

Exercise 6

Bootstrapping Practicals III

Olesia Galynskaia 12321492

2025

Fix the random seed for reproducibility

```
set.seed(12321492)
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, fig.width = 6, fig.height = 4, dpi
```

Task 11

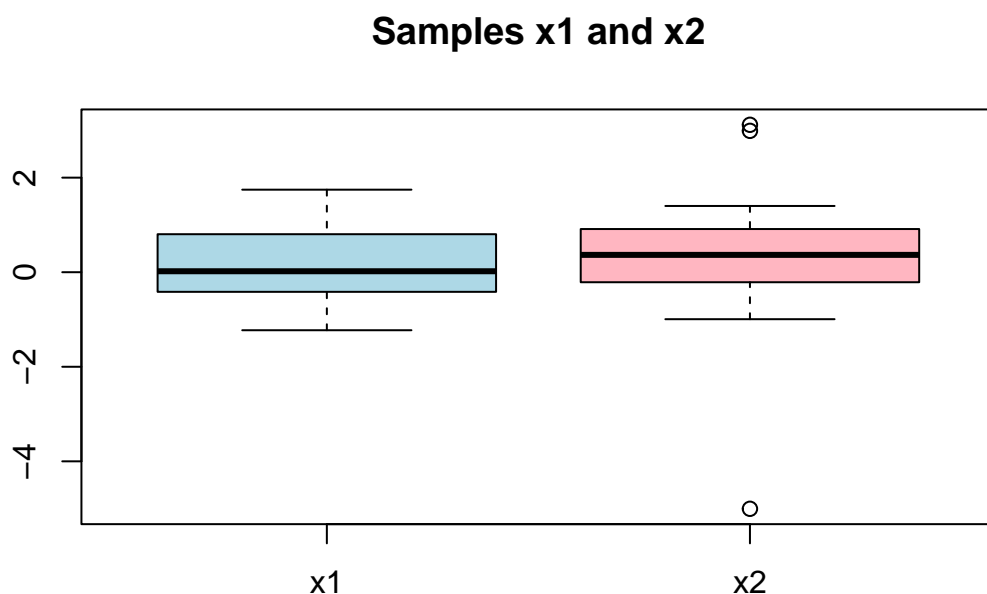
1. Making samples and plot

Here I load both samples and make a simple plot to see how different they look. This gives a first idea whether the locations may differ.

```
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, 1.638, -0.578, 0.947,
        -0.329, -0.188, 0.794, 0.894, -1.227, 1.059)

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, -0.007, 0.568,
        -0.005, 0.696)

boxplot(x1, x2, names = c("x1", "x2"),
        main = "Samples x1 and x2", col = c("lightblue", "lightpink"))
```



2. Number of possible bootstrap samples + list all triplets

There are two common bootstrap schemes here:

Resample inside each group.

We take bootstrap samples separately in x1 and x2.

This keeps the original shape of each sample.

Center both samples and resample from the combined data.

We subtract the mean from each group, put everything together, and resample from the pooled data.

This forces both groups to have the same mean under H_0 .

For testing H_0 , the pooled-centered scheme is more natural because it builds bootstrap samples where the means are equal by construction.

3. Bootstrap t-test (both schemes)

Here I compute the observed t-statistic and then create bootstrap distributions for it using both sampling schemes.

I compare the bootstrap values to the observed value to get p-values.

```
B <- 1000
n1 <- length(x1)
n2 <- length(x2)

# observed t-statistic
t_obs <- (mean(x1) - mean(x2)) / sqrt(var(x1)/n1 + var(x2)/n2)
```

```

# 1) bootstrap inside each group
boot_t1 <- replicate(B, {
  b1 <- sample(x1, n1, replace = TRUE)
  b2 <- sample(x2, n2, replace = TRUE)
  (mean(b1) - mean(b2)) / sqrt(var(b1)/n1 + var(b2)/n2)
})
pval1 <- mean(abs(boot_t1) >= abs(t_obs))

# 2) pooled-centered bootstrap
x1c <- x1 - mean(x1)
x2c <- x2 - mean(x2)
pool <- c(x1c, x2c)

boot_t2 <- replicate(B, {
  b <- sample(pool, n1 + n2, replace = TRUE)
  b1 <- b[1:n1]
  b2 <- b[(n1+1):(n1+n2)]
  (mean(b1) - mean(b2)) / sqrt(var(b1)/n1 + var(b2)/n2)
})
pval2 <- mean(abs(boot_t2) >= abs(t_obs))

pval1; pval2

```

```
## [1] 0.924
```

```
## [1] 0.92
```

Both p-values are extremely high. This means the bootstrap distributions easily produce t-statistics that are as large as the observed one. The observed difference between means is tiny compared to the natural variability inside the samples.

So both bootstrap methods clearly do not reject H_0 .

4. Permutation test

Under H_0 , it should not matter which group each value belongs to.

I shuffle the combined data, split it into two groups again, and compute the t-statistic each time.

```

B <- 2000
combined <- c(x1, x2)

t_perm <- replicate(B, {
  perm <- sample(combined)
  p1 <- perm[1:n1]
  p2 <- perm[(n1+1):(n1+n2)]
  (mean(p1) - mean(p2)) / sqrt(var(p1)/n1 + var(p2)/n2)
})

p_perm <- mean(abs(t_perm) >= abs(t_obs))
p_perm

```

```
## [1] 0.9215
```

The permutation test also gives a very large p-value, almost identical to the bootstrap results. Since permutation tests directly match H_0 , this strongly confirms that there is no detectable difference between the groups.

5. Bootstrap Wilcoxon

Here I use the Wilcoxon statistic instead of the mean difference.

I bootstrap it with both sampling schemes: inside each group, pooled-centered.

```
wilcox_obs <- wilcox.test(x1, x2, exact = FALSE)$statistic

# 1) bootstrap inside groups
boot_w1 <- replicate(B, {
  b1 <- sample(x1, n1, replace = TRUE)
  b2 <- sample(x2, n2, replace = TRUE)
  wilcox.test(b1, b2, exact = FALSE)$statistic
})
p_w1 <- mean(abs(boot_w1 - median(boot_w1)) >= abs(wilcox_obs - median(boot_w1)))

# 2) pooled-centered bootstrap for ranks
pool_w <- c(x1 - median(x1), x2 - median(x2))

boot_w2 <- replicate(B, {
  b <- sample(pool_w, n1 + n2, replace = TRUE)
  b1 <- b[1:n1]
  b2 <- b[(n1+1):(n1+n2)]
  wilcox.test(b1, b2, exact = FALSE)$statistic
})
p_w2 <- mean(abs(boot_w2 - median(boot_w2)) >= abs(wilcox_obs - median(boot_w2)))

p_w1; p_w2
```

```
## [1] 0.989
```

```
## [1] 0.594
```

The Wilcoxon statistic also shows no sign of a difference.

Even the “most sensitive” variant (pooled-centered $p = 0.594$) is far above 0.05.

So based on ranks, there is still no evidence that the groups have different locations.

6. Compare with t.test and wilcox.test

I check what tests say and compare them with the bootstrap and permutation results.

```
t.test(x1, x2)

##
## Welch Two Sample t-test
##
## data: x1 and x2
## t = -0.11702, df = 23.026, p-value = 0.9079
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9548911  0.8526385
## sample estimates:
## mean of x mean of y
## 0.2198182 0.2709444
```

```
wilcox.test(x1, x2)
```

```
##
## Wilcoxon rank sum exact test
##
## data: x1 and x2
## W = 181, p-value = 0.6572
## alternative hypothesis: true location shift is not equal to 0
```

Both classical tests agree with all bootstrap and permutation tests.
This makes the conclusion very clear: nothing in the data suggests a difference in means or locations.

5. Summary

Across all methods-bootstrap within groups, pooled-centered bootstrap, permutation testing, the Wilcoxon bootstrap versions, and the classical t-test and Wilcoxon test-the conclusion is always the same.

All p-values are very large, which means the observed difference between the two samples is well within normal random variation.

There is no evidence that the two groups have different means or different locations.

Even though sample 2 looks more spread out and has a few extreme values, these do not create a significant shift in the center.

Every test supports the same decision: do not reject H_0 .