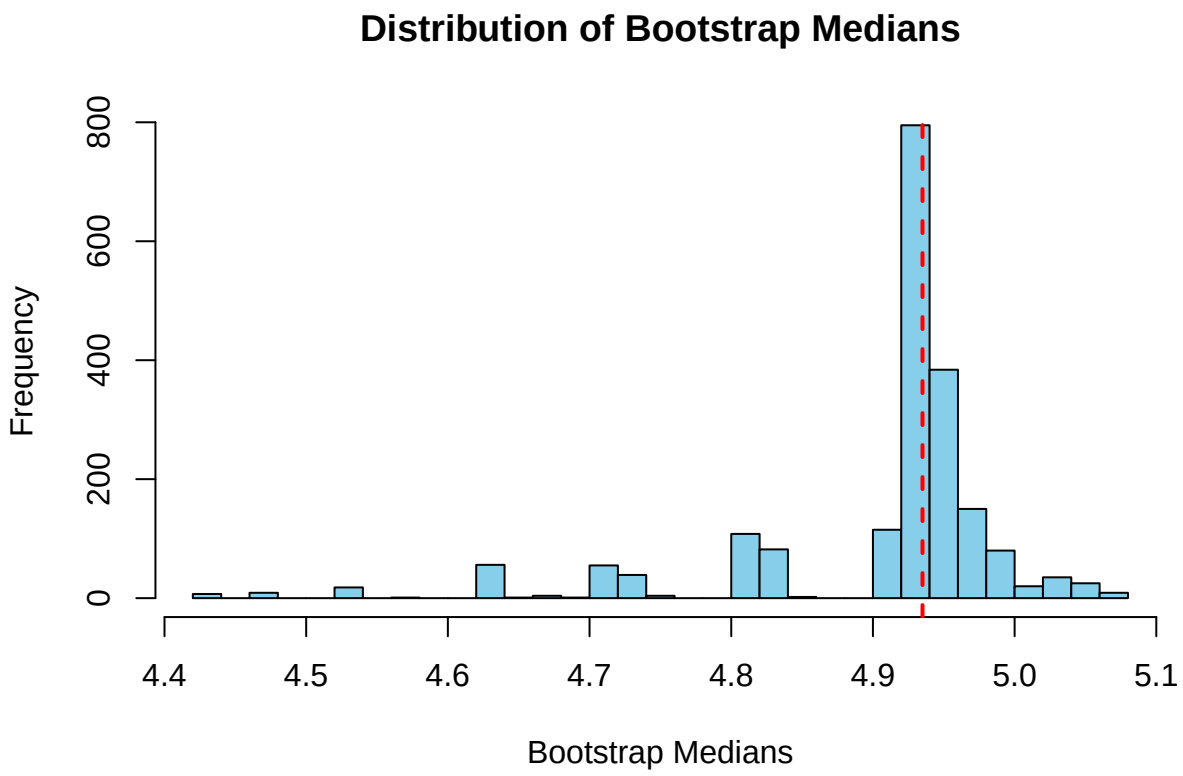
```r
# 2. Compute the mean of the first 200 bootstrap medians
mean_first_200_medians <- mean(bootstrap_medians[1:200])
cat("Mean of the first 200 bootstrap medians:", mean_first_200_medians, "\n")
```

```
## Mean of the first 200 bootstrap medians: 4.8989
```

```r
# 3. Compute the mean based on all 2000 bootstrap medians
mean_all_2000_medians <- mean(bootstrap_medians)
cat("Mean of all 2000 bootstrap medians:", mean_all_2000_medians, "\n")
```

```
## Mean of all 2000 bootstrap medians: 4.907913
```

```
# 4. Visualise the distribution of bootstrap medians
hist(
  bootstrap_medians,
  breaks = 30,
  main = "Distribution of Bootstrap Medians",
  xlab = "Bootstrap Medians",
  col = "skyblue"
)
abline(v = median(data), col = "red", lwd = 2, lty = 2)  # Add the sample median
```




```
# 5. Compute 0.025 and 0.975 quantiles for different bootstrap lengths
quantiles_20_medians <- quantile(bootstrap_medians[1:20], c(0.025, 0.975))
quantiles_200_medians <- quantile(bootstrap_medians[1:200], c(0.025, 0.975))
quantiles_2000_medians <- quantile(bootstrap_medians, c(0.025, 0.975))

# Print quantiles
cat("Quantiles for first 20 bootstrap medians: ", quantiles_20_medians, "\n")
```

```
## Quantiles for first 20 bootstrap medians:  4.548875 5.019625
```

```
cat("Quantiles for first 200 bootstrap medians: ", quantiles_200_medians, "\n")
```

```
## Quantiles for first 200 bootstrap medians:  4.622625 4.99025
```

```
cat("Quantiles for all 2000 bootstrap medians: ", quantiles_2000_medians, "\n")
```

```
## Quantiles for all 2000 bootstrap medians:  4.625 5.025
```

Results

1. Means of Bootstrap Samples

- **First 20 bootstrap means**: `4.882`
- **First 200 bootstrap means**: `4.899`
- **All 2000 bootstrap means**: `4.908`

The means of the bootstrap medians increase slightly as the number of bootstrap samples increases, converging toward the overall mean of all 2000 medians.

---

2. Quantiles (95% Confidence Intervals)

Bootstrap Samples	0.025 Quantile	0.975 Quantile
First 20	4.549	5.020
First 200	4.623	4.990
All 2000	4.625	5.025

---

3. Observations While the distribution is not perfectly symmetric, the bulk of the bootstrap medians is tightly clustered near the sample median, demonstrating that the bootstrap provides a reliable approximation of the sampling distribution. The slight asymmetry suggests that medians (a robust measure of central tendency) may not always conform to the normality assumption, but the distribution is still interpretable for interval estimation. The histogram reinforces the confidence in using bootstrap methods to estimate the variability and confidence intervals of the sample median. The concentration around the original median (4.907913) indicates the robustness of the median, even in the presence of variability in the bootstrap resampling process.

Explore the effect of outliers on the outcomes of Bootstrap Sampling

Create the Data

```r
# Step 1: Generate x.clean and x.cont
set.seed(1234)  # Set initial seed
x.clean <- rnorm(1960, mean = 0, sd = 1)  # 1960 points from N(0, 1)
x.cont <- runif(40, min = 4, max = 5)     # 40 outlier points from U(4, 5)
x <- c(x.clean, x.cont)                   # Combine clean data with outliers

# Set seed to immatriculation number
set.seed(1328781)
```

Estimate the Median, Mean, and Trimmed Mean

```r
# Median, Mean, and Trimmed Mean (alpha = 0.05)
alpha <- 0.05
median_x <- median(x)
mean_x <- mean(x)
trimmed_mean_x <- mean(x, trim = alpha)

median_x_clean <- median(x.clean)
```

```r
mean_x_clean <- mean(x.clean)
trimmed_mean_x_clean <- mean(x.clean, trim = alpha)

cat("Estimates for x (with outliers):\n")
```

```
## Estimates for x (with outliers):
```

```r
cat("Median:", median_x, "\nMean:", mean_x, "\nTrimmed Mean:", trimmed_mean_x, "\n")
```

```
## Median: 0.0113797
## Mean: 0.08395508
## Trimmed Mean: 0.03683294
```

```r
cat("Estimates for x.clean (no outliers):\n")
```

```
## Estimates for x.clean (no outliers):
```

```r
cat("Median:", median_x_clean, "\nMean:", mean_x_clean, "\nTrimmed Mean:", trimmed_mean_x_clean, "\n")
```

```
## Median: -0.0172536
## Mean: -0.005968976
## Trimmed Mean: -0.001462623
```

Nonparametric Bootstrap

```r
# Function to calculate bootstrap estimates
bootstrap <- function(data, stat_function, n_bootstrap = 2000) {
  boot_samples <- replicate(n_bootstrap, stat_function(sample(data, replace = TRUE)))
  std_error <- sd(boot_samples)
  ci <- quantile(boot_samples, c(0.025, 0.975))
  list(se = std_error, ci = ci)
}

# Apply bootstrap to x and x.clean for each statistic
set.seed(1234)  # Consistent results
boot_median_x <- bootstrap(x, median)
boot_mean_x <- bootstrap(x, mean)
boot_trimmed_x <- bootstrap(x, function(data) mean(data, trim = alpha))

boot_median_clean <- bootstrap(x.clean, median)
boot_mean_clean <- bootstrap(x.clean, mean)
boot_trimmed_clean <- bootstrap(x.clean, function(data) mean(data, trim = alpha))

# Display results
cat("Bootstrap results for x (with outliers):\n")
```

```
## Bootstrap results for x (with outliers):
```

```r
cat("Median - SE:", boot_median_x$se, ", 95% CI:", boot_median_x$ci, "\n")
```

```
## Median - SE: 0.02900692 , 95% CI: -0.04361683 0.06603434
```

```r
cat("Mean - SE:", boot_mean_x$se, ", 95% CI:", boot_mean_x$ci, "\n")
```

```
## Mean - SE: 0.02587052 , 95% CI: 0.03562963 0.1358448
```

```r
cat("Trimmed Mean - SE:", boot_trimmed_x$se, ", 95% CI:", boot_trimmed_x$ci, "\n")
```

```
## Trimmed Mean - SE: 0.02271149 , 95% CI: -0.005937803 0.08014803
```

```r
cat("Bootstrap results for x.clean (no outliers):\n")
```

```
## Bootstrap results for x.clean (no outliers):
```

```r
cat("Median - SE:", boot_median_clean$se, ", 95% CI:", boot_median_clean$ci, "\n")
```

```
## Median - SE: 0.02697784 , 95% CI: -0.0598802 0.03980495
```

```r
cat("Mean - SE:", boot_mean_clean$se, ", 95% CI:", boot_mean_clean$ci, "\n")
```

```
## Mean - SE: 0.02228079 , 95% CI: -0.0504058 0.0361294
```

```r
cat("Trimmed Mean - SE:", boot_trimmed_clean$se, ", 95% CI:", boot_trimmed_clean$ci, "\n")
```

```
## Trimmed Mean - SE: 0.02227685 , 95% CI: -0.04700681 0.04031197
```

Parametric Bootstrap

```r
# Parametric Bootstrap for x.clean
set.seed(123456)

# Fit normal model to x.clean
mean_clean <- mean(x.clean)
sd_clean <- sd(x.clean)

# Generate parametric bootstrap samples for x.clean
param_bootstrap_clean <- replicate(
  2000,
  {
    sample_clean <- rnorm(length(x.clean), mean = mean_clean, sd = sd_clean)
    list(
      mean = mean(sample_clean),
      trimmed_mean = mean(sample_clean, trim = alpha)
    )
  },
  simplify = FALSE # Prevent automatic simplification
)

# Extract results for x.clean
boot_means_clean <- sapply(param_bootstrap_clean, function(x) x$mean)
boot_trimmed_means_clean <- sapply(param_bootstrap_clean, function(x) x$trimmed_mean)

# Calculate bias, SE, CI, bias-corrected estimate for x.clean
bias_mean_clean <- mean(boot_means_clean) - mean_clean
se_mean_clean <- sd(boot_means_clean)
ci_mean_clean <- quantile(boot_means_clean, c(0.025, 0.975))

bias_trimmed_clean <- mean(boot_trimmed_means_clean) - mean(x.clean, trim = alpha)
se_trimmed_clean <- sd(boot_trimmed_means_clean)
ci_trimmed_clean <- quantile(boot_trimmed_means_clean, c(0.025, 0.975))

# Display results for x.clean
cat("Parametric Bootstrap for x.clean:\n")
```

```
## Parametric Bootstrap for x.clean:
```

```r
cat("Mean - Bias:", bias_mean_clean, ", SE:", se_mean_clean, ", 95% CI:", ci_mean_clean, "\n")
```

```
## Mean - Bias: 0.0002259111 , SE: 0.02191654 , 95% CI: -0.04990178 0.03767329
```

```r
cat("Trimmed Mean - Bias:", bias_trimmed_clean, ", SE:", se_trimmed_clean, ", 95% CI:", ci_trimmed_clean
```

```
## Trimmed Mean - Bias: -0.004352674 , SE: 0.02209885 , 95% CI: -0.05055274 0.03720506
```

```r
# Repeat similar process for x (with robust scale using MAD)
mad_x <- mad(x)
param_bootstrap_x <- replicate(
  2000,
  {
    sample_x <- rnorm(length(x), mean = median(x), sd = mad_x)
    list(
      mean = mean(sample_x),
      trimmed_mean = mean(sample_x, trim = alpha)
    )
  },
  simplify = FALSE # Prevent automatic simplification
)

# Extract results for x
boot_means_x <- sapply(param_bootstrap_x, function(x) x$mean)
boot_trimmed_means_x <- sapply(param_bootstrap_x, function(x) x$trimmed_mean)

bias_mean_x <- mean(boot_means_x) - mean(x)
se_mean_x <- sd(boot_means_x)
ci_mean_x <- quantile(boot_means_x, c(0.025, 0.975))

bias_trimmed_x <- mean(boot_trimmed_means_x) - mean(x, trim = alpha)
se_trimmed_x <- sd(boot_trimmed_means_x)
ci_trimmed_x <- quantile(boot_trimmed_means_x, c(0.025, 0.975))

# Display results for x
cat("Parametric Bootstrap for x (with outliers):\n")
```

```
## Parametric Bootstrap for x (with outliers):
```

```r
cat("Mean - Bias:", bias_mean_x, ", SE:", se_mean_x, ", 95% CI:", ci_mean_x, "\n")
```

```
## Mean - Bias: -0.07195156 , SE: 0.02145574 , 95% CI: -0.02903059 0.05480329
```

```r
cat("Trimmed Mean - Bias:", bias_trimmed_x, ", SE:", se_trimmed_x, ", 95% CI:", ci_trimmed_x, "\n")
```

```
## Trimmed Mean - Bias: -0.02483602 , SE: 0.021691 , 95% CI: -0.02987442 0.05544867
```

Summarize Finding

```r
# Create a summary table
library(knitr)
```

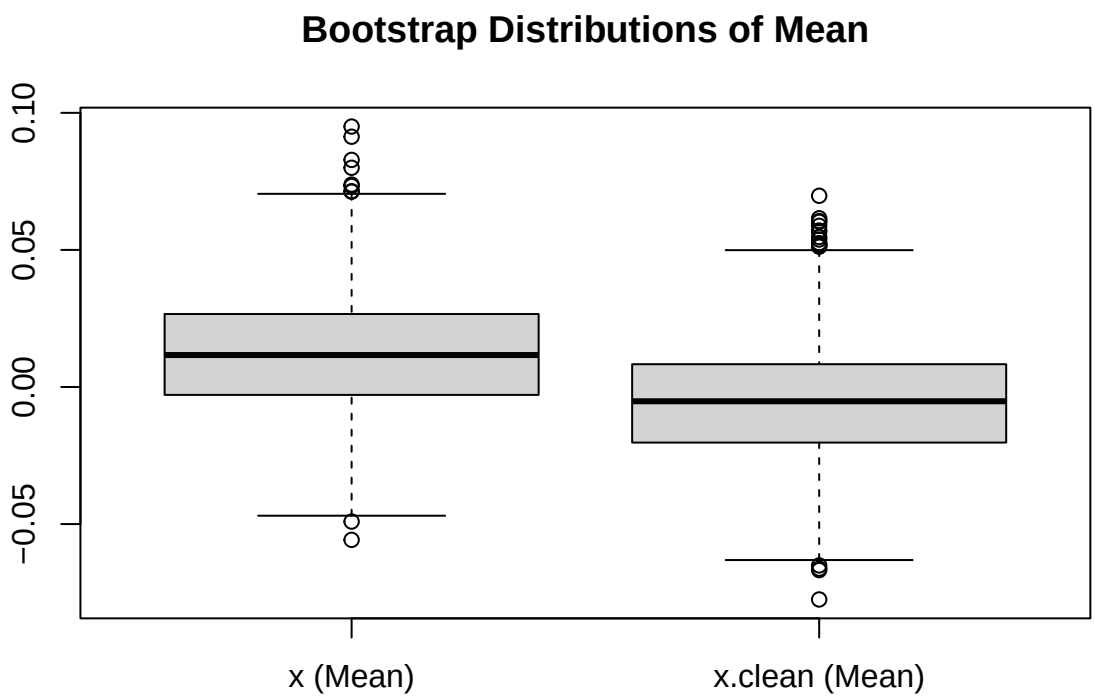
```
## Warning: Paket 'knitr' wurde unter R Version 4.3.2 erstellt
```

```r
results <- data.frame(
  Statistic = c("Median", "Mean", "Trimmed Mean"),
  `SE (x)` = c(boot_median_x$se, boot_mean_x$se, boot_trimmed_x$se),
```

```
  `95% CI Lower (x)` = c(boot_median_x$ci[1], boot_mean_x$ci[1], boot_trimmed_x$ci[1]),
  `95% CI Upper (x)` = c(boot_median_x$ci[2], boot_mean_x$ci[2], boot_trimmed_x$ci[2]),
  `SE (x.clean)` = c(boot_median_clean$se, boot_mean_clean$se, boot_trimmed_clean$se),
  `95% CI Lower (x.clean)` = c(boot_median_clean$ci[1], boot_mean_clean$ci[1], boot_trimmed_clean$ci[1]
  `95% CI Upper (x.clean)` = c(boot_median_clean$ci[2], boot_mean_clean$ci[2], boot_trimmed_clean$ci[2]
)
kable(results)
```

Statistic	SE..x.	X95..CI.Lower..x	X95..CI.Upper..x	SE..x.clean.	X95..CI.Lower..x.clean	X95..CI.Upper..x.clean.
Median	0.0290069	-0.0436168	0.0660343	0.0269778	-0.0598802	0.0398049
Mean	0.0258705	0.0356296	0.1358448	0.0222808	-0.0504058	0.0361294
Trimmed Mean	0.0227115	-0.0059378	0.0801480	0.0222769	-0.0470068	0.0403120

```
# Visualize bootstrap distributions for mean and trimmed mean
boxplot(boot_means_x, boot_means_clean,
        names = c("x (Mean)", "x.clean (Mean)"),
        main = "Bootstrap Distributions of Mean")
```



The boxplot comparison between the bootstrap distributions of the means for `x` (with outliers) and `x.clean` (without outliers) reveals that the presence of outliers in `x` introduces greater variability, as seen in the wider spread of the bootstrap means for `x`. The confidence intervals for `x` are much broader compared to `x.clean`, reflecting the distortion caused by the outliers. Additionally, the mean of the bootstrap means for `x` is slightly shifted, indicating that outliers can also bias the central tendency estimations.

Methodology of bootstrapping

Bootstrapping is a resampling technique used to estimate the sampling distribution of a statistic by repeatedly drawing samples with replacement from the observed data. For confidence intervals, the bootstrap computes the statistic for multiple resamples and uses the percentile or bias-corrected method to define bounds. Non-parametric bootstrapping relies solely on the data, while parametric bootstrapping assumes an underlying distribution. It is robust to small sample sizes and non-normality, making it suitable for constructing confidence intervals and conducting hypothesis tests, as demonstrated in the tasks where we accounted for outliers and explored the variability of different estimators.