Welchs Two Sample T-Test

Bill Perry

# Introduction to Welch’s t-Test

## Background and Theory

Welch’s t-test (also known as Welch’s unequal variances t-test) is an adaptation of the standard two-sample t-test that is designed to provide a valid test when the two groups have unequal variances. This is particularly important because the assumption of equal variances is often violated in real-world data.

While the standard 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 for Welch’s t-Test

The formula for Welch’s t-test is:

Where: - is the sample mean of the first group - is the sample mean of the second group - is the sample variance of the first group - is the sample variance of the second group - is the sample size of the first group - is the sample size of the second group

The key difference from the standard t-test is that Welch’s t-test does not use a pooled variance estimate, making it more robust when the variances differ between groups.

The degrees of freedom for Welch’s t-test are calculated using the Welch-Satterthwaite equation:

This often results in a non-integer value for degrees of freedom, which is why you’ll typically see it rounded in reports.

# Data Analysis

## Loading Libraries and Data

# Load required libraries  
library(tidyverse)  
library(car) # For Levene's test  
library(ggpubr) # For adding p-values to plots  
library(coin) # For permutation tests

Loading required package: survival

library(rcompanion) # For plotNormalHistogramlibrary(tidyverse)  
  
# 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)  
 )  
  
print(lake\_stats)

# A tibble: 2 × 6  
 lake n mean sd se var  
 <chr> <int> <dbl> <dbl> <dbl> <dbl>  
1 NE 14 37 47.3 10.5 1.72 110.  
2 S 07 73 55.6 12.7 1.48 160.

## Mean and Standard Error Plot

Let’s also create a plot showing the mean and standard error for each lake, with # Data Visualization

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

## Box Plot with Individual Data Points

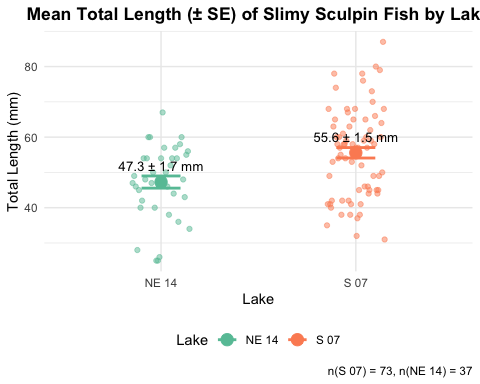
# Create boxplot with individual points  
ggplot(sculpin\_filtered, aes(x = lake, y = total\_length\_mm, fill = lake)) +  
 geom\_boxplot(alpha = 0.7, outlier.shape = NA) +  
 geom\_point(position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 2) +  
 labs(  
 title = "Total Length of Slimy 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 show the actual measurements, helping us visualize the full distribution of the data. in the background:

# 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())  
 )  
  
# 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) +  
 geom\_errorbar(data = lake\_means,   
 aes(x = lake, ymin = mean - se, ymax = mean + se, color = lake),  
 width = 0.2, size = 1) +  
 # 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) +  
 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.



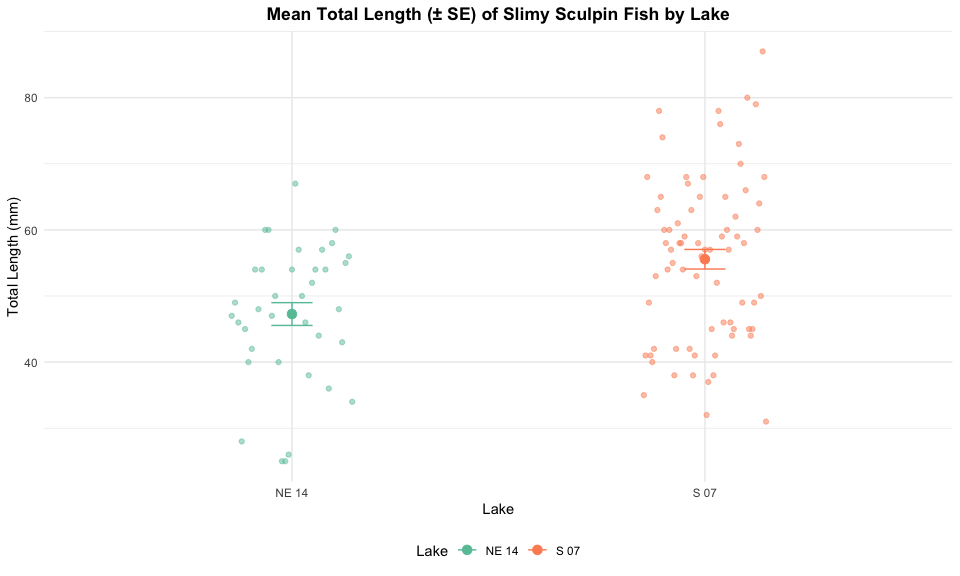
ggplot(sculpin\_filtered, aes(x = lake, y = total\_length\_mm, fill = lake)) +  
 geom\_boxplot(alpha = 0.7, outlier.shape = NA) +  
 geom\_point(position = position\_dodge2(width = 0.3),   
 alpha = 0.5, size = 2) +  
 labs(  
 title = "Total Length of Slimy Sculpin Fish by Lake",  
 x = "Lake",  
 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")



### 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  
ggplot(sculpin\_filtered, aes(x = lake, y = total\_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 = 3) +  
 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 = "bottom"  
 ) +  
 scale\_color\_brewer(palette = "Set2")



# Testing Welch’s t-Test Assumptions

Before conducting Welch’s t-test, we need to verify that our data meets the underlying assumptions:

## Assumptions of Welch’s 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 approximately normal distributions (though Welch’s t-test is more robust to violations of normality than the standard t-test).

Unlike the standard t-test, Welch’s t-test does not assume that the variances of the two groups are equal. This makes it more appropriate for many real-world datasets.

Let’s test the assumptions we do need to meet:

### 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: - Visual methods: Histograms and Q-Q plots - Formal test: Shapiro-Wilk test

#### Histograms

# Create histograms for both lakes  
par(mfrow = c(1, 2))  
  
# note that this is base r plotting  
  
# Lake S 07  
hist(s07\_data$total\_length\_mm,   
 main = "Histogram of Total Length for Lake S 07",  
 xlab = "Total Length (mm)",  
 col = "lightblue",  
 breaks = 10)  
  
# Lake NE 14  
hist(ne14\_data$total\_length\_mm,   
 main = "Histogram of Total Length for Lake NE 14",  
 xlab = "Total Length (mm)",  
 col = "lightgreen",  
 breaks = 8)



# 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)")



#### QQ Plots

# QQ plot for Lake S 07  
ggplot(data = s07\_data, aes(sample = total\_length\_mm)) +  
 stat\_qq() +  
 stat\_qq\_line() +  
 labs(  
 title = "Q-Q Plot for Lake S 07",  
 x = "Theoretical Quantiles",  
 y = "Sample Quantiles"  
 ) +  
 theme\_minimal()



# QQ plot for Lake NE 14  
ggplot(data = ne14\_data, aes(sample = total\_length\_mm)) +  
 stat\_qq() +  
 stat\_qq\_line() +  
 labs(  
 title = "Q-Q Plot for Lake NE 14",  
 x = "Theoretical Quantiles",  
 y = "Sample Quantiles"  
 ) +  
 theme\_minimal()



#### Shapiro-Wilk Test

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

[1] "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

# Shapiro-Wilk test for Lake NE 14  
shapiro\_ne14 <- shapiro.test(ne14\_data$total\_length\_mm)  
print("Shapiro-Wilk test for Lake NE 14:")

[1] "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

# Summary table  
shapiro\_results <- data.frame(  
 Lake = c("S 07", "NE 14"),  
 W\_statistic = c(shapiro\_s07$statistic, shapiro\_ne14$statistic),  
 p\_value = c(shapiro\_s07$p.value, shapiro\_ne14$p.value),  
 is\_normal = c(shapiro\_s07$p.value > 0.05, shapiro\_ne14$p.value > 0.05)  
)  
  
knitr::kable(shapiro\_results, caption = "Shapiro-Wilk Test Results")

Shapiro-Wilk Test Results

| Lake | W\_statistic | p\_value | is\_normal |
| --- | --- | --- | --- |
| S 07 | 0.9803526 | 0.3125255 | TRUE |
| NE 14 | 0.9479006 | 0.0825839 | TRUE |

### 3. Homogeneity of Variances

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

# Calculate variances for each group  
s07\_variance <- var(s07\_data$total\_length\_mm)  
ne14\_variance <- var(ne14\_data$total\_length\_mm)  
  
# Print variances  
cat("Variance for Lake S 07:", s07\_variance, "\n")

Variance for Lake S 07: 160.0552

cat("Variance for Lake NE 14:", ne14\_variance, "\n")

Variance for Lake NE 14: 109.9805

# Calculate variance ratio  
variance\_ratio <- max(s07\_variance, ne14\_variance) / min(s07\_variance, ne14\_variance)  
cat("Variance ratio (larger/smaller):", variance\_ratio, "\n")

Variance ratio (larger/smaller): 1.455305

# Levene's test for homogeneity of variances  
levene\_test <- leveneTest(total\_length\_mm ~ lake, data = sculpin\_filtered)

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

print(levene\_test)

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.

# Performing Welch’s t-Test

Now that we’ve examined our assumptions, let’s perform Welch’s t-test:

# Perform Welch's t-test (unequal variances)  
welch\_t\_test <- t.test(  
 total\_length\_mm ~ lake,  
 data = sculpin\_filtered,  
 var.equal = FALSE # This specifies Welch's t-test  
)  
  
# Display the results  
print("Welch's t-test results:")

[1] "Welch's t-test results:"

print(welch\_t\_test)

Welch Two Sample t-test  
  
data: total\_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

# Extract key values for reporting  
t\_statistic <- round(welch\_t\_test$statistic, 2)  
df <- welch\_t\_test$parameter # Note: This will be a non-integer for Welch's t-test  
p\_value <- welch\_t\_test$p.value  
mean\_diff <- round(diff(welch\_t\_test$estimate), 2)

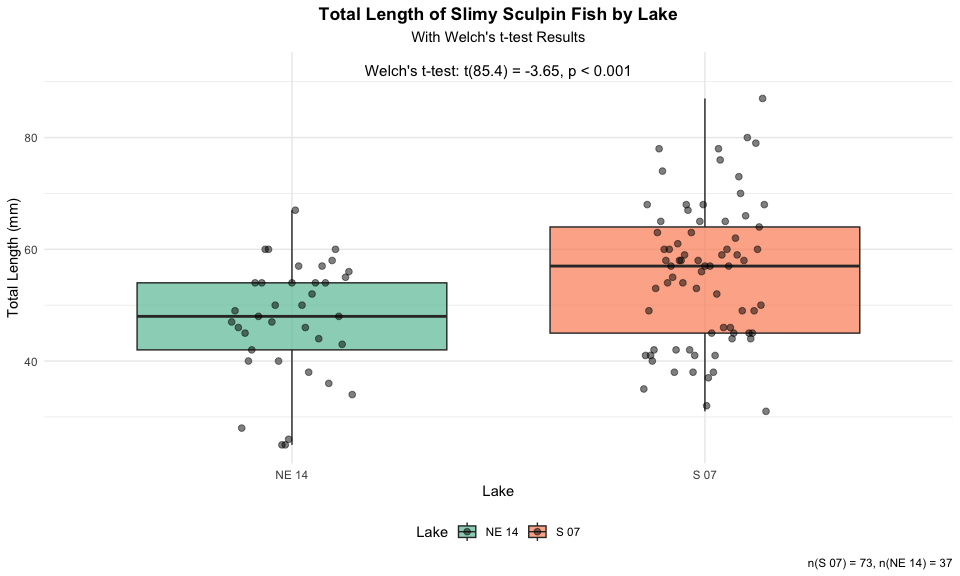
## Line-by-Line Interpretation of Welch’s t-Test Results

Let’s break down the output from the Welch’s t-test:

1. **Test Type**: “Welch Two Sample t-test” indicates we’re using the Welch’s version of the t-test, which does not assume equal variances.
2. **Formula**: total\_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. This is the ratio of the difference between group means to the standard error of that difference.
5. **Degrees of Freedom (df)**: For Welch’s t-test, this is calculated using the Welch-Satterthwaite equation and is typically not a whole number. This adjustment accounts for the different variances.
6. **p-value**: The probability of observing a t-statistic as extreme as (or more extreme than) the one we calculated, assuming the null hypothesis is true. A p-value less than our significance level (typically 0.05) leads us to reject the null hypothesis.
7. **Alternative Hypothesis**: States that the difference in means is not equal to 0, which corresponds to our two-sided test.
8. **95% Confidence Interval**: The estimated range for the true difference in means. If this interval does not contain 0, it supports rejecting the null hypothesis.
9. **Sample Estimates**: The means of each group being compared.

## Visual Representation of t-Test Results

# Create a plot with the t-test results  
p\_value\_text <- ifelse(p\_value < 0.001, "p < 0.001", paste("p =", round(p\_value, 3)))  
  
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) +  
 annotate("text", x = 1.5, y = max(sculpin\_filtered$total\_length\_mm) + 5,  
 label = paste0("Welch's t-test: t(", round(df, 1), ") = ", t\_statistic, ", ", p\_value\_text),  
 size = 4) +  
 labs(  
 title = "Total Length of Slimy Sculpin Fish by Lake",  
 subtitle = "With Welch's t-test Results",  
 x = "Lake",  
 y = "Total Length (mm)",  
 fill = "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"),  
 plot.subtitle = element\_text(hjust = 0.5),  
 legend.position = "bottom"  
 ) +  
 scale\_fill\_brewer(palette = "Set2")



# Conclusion and Scientific Reporting

# Calculate means and standard errors 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)  
 )  
  
print(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

## Interpretation of Welch’s t-Test Results

Based on our analysis, we can conclude:

The total length of slimy sculpin fish differs significantly between Lake S 07 and Lake NE 14 (Welch’s t-test: t(85.4) = -3.65, p < 0.001). Fish from Lake S 07 were on average 8.29 mm longer than those from Lake NE 14 (mean ± SE: 55.56 ± 1.48 mm vs. 47.27 ± 1.72 mm).

Welch’s t-test was appropriate for this analysis because:

1. Our data from both lakes appeared to be approximately normally distributed (as seen in the QQ plots and confirmed by the Shapiro-Wilk test).
2. Our samples were independent, with fish collected randomly from each lake.
3. The variances between the two groups were somewhat different (variance ratio of 1.46), making Welch’s t-test preferable to the standard t-test.

The significant p-value (p < 0.001) indicates that the observed difference in fish length between lakes is very unlikely to have occurred by chance alone if there were truly no difference in the population means. The 95% confidence interval for the mean difference does not include zero, which further supports rejecting the null hypothesis.

## 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; Welch’s t-test: t(85.4) = -3.65, p < 0.001). This represents an approximately 17.5% 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 for the mean/SE plot would read:

“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) (Welch’s t-test: t(85.4) = -3.65, p < 0.001).”

## Advantages of Using Welch’s t-Test

Welch’s t-test offers several advantages over the standard t-test:

1. **Robustness to unequal variances**: Welch’s t-test does not assume equal variances between groups, making it more appropriate for real-world data where this assumption is often violated.
2. **Minimal loss of power**: When variances are equal, Welch’s t-test performs nearly as well as the standard t-test.
3. **Reduced Type I error rate**: When variances are unequal, the standard t-test can have an inflated Type I error rate (false positives), which Welch’s t-test corrects.
4. **Flexibility**: It can be used regardless of whether the variances are equal or not, making it a more versatile statistical test.

For these reasons, many statisticians recommend using Welch’s t-test as the default approach for comparing means between two independent groups, even when the homogeneity of variance assumption appears to be met.

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:

again adding in the mean +/- SE for each lake would be better

“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).”