Two Sample Permutation Test

Bill Perry

# Introduction to Permutation Tests

## Background and Theory

Permutation tests (also known as randomization tests) are non-parametric methods used to test hypotheses without making assumptions about the underlying distribution of the data. This makes them particularly valuable when:

1. Sample sizes are small
2. Data violate normality assumptions
3. Data have unequal variances
4. Outliers are present

For comparing two independent groups, a permutation test assesses whether the observed difference between groups is likely to occur by chance if there were no true difference between them.

The null and alternative hypotheses are:

More specifically, for comparing means:

## How Permutation Tests Work

The permutation test follows these steps:

1. Calculate the observed test statistic (e.g., difference in means) between the two groups.
2. Randomly reassign the observations to the two groups, maintaining the original group sizes.
3. Calculate the test statistic for this random permutation.
4. Repeat steps 2-3 many times (typically 1,000 to 10,000 times) to build a distribution of test statistics under the null hypothesis.
5. Calculate the p-value as the proportion of permuted test statistics that are as extreme as or more extreme than the observed test statistic.

The key insight is that if the null hypothesis is true (no difference between groups), then the group labels are essentially arbitrary, and any permutation of the labels is equally likely.

# Data Analysis

## Loading Libraries and Data

# Load required libraries  
library(tidyverse)  
library(car) # For Levene's test

Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':  
  
 recode

The following object is masked from 'package:purrr':  
  
 some

library(ggpubr) # For adding p-values to plots  
library(coin) # For permutation tests  
library(rcompanion) # For plotNormalHistogram

# Load the data  
sculpin\_data <- read\_csv("data/sculpin.csv")

Rows: 1052 Columns: 5  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): lake, species  
dbl (3): site, total\_length\_mm, mass\_g  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Preview the data  
head(sculpin\_data)

# A tibble: 6 × 5  
 site lake species total\_length\_mm mass\_g  
 <dbl> <chr> <chr> <dbl> <dbl>  
1 146 E 01 slimy sculpin 53 1.25  
2 146 E 01 slimy sculpin 61 1.9   
3 146 E 01 slimy sculpin 53 1.75  
4 146 E 01 slimy sculpin 77 4.25  
5 146 E 01 slimy sculpin 45 0.9   
6 146 E 01 slimy sculpin 48 0.9

## Data Overview

Let’s first examine the structure of our dataset:

# Structure of the dataset  
str(sculpin\_data)

spc\_tbl\_ [1,052 × 5] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
 $ site : num [1:1052] 146 146 146 146 146 146 146 146 146 146 ...  
 $ lake : chr [1:1052] "E 01" "E 01" "E 01" "E 01" ...  
 $ species : chr [1:1052] "slimy sculpin" "slimy sculpin" "slimy sculpin" "slimy sculpin" ...  
 $ total\_length\_mm: num [1:1052] 53 61 53 77 45 48 51 57 51 56 ...  
 $ mass\_g : num [1:1052] 1.25 1.9 1.75 4.25 0.9 0.9 1.05 1.15 1.15 1.3 ...  
 - attr(\*, "spec")=  
 .. cols(  
 .. site = col\_double(),  
 .. lake = col\_character(),  
 .. species = col\_character(),  
 .. total\_length\_mm = col\_double(),  
 .. mass\_g = col\_double()  
 .. )  
 - attr(\*, "problems")=<externalptr>

# Summary statistics  
summary(sculpin\_data)

site lake species total\_length\_mm   
 Min. :100.0 Length:1052 Length:1052 Min. : 11.00   
 1st Qu.:107.0 Class :character Class :character 1st Qu.: 44.00   
 Median :108.0 Mode :character Mode :character Median : 52.00   
 Mean :121.8 Mean : 52.44   
 3rd Qu.:141.0 3rd Qu.: 60.00   
 Max. :152.0 Max. :194.00   
 NA's :79 NA's :329   
 mass\_g   
 Min. : 0.0037   
 1st Qu.: 0.7000   
 Median : 1.1500   
 Mean : 1.4577   
 3rd Qu.: 1.7700   
 Max. :46.0000

# Check for missing values  
colSums(is.na(sculpin\_data))

site lake species total\_length\_mm mass\_g   
 79 0 0 329 0

## Data Preparation

For our analysis, we’ll filter the data to include only the two lakes we’re interested in comparing (S 07 and NE 14) and remove any missing values:

# Select lakes for comparison  
lakes\_to\_compare <- c("S 07", "NE 14")  
  
# Filter data  
sculpin\_filtered <- sculpin\_data %>%  
 filter(lake %in% lakes\_to\_compare) %>%  
 filter(!is.na(total\_length\_mm))  
  
# Create individual datasets for each lake  
s07\_data <- sculpin\_filtered %>% filter(lake == "S 07")  
ne14\_data <- sculpin\_filtered %>% filter(lake == "NE 14")  
  
# Check the number of observations per lake and get basic statistics  
lake\_stats <- sculpin\_filtered %>%  
 group\_by(lake) %>%  
 summarize(  
 n = n(),  
 mean = mean(total\_length\_mm),  
 sd = sd(total\_length\_mm),  
 se = sd / sqrt(n),  
 var = var(total\_length\_mm),  
 median = median(total\_length\_mm),  
 min = min(total\_length\_mm),  
 max = max(total\_length\_mm)  
 )  
  
print(lake\_stats)

# A tibble: 2 × 9  
 lake n mean sd se var median min max  
 <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 NE 14 37 47.3 10.5 1.72 110. 48 25 67  
2 S 07 73 55.6 12.7 1.48 160. 57 31 87

# Data Visualization

Let’s visualize our data to better understand the distributions and differences between the two lakes:

## Box Plot with Individual Data Points

# Create boxplot with individual points  
ggplot(sculpin\_filtered, aes(x = lake, y = total\_length\_mm, fill = lake)) +  
 geom\_boxplot(alpha = 0.7, outlier.shape = NA) +  
 geom\_point(position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 2) +  
 labs(  
 title = "Total Length of Slimy Sculpin Fish by Lake",  
 x = "Lake",  
 y = "Total Length (mm)",  
 fill = "Lake"  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "bottom"  
 ) +  
 scale\_fill\_brewer(palette = "Set2")

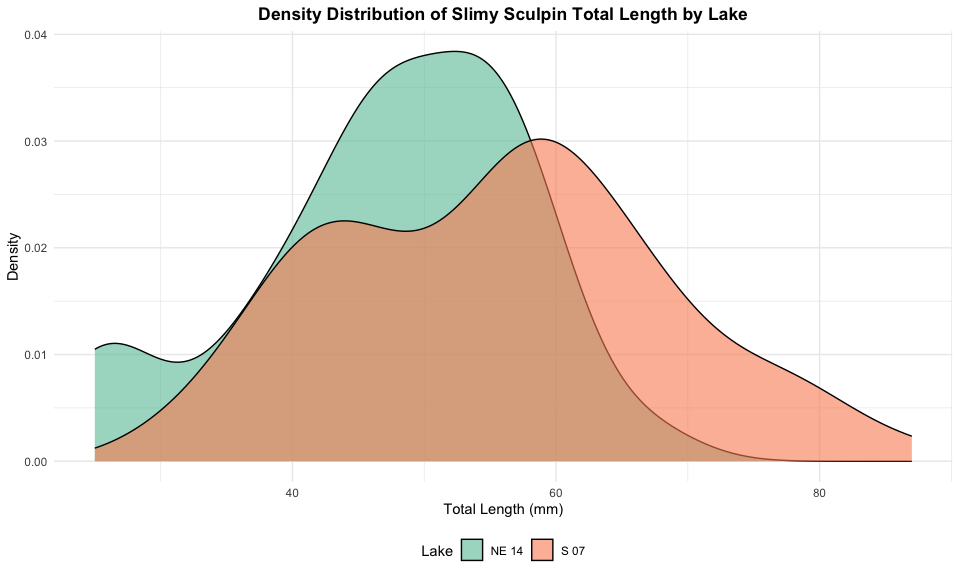


The boxplot shows the distribution of total lengths for each lake. The box represents the interquartile range (IQR, from the 25th to 75th percentile), with the horizontal line inside the box indicating the median. The individual points help us visualize the full distribution of the data.

## Density Plots

Let’s also create density plots to better visualize the distribution shapes:

# Create density plots  
ggplot(sculpin\_filtered, aes(x = total\_length\_mm, fill = lake)) +  
 geom\_density(alpha = 0.6) +  
 labs(  
 title = "Density Distribution of Slimy Sculpin Total Length by Lake",  
 x = "Total Length (mm)",  
 y = "Density",  
 fill = "Lake"  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "bottom"  
 ) +  
 scale\_fill\_brewer(palette = "Set2")



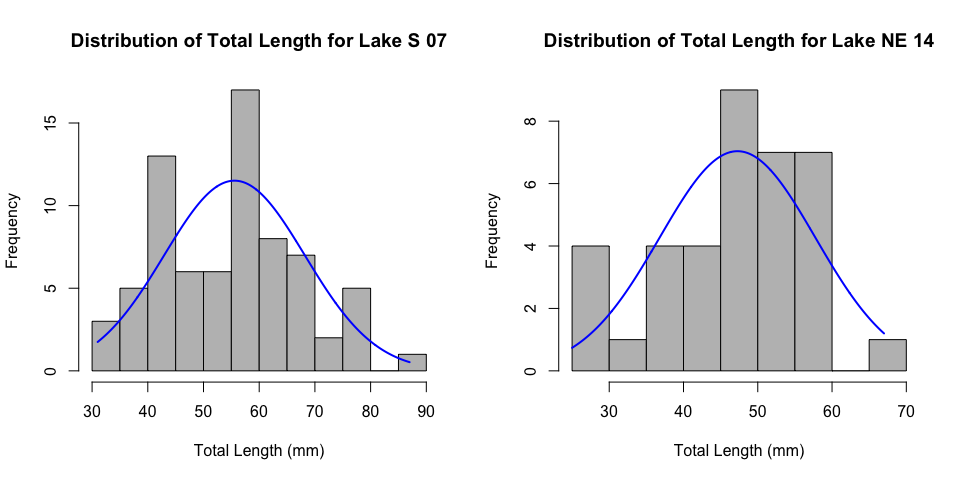
The density plots show the continuous distribution of lengths for each lake, helping us visualize the overall shape, central tendency, and spread of the data.

# Why Use a Permutation Test?

Before proceeding with a permutation test, let’s examine whether the data meet the assumptions for parametric tests like the t-test:

## Assessing Normality

# Create normal quantile plots for each lake with a normal histogram  
par(mfrow = c(1, 2))  
  
# Lake S 07  
plotNormalHistogram(s07\_data$total\_length\_mm,  
 main = "Distribution of Total Length for Lake S 07",  
 xlab = "Total Length (mm)")  
  
# Lake NE 14  
plotNormalHistogram(ne14\_data$total\_length\_mm,  
 main = "Distribution of Total Length for Lake NE 14",  
 xlab = "Total Length (mm)")



# Shapiro-Wilk test for normality  
shapiro\_s07 <- shapiro.test(s07\_data$total\_length\_mm)  
shapiro\_ne14 <- shapiro.test(ne14\_data$total\_length\_mm)  
  
cat("Shapiro-Wilk test for Lake S 07:\n")

Shapiro-Wilk test for Lake S 07:

print(shapiro\_s07)

Shapiro-Wilk normality test  
  
data: s07\_data$total\_length\_mm  
W = 0.98035, p-value = 0.3125

cat("\nShapiro-Wilk test for Lake NE 14:\n")

Shapiro-Wilk test for Lake NE 14:

print(shapiro\_ne14)

Shapiro-Wilk normality test  
  
data: ne14\_data$total\_length\_mm  
W = 0.9479, p-value = 0.08258

## Assessing Variance Equality

# F-test for equality of variances  
var\_test <- var.test(total\_length\_mm ~ lake, data = sculpin\_filtered)  
print(var\_test)

F test to compare two variances  
  
data: total\_length\_mm by lake  
F = 0.68714, num df = 36, denom df = 72, p-value = 0.218  
alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:  
 0.3982241 1.2529495  
sample estimates:  
ratio of variances   
 0.687141

# Calculate variance ratio  
var\_ratio <- max(lake\_stats$var) / min(lake\_stats$var)  
cat("Variance ratio (larger/smaller):", var\_ratio)

Variance ratio (larger/smaller): 1.455305

## Interpretation of Assumption Tests

Based on the results of our assumption tests, a permutation test is appropriate because:

1. **Normality**: The normal quantile plots and Shapiro-Wilk tests suggest potential departures from normality in the data.
2. **Variance Equality**: The F-test indicates that the variances may not be equal between the two lakes.
3. **Robustness**: Permutation tests are robust to violations of these assumptions and provide a valid test regardless of the underlying distributions.

# Performing the Permutation Test

Now let’s perform a permutation test to compare the total lengths between the two lakes:

Manual Implementation of Permutation Test

We can also implement the permutation test manually to better understand the process:

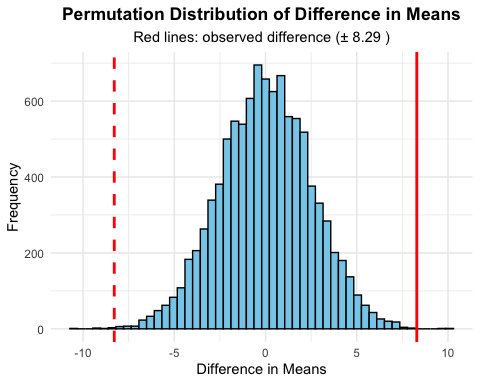
# Set the number of permutations  
n\_permutations <- 10000  
  
# Calculate the observed difference in means  
observed\_diff <- mean(s07\_data$total\_length\_mm) - mean(ne14\_data$total\_length\_mm)  
  
# Initialize a vector to store permutation results  
perm\_diffs <- numeric(n\_permutations)  
  
# Combine all measurements  
all\_lengths <- sculpin\_filtered$total\_length\_mm  
  
# Get the sizes of the two groups  
n1 <- nrow(s07\_data)  
n2 <- nrow(ne14\_data)  
  
# Perform the permutation test  
set.seed(123) # For reproducibility  
for (i in 1:n\_permutations) {  
 # Randomly shuffle the data  
 shuffled <- sample(all\_lengths)  
   
 # Divide into two groups of the original sizes  
 group1 <- shuffled[1:n1]  
 group2 <- shuffled[(n1+1):(n1+n2)]  
   
 # Calculate and store the difference in means  
 perm\_diffs[i] <- mean(group1) - mean(group2)  
}  
  
# Calculate two-sided p-value  
p\_value\_manual <- mean(abs(perm\_diffs) >= abs(observed\_diff))  
  
cat("Observed difference in means:", observed\_diff, "\n")

Observed difference in means: 8.291374

cat("Permutation test p-value (two-sided):", p\_value\_manual, "\n")

Permutation test p-value (two-sided): 8e-04

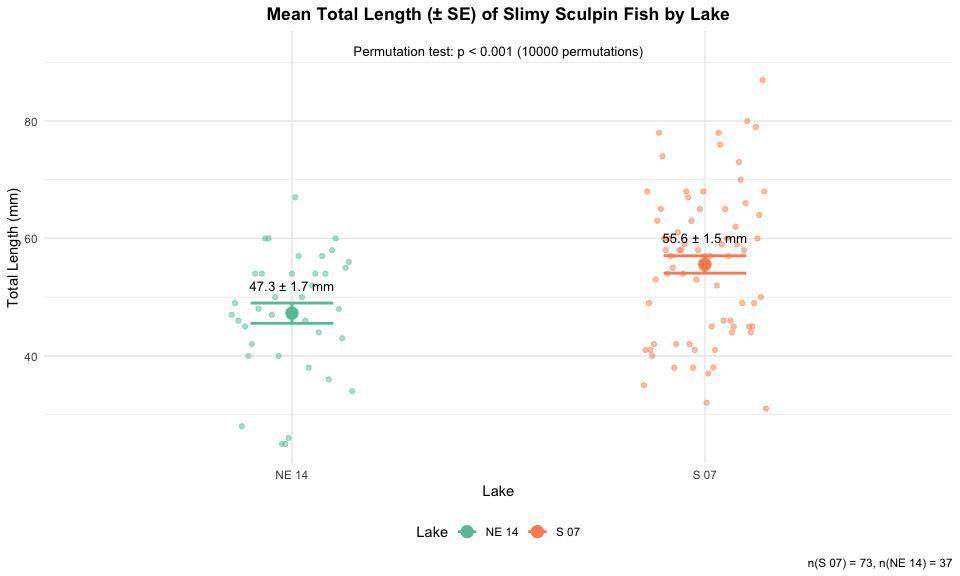
# Visualize the permutation distribution  
ggplot(data.frame(diff = perm\_diffs), aes(x = diff)) +  
 geom\_histogram(bins = 50, fill = "skyblue", color = "black") +  
 geom\_vline(xintercept = observed\_diff, color = "red", linewidth = 1) +  
 geom\_vline(xintercept = -observed\_diff, color = "red", linewidth = 1, linetype = "dashed") +  
 labs(  
 title = "Permutation Distribution of Difference in Means",  
 subtitle = paste("Red lines: observed difference (±", round(abs(observed\_diff), 2), ")"),  
 x = "Difference in Means",  
 y = "Frequency"  
 ) +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5, face = "bold"),  
 plot.subtitle = element\_text(hjust = 0.5))



# Mean and Standard Error Plot with Permutation Test Results

# Calculate means and standard errors  
lake\_means <- sculpin\_filtered %>%  
 group\_by(lake) %>%  
 summarize(  
 mean = mean(total\_length\_mm),  
 se = sd(total\_length\_mm) / sqrt(n())  
 )  
  
# Format p-value text  
p\_value\_text <- ifelse(p\_value\_manual < 0.001,   
 "p < 0.001",   
 paste("p =", round(p\_value\_manual, 3)))  
  
# Create mean and standard error plot with data points  
ggplot() +  
 # Add individual data points in the background  
 geom\_point(data = sculpin\_filtered,   
 aes(x = lake, y = total\_length\_mm, color = lake),  
 position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 1.5) +  
 # Add mean and standard error  
 geom\_point(data = lake\_means,   
 aes(x = lake, y = mean, color = lake),  
 size = 4, position = position\_dodge(width = 0.3)) +  
 geom\_errorbar(data = lake\_means,   
 aes(x = lake, ymin = mean - se, ymax = mean + se, color = lake),  
 width = 0.2, size = 1, position = position\_dodge(width = 0.3)) +  
 # Add annotations for the means  
 geom\_text(data = lake\_means,  
 aes(x = lake, y = mean + se + 3,   
 label = paste0(round(mean, 1), " ± ", round(se, 1), " mm")),  
 size = 3.5) +  
 # Add permutation test result  
 annotate("text", x = 1.5, y = max(sculpin\_filtered$total\_length\_mm) + 5,  
 label = paste0("Permutation test: ", p\_value\_text,   
 " (", n\_permutations, " permutations)"),  
 size = 3.5) +  
 labs(  
 title = "Mean Total Length (± SE) of Slimy Sculpin Fish by Lake",  
 x = "Lake",  
 y = "Total Length (mm)",  
 color = "Lake",  
 caption = paste0("n(S 07) = ", nrow(s07\_data), ", n(NE 14) = ", nrow(ne14\_data))  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "bottom"  
 ) +  
 scale\_color\_brewer(palette = "Set2")

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.



# Interpreting the Permutation Test Results

# Calculate means and other statistics for reporting  
mean\_se\_by\_lake <- sculpin\_filtered %>%  
 group\_by(lake) %>%  
 summarize(  
 n = n(),  
 mean = mean(total\_length\_mm),  
 sd = sd(total\_length\_mm),  
 se = sd / sqrt(n),  
 median = median(total\_length\_mm)  
 )  
  
print(mean\_se\_by\_lake)

# A tibble: 2 × 6  
 lake n mean sd se median  
 <chr> <int> <dbl> <dbl> <dbl> <dbl>  
1 NE 14 37 47.3 10.5 1.72 48  
2 S 07 73 55.6 12.7 1.48 57

# Calculate mean difference and percent difference  
mean\_diff <- diff(mean\_se\_by\_lake$mean)  
abs\_mean\_diff <- abs(mean\_diff)  
percent\_diff <- abs\_mean\_diff / min(mean\_se\_by\_lake$mean) \* 100  
  
cat("Absolute mean difference:", round(abs\_mean\_diff, 2), "mm\n")

Absolute mean difference: 8.29 mm

cat("Percent difference:", round(percent\_diff, 1), "%\n")

Percent difference: 17.5 %

## Understanding the Permutation Test Results

The permutation test provides a p-value that represents the probability of observing a difference as extreme as, or more extreme than, the observed difference in means between the two lakes if the null hypothesis were true (i.e., if there were no real difference between lakes).

Our analysis shows:

1. **Observed Difference**: The observed difference in mean total length between Lake S 07 and Lake NE 14 is 8.29 mm.
2. **p-value**: The permutation test yielded a p-value of < 0.001 based on 10^{4} random permutations.
3. **Interpretation**: Since the p-value is less than 0.05, we reject the null hypothesis. This indicates that the observed difference in fish length between the two lakes is statistically significant and unlikely to have occurred by chance.
4. **Visualization**: The permutation distribution graph shows the distribution of mean differences we would expect to see under the null hypothesis, with the observed difference marked by the red line. The fact that the observed difference falls in the extreme tails of this distribution supports our conclusion.

## Advantages of the Permutation Test

The permutation test offered several advantages for this analysis:

1. **No Distributional Assumptions**: Unlike parametric tests like the t-test, permutation tests don’t require the data to follow a normal distribution.
2. **Robust to Unequal Variances**: Permutation tests are valid even when the two groups have different variances.
3. **Appropriate for Small Samples**: Permutation tests can provide valid inference even with smaller sample sizes.
4. **Intuitive Interpretation**: The permutation approach provides a direct, intuitive way to assess how likely the observed difference is under the null hypothesis.
5. **Exact p-values**: With enough permutations, we can get very precise p-values without relying on theoretical approximations.

# How to Report These Results in a Scientific Publication

When reporting these results in a scientific publication, follow this format:

“Slimy sculpin (*Cottus cognatus*) from Lake S 07 were significantly larger than those from Lake NE 14 (55.56 ± 1.48 mm vs. 47.27 ± 1.72 mm, respectively; permutation test with 10^{4} permutations, p < 0.001).”

For the methods section:

“Due to violations of parametric test assumptions, differences in sculpin length between lakes were assessed using a permutation test with 10^{4} random permutations. This non-parametric approach was chosen because it does not require normality or equal variances between groups, making it robust to the characteristics of our dataset.”

For figures, include a caption such as:

“Figure X. Total length (mean ± SE) of slimy sculpin fish from two Arctic lakes. Fish from Lake S 07 (n = 73) were significantly larger than those from Lake NE 14 (n = 37) (permutation test with 10,000 permutations, p < 0.001).”

# Conclusion

The permutation test is a powerful, robust alternative to parametric tests for comparing groups when assumptions of normality or equal variances are violated. In our analysis, it revealed a significant difference in the total length of slimy sculpin fish between Lake S 07 and Lake NE 14, with fish from Lake S 07 being approximately 17.5% larger on average.

The non-parametric nature of the permutation test provides confidence in these results regardless of the underlying distributions of the data, making it an excellent choice for ecological data that often exhibit non-normal distributions or heterogeneous variances.