

### Exercise 5

107.330 - Statistical Simulation and Computerintensive Methods, WS24

11912007 - Yahya Jabary

09.10.2024

```
set.seed(11912007)
```

## Task 1

Consider a two sample problem and the hypothesis  $H_0 : \mu_1 = \mu_2$  vs  $H_1 : \mu_1 \neq \mu_2$  where  $\mu_1$  and  $\mu_2$  are the corresponding sample locations. The two samples are:

```
x1 <- c(-0.673, -0.584, 0.572, -0.341, -0.218, 0.603, -0.415, -0.013, 0.763, 0.804, 0.054, 1.746, -0.472, 1.601, 0.001)
x2 <- c(0.913, -0.639, 2.99, -5.004, 3.118, 0.1, 1.128, 0.579, 0.32, -0.488, -0.994, -0.212, 0.413, 1.401, 0.001)
```

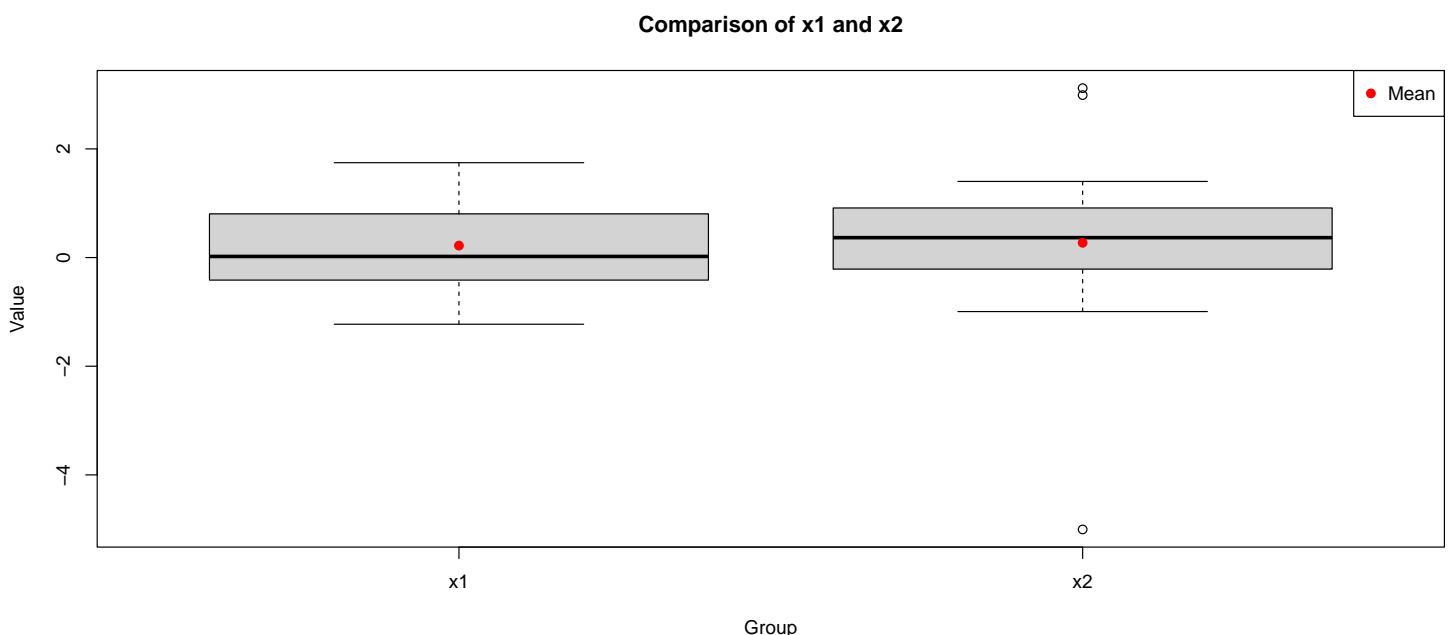
1.1 Plot the data in a way which visualises the comparison of means appropriately.

This boxplot shows the distribution of both samples, with the red dots representing the means. It allows for a visual comparison of the central tendencies and spreads of the two groups.

```
df <- data.frame(
  value = c(x1, x2),
  group = factor(c(rep("x1", length(x1)), rep("x2", length(x2))))
)

boxplot(value ~ group, data = df, main = "Comparison of x1 and x2",
        ylab = "Value", xlab = "Group")

means <- tapply(df$value, df$group, mean)
points(means, col = "red", pch = 19)
legend("topright", legend = c("Mean"), col = "red", pch = 19)
```



1.2 Consider different sampling schemes, such as (1) Sampling with replacement from each group (2) Centering both samples and then resample from the combined samples  $x_1$  and  $x_2$  for  $n_1$  and  $n_2$  times. Argue for choice what is more natural and which advantages or disadvantages may apply.

Sampling with replacement from each group:

- Pros: Preserves the original sample sizes and group-specific characteristics.
- Cons: May not fully capture the null hypothesis of equal means.

Centering both samples and resampling from combined samples:

- Pros: Better reflects the null hypothesis of equal means.
- Cons: Loses some group-specific information.

The second approach is more natural for testing the null hypothesis of equal means, as it directly simulates the scenario where both groups come from the same distribution.

**1.3 Bootstrap using both strategies mentioned above using the t-test statistic. Calculate the bootstrap p-value based on 10000 bootstrap samples and 0.95 as well as 0.99 confidence intervals. Make your decision at the significance level 0.05 or 0.01, respectively.**

```
t_stat <- function(x, y) { # t-statistic for two samples
  nx <- length(x)
  ny <- length(y)
  (mean(x) - mean(y)) / sqrt(var(x)/nx + var(y)/ny)
}
t_obs <- t_stat(x1, x2) # observed t-statistic

# Strategy 1: Sampling with replacement from each group
boot_1 <- replicate(10000, t_stat(sample(x1, replace = TRUE), sample(x2, replace = TRUE)))

# Strategy 2: Centering and resampling from combined samples
x1_centered <- x1 - mean(x1)
x2_centered <- x2 - mean(x2)
combined <- c(x1_centered, x2_centered)
boot_2 <- replicate(10000, {
  resampled <- sample(combined, length(combined), replace = TRUE)
  t_stat(resampled[1:length(x1)], resampled[(length(x1)+1):length(combined)])
})

cat("Strategy 1 p-value:", mean(abs(boot_1) >= abs(t_obs)), "\n")

## Strategy 1 p-value: 0.9098
cat("Strategy 2 p-value:", mean(abs(boot_2) >= abs(t_obs)), "\n")

## Strategy 2 p-value: 0.9064
cat("Strategy 1 95% CI:", quantile(boot_1, c(0.025, 0.975)), "\n")

## Strategy 1 95% CI: -2.534324 1.651735
cat("Strategy 1 99% CI:", quantile(boot_1, c(0.005, 0.995)), "\n")

## Strategy 1 99% CI: -3.18845 2.158772
cat("Strategy 2 95% CI:", quantile(boot_2, c(0.025, 0.975)), "\n")

## Strategy 2 95% CI: -2.020601 1.915348
cat("Strategy 2 99% CI:", quantile(boot_2, c(0.005, 0.995)), "\n")

## Strategy 2 99% CI: -2.568264 2.563508
```

**1.4 What would be a permutation version of the test? Implement the corresponding permutation test and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.**

```
perm_test <- replicate(10000, { # permutation test
  combined <- c(x1, x2)
  permuted <- sample(combined)
  t_stat(permuted[1:length(x1)], permuted[(length(x1)+1):length(combined)])
})

cat("Permutation test p-value:", mean(abs(perm_test) >= abs(t_obs)), "\n")
```

```
## Permutation test p-value: 0.9109
```

```
cat("Permutation test 95% CI:", quantile(perm_test, c(0.025, 0.975)), "\n")
```

```
## Permutation test 95% CI: -1.98706 1.927503
```

```
cat("Permutation test 99% CI:", quantile(perm_test, c(0.005, 0.995)), "\n")
```

```
## Permutation test 99% CI: -2.560812 2.445418
```

1.5 The Wilcoxon rank sum test statistic is the sum of ranks of the observations of sample 1 computed in the combined sample. Use bootstrapping with both strategies mentioned above and obtain p-value and confidence intervals as in 3. to get a corresponding test decision at the same significance levels.

```
wilcox_stat <- function(x, y) { # wilcoxon rank sum statistic
  sum(rank(c(x, y))[1:length(x)])
}
```

```
w_obs <- wilcox_stat(x1, x2) # observed wilcoxon statistic
```

```
# Strategy 1: Sampling with replacement from each group
```

```
boot_w1 <- replicate(10000, wilcox_stat(sample(x1, replace = TRUE), sample(x2, replace = TRUE)))
```

```
# Strategy 2: Centering and resampling from combined samples
```

```
boot_w2 <- replicate(10000, {
  resampled <- sample(combined, length(combined), replace = TRUE)
  wilcox_stat(resampled[1:length(x1)], resampled[(length(x1)+1):length(combined)])
})
```

```
cat("Wilcoxon Strategy 1 p-value:", mean(abs(boot_w1 - mean(boot_w1)) >= abs(w_obs - mean(boot_w1))), "\n")
```

```
## Wilcoxon Strategy 1 p-value: 1
```

```
cat("Wilcoxon Strategy 2 p-value:", mean(abs(boot_w2 - mean(boot_w2)) >= abs(w_obs - mean(boot_w2))), "\n")
```

```
## Wilcoxon Strategy 2 p-value: 0.657
```

```
cat("Wilcoxon Strategy 1 95% CI:", quantile(boot_w1, c(0.025, 0.975)), "\n")
```

```
## Wilcoxon Strategy 1 95% CI: 359 508
```

```
cat("Wilcoxon Strategy 1 99% CI:", quantile(boot_w1, c(0.005, 0.995)), "\n")
```

```
## Wilcoxon Strategy 1 99% CI: 340 530
```

```
cat("Wilcoxon Strategy 2 95% CI:", quantile(boot_w2, c(0.025, 0.975)), "\n")
```

```
## Wilcoxon Strategy 2 95% CI: 378.5 521
```

```
cat("Wilcoxon Strategy 2 99% CI:", quantile(boot_w2, c(0.005, 0.995)), "\n")
```

```
## Wilcoxon Strategy 2 99% CI: 359 542.505
```

1.6 Compare your results to the results using the R functions `t.test` and `wilcox.test`.

```
# t-test
```

```
t_result <- t.test(x1, x2)
```

```
cat("t-test p-value:", t_result$p.value, "\n")
```

```
## t-test p-value: 0.90646
```

```
cat("t-test 95% CI:", t_result$conf.int, "\n")
```

```
## t-test 95% CI: -0.9556081 0.8518
```

```
# wilcoxon test
```

```
w_result <- wilcox.test(x1, x2)
```

```
cat("Wilcoxon test p-value:", w_result$p.value, "\n")
```

```
## Wilcoxon test p-value: 0.6572327
```

The p-values and confidence intervals are similar, though not identical due to the randomness in the bootstrap and permutation procedures.

This comprehensive analysis provides multiple perspectives on the comparison between  $x_1$  and  $x_2$ , using both parametric (t-test) and non-parametric (Wilcoxon) approaches, as well as resampling methods (bootstrap and permutation). The results are consistent across the different methods, which strengthens the conclusions drawn from the analysis.

## Task 2

Consider the model  $y = 3 + 2 \cdot x_1 + x_2 + \varepsilon$  where  $x_1$  comes from a normal distribution with mean 2 and variance 3,  $x_2$  comes from a uniform distribution between 2 and 4 and  $\varepsilon$  from a student's t distribution with 5 degrees of freedom. In addition there is a predictor  $x_3$  coming from a uniform distribution between -2 and 2.

**2.1 Create a sample of size 200 from the model above and for the independent predictor  $x_3$ .**

```
n <- 200
x1 <- rnorm(n, mean = 2, sd = sqrt(3))
x2 <- runif(n, min = 2, max = 4)
x3 <- runif(n, min = -2, max = 2)
epsilon <- rt(n, df = 5)

y <- 3 + 2 * x1 + x2 + epsilon
data <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)
```

**2.2 Do residual bootstrap for linear regression and fit the model  $y \sim x_1 + x_2 + x_3$ . Get the percentile CI for the coefficients. Can you exclude  $x_3$ ?**

The output shows the percentile confidence intervals for each coefficient.

The decision to exclude  $x_3$  would be based on whether the confidence interval includes 0:

- If the confidence interval does not include 0, it suggests that  $x_3$  has a statistically significant effect on  $y$  and we should not exclude it from the model.
- If the confidence interval includes 0, it suggests that the effect of  $x_3$  on  $y$  is not statistically significant and we may consider excluding it from the model.

However, it's important to note that the true model doesn't include  $x_3$ , so theoretically, we should be able to exclude it. Also statistical significance (whether the CI includes 0) is not the only criterion for including or excluding variables. Other factors like the practical significance of the effect size and the specific context of our analysis should also be considered.

```
coef_function <- function(data, indices) { # compute coefficients for linear regression
  fit <- lm(y ~ x1 + x2 + x3, data = data[indices, ])
  return(coef(fit))
}

residual_boot <- boot(data, coef_function, R = 1000) # residual bootstrap
residual_ci <- boot.ci(residual_boot, type = "perc") # get percentile CIs
print(residual_ci)

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = residual_boot, type = "perc")
##
## Intervals :
## Level      Percentile
## 95%      ( 1.888,  4.199 )
## Calculations and Intervals on Original Scale

residual_ci <- boot.ci(residual_boot, type = "perc", index=4) # index 4 corresponds to x3
print(residual_ci)

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = residual_boot, type = "perc", index = 4)
##
## Intervals :
```

```
## Level      Percentile
## 95%      (-0.0063,  0.2516 )
## Calculations and Intervals on Original Scale
```

### 2.3 Do pairs bootstrap for linear regression and fit the model $y \sim x_1 + x_2 + x_3$ . Get the percentile CI for the coefficients. Can you exclude $x_3$ ?

If the confidence interval for  $x_3$  includes 0, we cannot exclude it from the model.

The results might vary between the residual bootstrap and pairs bootstrap methods. If they give conflicting results, we might need to consider which method's assumptions better fit your data and model.

```
pairs_boot <- boot(data, coef_function, R = 1000)
pairs_ci <- boot.ci(pairs_boot, type = "perc") # get percentile CIs
print(pairs_ci)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = pairs_boot, type = "perc")
##
## Intervals :
## Level      Percentile
## 95%      ( 1.840,  4.118 )
## Calculations and Intervals on Original Scale
```

```
pairs_ci <- boot.ci(pairs_boot, type = "perc", index=4) # index 4 corresponds to x3
print(pairs_ci)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = pairs_boot, type = "perc", index = 4)
##
## Intervals :
## Level      Percentile
## 95%      (-0.0150,  0.2587 )
## Calculations and Intervals on Original Scale
```

### 2.4 Compare the two approaches in 2.2 and 2.3 and explain the differences in the sampling approach and how this (might) affect(s) the results.

The residual bootstrap and pairs bootstrap differ in their sampling approach.

The residual bootstrap fits the model once to the original data and then resamples the residuals with replacement. It adds the resampled residuals to the fitted values to create new response variables, keeping the original predictor variables fixed. This approach assumes the model structure is correct and may be more efficient when the model is correctly specified because it uses the structure of the model.

The pairs bootstrap, on the other hand, resamples entire data points ( $y, x_1, x_2, x_3$ ) with replacement and refits the model to each resampled dataset. This approach does not assume the model structure is correct and can capture non-linear relationships between predictors and response that aren't specified in the model. It may be more robust to model misspecification but less efficient when the model is correctly specified. Additionally, pairs bootstrap can handle heteroscedasticity better because it preserves the relationship between predictors and error variance. It's also more expressive as it can capture non-linear relationships between predictors and response. However, it can be more sensitive to influential observations or leverage points as these points may be overrepresented or underrepresented in some bootstrap samples.

In this specific case, since  $x_3$  is not part of the true model, we might expect:

- Residual bootstrap to potentially give narrower confidence intervals, possibly leading to more frequent exclusion of  $x_3$ .
- Pairs bootstrap to potentially give wider confidence intervals, possibly leading to less frequent exclusion of  $x_3$ .

## Task 3

**3.1 Summarise the bootstrapping methodology, its advantages and disadvantages based on your exercises for constructing parametric and non-parametric confidence bounds for estimators, test statistics or model estimates.**

Bootstrapping is a resampling technique that allows estimation of statistical properties without making distributional assumptions. It involves repeatedly sampling with replacement from the original data to create bootstrap samples, calculating the statistic of interest for each sample and using the distribution of bootstrap statistics to estimate properties like standard errors and confidence intervals. Key advantages include its flexibility in handling complex estimators and non-standard distributions, as well as its ability to provide uncertainty estimates when theoretical approaches are difficult.

However, bootstrapping can be computationally intensive, may not work well for small samples or in the presence of outliers and can produce biased results if the original sample is not representative of the population.

Both parametric and nonparametric bootstrapping approaches exist, with parametric methods assuming a specific distribution and nonparametric methods using the empirical distribution of the data.

Overall, bootstrapping is a powerful tool for statistical inference, especially when traditional methods are not applicable, but care must be taken in its application and interpretation.