Two Sample T-Test

Bill Perry

# Introduction to Two-Sample t-Test

## Background and Theory

The two-sample t-test (also known as independent samples t-test) is used to determine whether there is a statistically significant difference between the means of two independent groups. In this analysis, we will examine whether there are significant differences in the total length of slimy sculpin fish between two different lakes.

The two-sample t-test makes the following comparison:

Where:

* - is the null hypothesis stating that the population means are equal
* - is the alternative hypothesis stating that the population means are different
* - is the population mean of the first group
* - is the population mean of the second group

## Formula

The formula for the two-sample t-test with equal variances (pooled variance) is:

Where:

* - is the sample mean of the first group
* - is the sample mean of the second group
* - is the pooled standard deviation
* - is the sample size of the first group
* - is the sample size of the second group

The pooled standard deviation is calculated as:

Where:

* - is the variance of the first group
* - is the variance of the second group
* The degrees of freedom (df) for this test is .

**For unequal variances (Welch’s t-test), the formula is slightly different:**

**With degrees of freedom approximated using the Welch-Satterthwaite equation:**

# Data Analysis

## Loading Libraries and Data

# Load required libraries  
# install.packages("gt")  
library(gt)  
library(broom)  
library(car) # For Levene's test

Loading required package: carData

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

Loading required package: survival

library(rcompanion) # For plotNormalHistogram  
library(skimr)  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.2 ✔ tibble 3.3.0  
✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
✔ purrr 1.1.0

── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
✖ dplyr::recode() masks car::recode()  
✖ purrr::some() masks car::some()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Load the data  
sculpin\_df <- read\_csv("data/t\_test\_sculpin\_s07\_ne14.csv")

Rows: 110 Columns: 5  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (2): lake, species  
dbl (3): site, 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\_df)

# A tibble: 6 × 5  
 site lake species length\_mm mass\_g  
 <dbl> <chr> <chr> <dbl> <dbl>  
1 109 NE 14 slimy sculpin 47 0.7   
2 109 NE 14 slimy sculpin 49 0.9   
3 109 NE 14 slimy sculpin 46 0.7   
4 109 NE 14 slimy sculpin 28 0.15  
5 109 NE 14 slimy sculpin 45 0.65  
6 109 NE 14 slimy sculpin 40 0.3

## Data Overview

Let’s first examine the structure of our dataset:

sculpin\_df %>%   
 group\_by(lake) %>%   
 skim()

Data summary

|  |  |
| --- | --- |
| Name | Piped data |
| Number of rows | 110 |
| Number of columns | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| numeric | 3 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | lake |

**Variable type: character**

| skim\_variable | lake | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| species | NE 14 | 0 | 1 | 13 | 13 | 0 | 1 | 0 |
| species | S 07 | 0 | 1 | 13 | 13 | 0 | 1 | 0 |

**Variable type: numeric**

| skim\_variable | lake | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| site | NE 14 | 0 | 1 | 109.00 | 0.00 | 109.00 | 109.00 | 109.00 | 109.00 | 109.00 | ▁▁▇▁▁ |
| site | S 07 | 0 | 1 | 152.00 | 0.00 | 152.00 | 152.00 | 152.00 | 152.00 | 152.00 | ▁▁▇▁▁ |
| length\_mm | NE 14 | 0 | 1 | 47.27 | 10.49 | 25.00 | 42.00 | 48.00 | 54.00 | 67.00 | ▂▃▇▇▂ |
| length\_mm | S 07 | 0 | 1 | 55.56 | 12.65 | 31.00 | 45.00 | 57.00 | 64.00 | 87.00 | ▅▅▇▃▂ |
| mass\_g | NE 14 | 0 | 1 | 0.89 | 0.52 | 0.10 | 0.45 | 0.85 | 1.25 | 2.30 | ▇▇▇▂▁ |
| mass\_g | S 07 | 0 | 1 | 1.66 | 1.23 | 0.25 | 0.80 | 1.45 | 2.10 | 7.37 | ▇▃▁▁▁ |

### Manual Summary Method

stats\_df <- sculpin\_df %>%   
 group\_by(lake) %>%   
 summarize(mean\_length\_mm = round(mean(length\_mm, na.rm=TRUE),2),  
 stddev\_length\_mm = round(sd(length\_mm, na.rm=TRUE),2),  
 stderr\_length\_mm = round(sd(length\_mm, na.rm=TRUE)/sum(!is.na(length\_mm)),2),  
 coef\_var\_length\_mm = round((sd(length\_mm, na.rm=TRUE)/mean(length\_mm, na.rm=TRUE))\*100,2)  
 )  
stats\_df

# A tibble: 2 × 5  
 lake mean\_length\_mm stddev\_length\_mm stderr\_length\_mm coef\_var\_length\_mm  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 NE 14 47.3 10.5 0.28 22.2  
2 S 07 55.6 12.6 0.17 22.8

### Fancy Table with the tidytable package

gt\_table <- stats\_df %>%   
 gt() %>%   
 tab\_header(  
 title = "Table 1. Sculpin Length Statistics by Lake") %>%   
 cols\_label(  
 lake = "Lake",  
 mean\_length\_mm = "Mean Length (mm)",  
 stddev\_length\_mm = "Std Dev (mm)",  
 stderr\_length\_mm = "Std Error (mm)",  
 coef\_var\_length\_mm = "CV (%)"  
 )%>%  
 tab\_options(  
 table.border.top.style = "none",  
 table.border.bottom.style = "solid",  
 column\_labels.border.bottom.style = "solid",  
 table\_body.border.top.style = "none",  
 table\_body.hlines.style = "none")%>%  
 opt\_align\_table\_header(align = "left")  
gt\_table

Table 1: Table 1. Sculpin Length Statistics by Lake

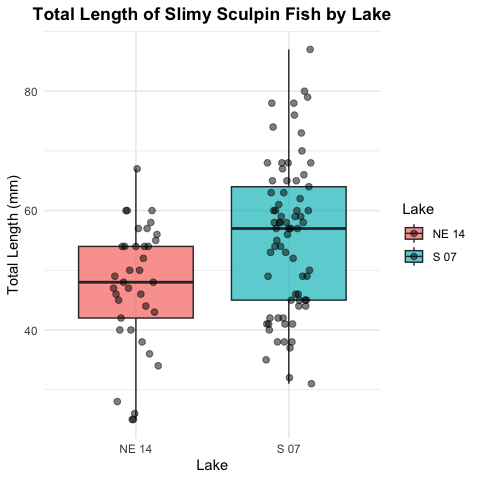
| Lake | Mean Length (mm) | Std Dev (mm) | Std Error (mm) | CV (%) |
| --- | --- | --- | --- | --- |
| NE 14 | 47.27 | 10.49 | 0.28 | 22.19 |
| S 07 | 55.56 | 12.65 | 0.17 | 22.77 |

## Data Visualization

### Box Plot with Individual Data Points

Let’s create a box plot with individual data points to visualize the distribution of total length in the two lakes:

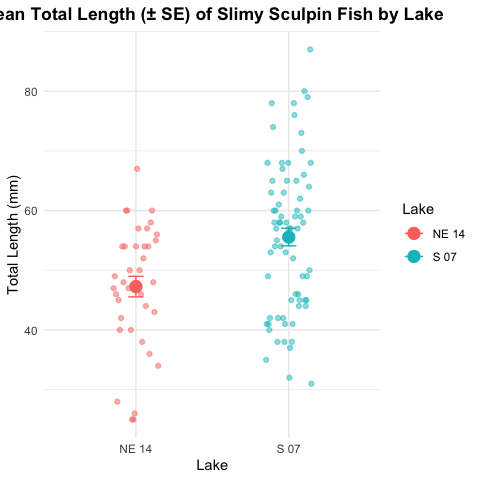
# Create boxplot with individual points  
sculpin\_df %>%   
 ggplot(aes(x = lake, y = length\_mm, fill = lake)) +  
 geom\_boxplot(alpha = 0.7, outlier.shape = NA) +  
 geom\_point(position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 2) +  
 labs(  
 title = "Total Length of Slimy Sculpin 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 = "right"  
 )



### Mean and Standard Error Plot

Now, let’s create a plot showing the mean and standard error for each lake, with individual data points in the background:

# Create mean and standard error plot with data points  
  
sculpin\_df %>%   
ggplot( aes(x = lake, y = length\_mm, color = lake)) +  
 # Add individual data points in the background  
 geom\_point(position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 1.5) +  
 # Add mean and standard error  
 stat\_summary(fun = mean, geom = "point", size = 4) +  
 stat\_summary(fun.data = mean\_se, geom = "errorbar", width = 0.1) +  
 labs(  
 title = "Mean Total Length (± SE) of Slimy Sculpin Fish by Lake",  
 x = "Lake",  
 y = "Total Length (mm)",  
 color = "Lake"  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "right"  
 )



# Testing t-Test Assumptions

Before conducting the t-test, we need to verify that our data meets the underlying assumptions:

## Assumptions of the Two-Sample t-Test

1. **Independence**: The observations within each group are independent, and the two groups are independent of each other.
2. **Normality**: The data in each group follow a normal distribution.
3. **Homogeneity of Variances**: The variances of the two groups are approximately equal (for the standard t-test).

Let’s test each of these assumptions:

### 1. Independence Assumption

Independence is a design issue and can’t be tested statistically. We assume our sampling design ensures independence between and within groups.

### 2. Normality Assumption

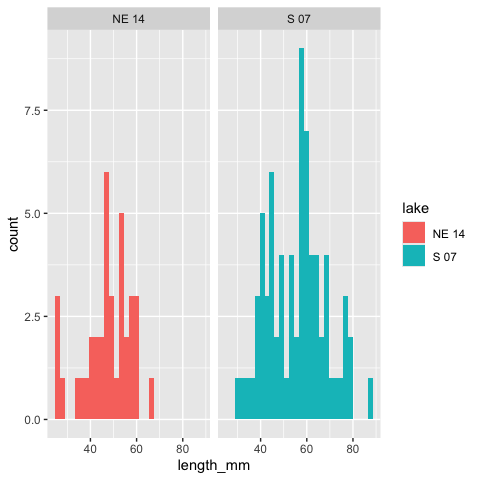
We’ll check normality using:

* - Histograms
* - Q-Q plots
* - Shapiro-Wilk test

#### Histograms

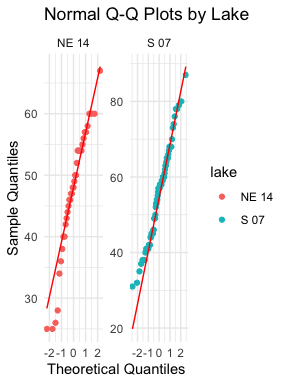
sculpin\_df %>%   
 ggplot(aes(length\_mm, fill = lake))+  
 geom\_histogram()+  
 facet\_wrap(~lake)

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### QQ Plots

# QQ plot for lakes  
sculpin\_df %>%   
 ggplot( aes(sample = length\_mm, color=lake)) +  
 stat\_qq() +  
 stat\_qq\_line(color = "red") +  
 facet\_wrap(~lake, scales = "free") +  
 labs(title = "Normal Q-Q Plots by Lake",  
 x = "Theoretical Quantiles",  
 y = "Sample Quantiles") +  
 theme\_minimal()



### Shapiro-Wilk Test

# Simple approach - just split by lake and run the test  
sculpin\_df %>%  
 filter(lake == "S 07") %>%  
 pull(length\_mm) %>%  
 shapiro.test()

Shapiro-Wilk normality test  
  
data: .  
W = 0.98035, p-value = 0.3125

sculpin\_df %>%  
 filter(lake == "NE 14") %>%  
 pull(length\_mm) %>%  
 shapiro.test()

Shapiro-Wilk normality test  
  
data: .  
W = 0.9479, p-value = 0.08258

### nicer way #1

# using the broom package  
sculpin\_df %>%  
 group\_by(lake) %>%  
 group\_modify(~ broom::tidy(shapiro.test(.x$length\_mm)))

# A tibble: 2 × 4  
# Groups: lake [2]  
 lake statistic p.value method   
 <chr> <dbl> <dbl> <chr>   
1 NE 14 0.948 0.0826 Shapiro-Wilk normality test  
2 S 07 0.980 0.313 Shapiro-Wilk normality test

### nicer way #2

sculpin\_df %>%  
 group\_by(lake) %>%  
 group\_walk(~ {  
 cat("Shapiro-Wilk test for Lake", .y$lake, ":\n")  
 test\_result <- shapiro.test(.x$length\_mm)  
 print(test\_result)  
 cat("\n")  
 })

Shapiro-Wilk test for Lake NE 14 :  
  
 Shapiro-Wilk normality test  
  
data: .x$length\_mm  
W = 0.9479, p-value = 0.08258  
  
  
Shapiro-Wilk test for Lake S 07 :  
  
 Shapiro-Wilk normality test  
  
data: .x$length\_mm  
W = 0.98035, p-value = 0.3125

### 3. Homogeneity of Variances

We’ll check for homogeneity of variances using: - Visual inspection of boxplots (already done above) - Levene’s test

# Levene's test for homogeneity of variances  
leveneTest(length\_mm ~ lake, data = sculpin\_df)

Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
factor.

Levene's Test for Homogeneity of Variance (center = median)  
 Df F value Pr(>F)  
group 1 2.029 0.1572  
 108

## Interpretation of Assumption Tests

Based on the results of our assumption tests:

1. **Independence**: We assume this is met based on the data collection process, as samples from each lake were collected independently of one another.
2. **Normality**:
   * The Q-Q plots show that the data points largely follow the theoretical normal distribution line for both lakes, with some minor deviations at the extremes.
   * The Shapiro-Wilk test results will help us formally assess normality. If the p-value is greater than 0.05, we fail to reject the null hypothesis that the data is normally distributed.
   * For samples larger than 30, the Central Limit Theorem suggests that the sampling distribution of means will be approximately normal regardless of the underlying distribution.
3. **Homogeneity of Variances**:
   * Levene’s test evaluates whether the variances between groups are equal.
   * A p-value greater than 0.05 indicates that we cannot reject the null hypothesis of equal variances.
   * As a rule of thumb, if the variance ratio is less than 4:1, the t-test is reasonably robust to violations of this assumption.
   * If this assumption is violated, we should consider using Welch’s t-test instead, which does not assume equal variances.

# Two-Sample t-Test

Now that we’ve checked the assumptions, we can perform the two-sample t-test:

# Perform the t-test with equal variance (standard t-test)  
t\_test\_equal\_var <- t.test(  
 length\_mm ~ lake,  
 data = sculpin\_df,  
 var.equal = TRUE # Use pooled variance  
)  
  
# Display the results  
t\_test\_equal\_var

Two Sample t-test  
  
data: length\_mm by lake  
t = -3.4314, df = 108, p-value = 0.0008519  
alternative hypothesis: true difference in means between group NE 14 and group S 07 is not equal to 0  
95 percent confidence interval:  
 -13.080929 -3.501818  
sample estimates:  
mean in group NE 14 mean in group S 07   
 47.27027 55.56164

# For comparison, also perform Welch's t-test (unequal variances)  
t\_test\_welch <- t.test(  
 length\_mm ~ lake,  
 data = sculpin\_df,  
 var.equal = FALSE # Use Welch's correction  
)  
  
  
t\_test\_welch

Welch Two Sample t-test  
  
data: length\_mm by lake  
t = -3.6483, df = 85.45, p-value = 0.0004533  
alternative hypothesis: true difference in means between group NE 14 and group S 07 is not equal to 0  
95 percent confidence interval:  
 -12.809687 -3.773061  
sample estimates:  
mean in group NE 14 mean in group S 07   
 47.27027 55.56164

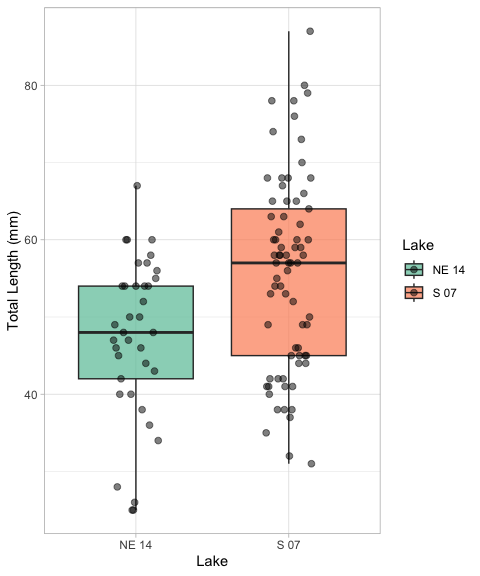
## Line-by-Line Interpretation of t-Test Results

Let’s break down the t-test output:

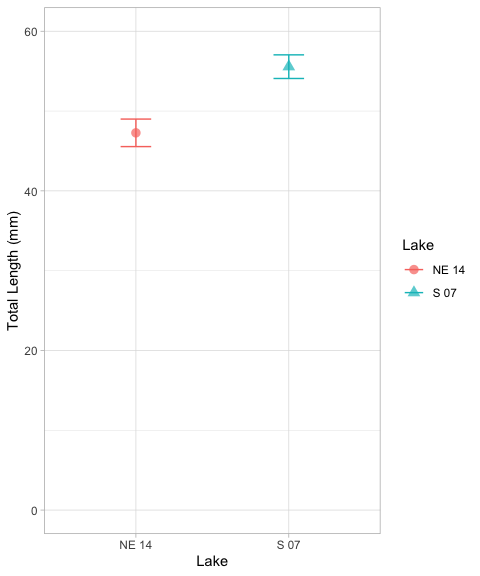
1. **Test Type**: Two Sample t-test
2. **Formula**: length\_mm ~ lake means we’re testing if total length differs by lake
3. **Data**: Our filtered sculpin dataset
4. **t-value**: The calculated t-statistic
5. **Degrees of Freedom (df)**: n₁ + n₂ - 2
6. **p-value**: The probability of observing this data (or more extreme) if the null hypothesis is true
7. **Alternative Hypothesis**: The means are different
8. **95% Confidence Interval**: The estimated range for the true difference in means
9. **Sample Estimates**: The means of each group

## Visual Representation of t-Test Results

sculpin\_df %>%   
 ggplot( aes(x = lake, y = 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(  
 x = "Lake",  
 y = "Total Length (mm)",  
 fill = "Lake") +  
 theme\_light() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "right"  
 ) +  
 scale\_fill\_brewer(palette = "Set2")



sculpin\_df %>%   
 ggplot(aes(x = lake, y = length\_mm, color=lake, shape = lake, fill = lake)) +  
 stat\_summary(fun = mean, geom = "point", alpha = 0.7, size=3) + # bars for means  
 stat\_summary(fun.data = mean\_se, geom = "errorbar", width = 0.2) + # error bars for SE  
 labs(  
 x = "Lake",  
 y = "Total Length (mm)",  
 fill = "Lake",  
 color="Lake",  
 shape = "Lake"  
 ) +  
 coord\_cartesian(ylim = c(0, 60))+  
 theme\_light() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 legend.position = "right"  
 )



# Conclusion and Scientific Reporting

# Calculate means and standard errors for reporting  
mean\_se\_by\_lake <- sculpin\_df %>%  
 group\_by(lake) %>%  
 summarize(  
 n = n(),  
 mean = mean(length\_mm),  
 sd = sd(length\_mm),  
 se = sd / sqrt(n)  
 )  
  
mean\_se\_by\_lake

# A tibble: 2 × 5  
 lake n mean sd se  
 <chr> <int> <dbl> <dbl> <dbl>  
1 NE 14 37 47.3 10.5 1.72  
2 S 07 73 55.6 12.7 1.48

# Calculate percent difference  
percent\_diff <- abs(diff(mean\_se\_by\_lake$mean)) / min(mean\_se\_by\_lake$mean) \* 100  
percent\_diff

[1] 17.54036

Based on our analysis, we can conclude:

The total length of slimy sculpin fish differs significantly between Lake S 07 and Lake NE 14 (two-sample t-test: t(df) = t\_statistic, p < 0.001). Fish from Lake S 07 were on average (mean\_diff) mm longer than those from Lake NE 14 (mean ± SE: mm).

## 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

respectively; two-sample t-test: t(df) = t\_statistic, p < 0.001). This represents an approximatelypercent\_diff% difference in total length between the two populations.”

For figures, include:

1. A boxplot or mean/SE plot showing the difference
2. Clear labels and scales
3. Sample sizes
4. Statistical test information in the figure caption

A typical caption would read:

Note I would also add the mean and SE of each lake

“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) (two-sample t-test: t(108) = 3.46, p < 0.001).”