05\_Class\_Activity

Bill Perry

# In class activity 5:

## What did we do last time in activity 4?

* Created and interpreted frequency distributions with histograms
* Compared lakes using side-by-side histograms
* Explored how sample size affects our view of a population
* Created density plots and calculated probabilities

## Today’s focus:

* Understanding standard normal distributions and z-scores
* Calculating and interpreting standard error
* Creating confidence intervals
* Working with the Student’s t-distribution

# Introduction

In this active learning module, we’ll explore statistical inference using the sculpin fish dataset from Alaska. We’ll focus on:

* Converting raw data to standardized z-scores
* Calculating standard error and confidence intervals
* Understanding the Student’s t-distribution
* Making statistical comparisons between lakes

We’ll use the tidyverse package for data manipulation and visualization, along with patchwork for combining plots.

## Setup

First, let’s load the packages we need and the dataframe:

# Load required packages  
library(tidyverse)  
library(patchwork)  
  
# Read in the data file  
sculpin\_df <- read\_csv("data/sculpin.csv")  
  
# Look at the first few rows  
head(sculpin\_df)

# 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

|  |
| --- |
| Practice Exercise 1: Recreating Our Last Histograms |
| Let’s recreate the basic histogram of fish lengths from our last class. Use the sculpin\_df data frame that’s already loaded.  # Write your code here to create a histogram of fish lengths from Toolik Lake # Remember to use the pipe operator %>% and ggplot with geom\_histogram() |

|  |
| --- |
| Practice Exercise 2: Compare Fish Distributions from Different Lakes |
| Let’s look at what lakes are in our dataframe:  # View the unique lake names unique(sculpin\_df$lake)  [1] "E 01" "E 05" "NE 12" "NE 14" "S 06" "S 07" "Toolik"  Now, select two lakes and create a comparison of their fish length distributions using facet\_grid():  # Your code here to compare fish lengths between two lakes of your choice |

Let’s calculate some basic statistics for Toolik Lake fish:

# Calculate basic statistics for Toolik Lake  
toolik\_stats <- sculpin\_df %>%   
 filter(lake == "Toolik") %>%   
 summarize(  
 mean\_length = mean(total\_length\_mm, na.rm = TRUE),  
 sd\_length = sd(total\_length\_mm, na.rm = TRUE),  
 n = sum(!is.na(total\_length\_mm)),  
 se\_length = sd\_length / sqrt(n)  
 )  
  
# Display the statistics  
toolik\_stats

# A tibble: 1 × 4  
 mean\_length sd\_length n se\_length  
 <dbl> <dbl> <int> <dbl>  
1 51.7 12.0 208 0.834

# Part 1: Understanding the Standard Normal Distribution

## Converting to Z-scores

Z-scores tell us how many standard deviations a data point is from the mean. The formula is:

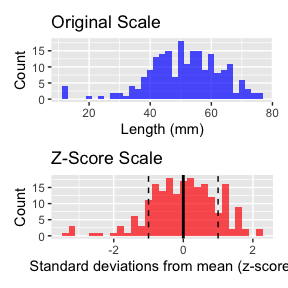
Let’s calculate z-scores for the Toolik Lake fish:

# Add a z-score column to our Toolik Lake data  
toolik\_z <- sculpin\_df %>%  
 filter(lake == "Toolik") %>%  
 mutate(z\_score = (total\_length\_mm - toolik\_stats$mean\_length) / toolik\_stats$sd\_length)  
  
# Display the first few rows with z-scores  
head(toolik\_z)

# A tibble: 6 × 6  
 site lake species total\_length\_mm mass\_g z\_score  
 <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
1 100 Toolik slimy sculpin 12.5 0.0152 -3.26   
2 100 Toolik slimy sculpin 13 0.0104 -3.22   
3 100 Toolik slimy sculpin 74 2.3 1.85   
4 100 Toolik slimy sculpin 42 0.5 -0.806  
5 100 Toolik slimy sculpin 39 0.4 -1.06   
6 100 Toolik slimy sculpin 71 3 1.61

Now, let’s create a histogram of the original data and the z-scores side by side:

# Create histogram of original data  
p1 <- toolik\_z %>%  
 ggplot(aes(total\_length\_mm)) +  
 geom\_histogram(binwidth = 2, fill = "blue", alpha = 0.7) +  
 labs(title = "Original Scale",  
 x = "Length (mm)",  
 y = "Count")  
  
# Create histogram of z-scores  
p2 <- toolik\_z %>%  
 ggplot(aes(z\_score)) +  
 geom\_histogram(binwidth = 0.2, fill = "red", alpha = 0.7) +  
 geom\_vline(xintercept = 0, linewidth = 1) +  
 geom\_vline(xintercept = c(-1, 1), color = "black", linetype = "dashed") +  
 labs(title = "Z-Score Scale",  
 x = "Standard deviations from mean (z-score)",  
 y = "Count")  
  
# Combine plots  
p1 / p2



|  |
| --- |
| Activity 1 |
| Calculate the z-score for a fish with a length of 25mm from Toolik Lake. Is this fish length common or unusual?  # Calculate the z-score for a 25mm fish fish\_length <- 25 z\_score\_25mm <- (fish\_length - toolik\_stats$mean\_length) / toolik\_stats$sd\_length  # Print the result z\_score\_25mm  [1] -2.220007  # Determine if this is common or unusual # A z-score beyond -2 or 2 is generally considered unusual |

## Understanding Areas Under the Standard Normal Curve

The standard normal distribution has known probabilities associated with z-scores. We can calculate these probabilities in R:

# Calculate probability of z-score less than 1  
pnorm(1) # This gives us the area under the standard normal curve to the left of z=1

[1] 0.8413447

# Calculate probability of z-score between -1 and 1  
pnorm(1) - pnorm(-1) # This is approximately 68% (the 68-95-99.7 rule)

[1] 0.6826895

|  |
| --- |
| Activity 2 |
| Using the pnorm() function, answer: 1. What proportion of fish in Toolik Lake are shorter than 40mm?  # First, convert 40mm to a z-score z\_score\_40mm <- (40 - toolik\_stats$mean\_length) / toolik\_stats$sd\_length z\_score\_40mm  [1] -0.9725651  # Then, find the proportion of fish shorter than 40mm proportion\_less\_than\_40mm <- pnorm(z\_score\_40mm) proportion\_less\_than\_40mm  [1] 0.1653847 |

# Part 2: Standard Error and Sampling Distributions

## Taking Multiple Samples

When we take multiple samples from a population, each sample gives a slightly different estimate of the population mean. Let’s see this in action:

# Set seed for reproducibility  
set.seed(123)  
  
# Take a small sample  
small\_sample <- sculpin\_df %>%  
 filter(lake == "Toolik") %>%  
 sample\_n(10)  
  
# Calculate the mean of this sample  
small\_sample\_mean <- mean(small\_sample$total\_length\_mm, na.rm = TRUE)  
  
# can see the difference using this formula  
abs(small\_sample\_mean - toolik\_stats$mean\_length)

[1] 1.26614

# if you ever want to get fancier you can do the following..  
# Print the result  
cat("Sample mean (n=10):", small\_sample\_mean, "mm\n")

Sample mean (n=10): 50.42857 mm

cat("Population mean:", toolik\_stats$mean\_length, "mm\n")

Population mean: 51.69471 mm

cat("Difference:", abs(small\_sample\_mean - toolik\_stats$mean\_length), "mm\n")

Difference: 1.26614 mm

## Code to play with sample size and replicates

Let’s take multiple small samples and see how their means vary:

# Function to generate sample means with user-specified number of samples  
# this is really advanced and we will get into this more later on  
# it is a way to make your own code to run a set of tasks  
# this one takes the subsample and averages them  
# Function to generate sample means with user-specified number of samples  
generate\_sample\_means <- function(num\_samples, sample\_size = 10) {  
 # Create a dataframe of sample means  
 sample\_means\_df <- tibble(  
 sample\_number = 1:num\_samples,  
 sample\_means = sapply(1:num\_samples, function(x) {  
 sculpin\_df %>%  
 filter(lake == "Toolik") %>%  
 sample\_n(sample\_size) %>%  
 summarize(mean\_length = mean(total\_length\_mm, na.rm = TRUE)) %>%  
 pull(mean\_length)  
 })  
 )  
   
 # Return the dataframe  
 return(sample\_means\_df)  
}  
  
# Example usage  
sample\_means\_dataframe <- generate\_sample\_means(num\_samples = 1000, sample\_size = 10)  
  
# Print the dataframe  
print(sample\_means\_dataframe)

# A tibble: 1,000 × 2  
 sample\_number sample\_means  
 <int> <dbl>  
 1 1 48.2  
 2 2 52.9  
 3 3 50.1  
 4 4 46.3  
 5 5 49.2  
 6 6 54   
 7 7 55.4  
 8 8 52.5  
 9 9 51   
10 10 53.4  
# ℹ 990 more rows

# Print summary statistics of the sample means  
summary(sample\_means\_dataframe$sample\_means)

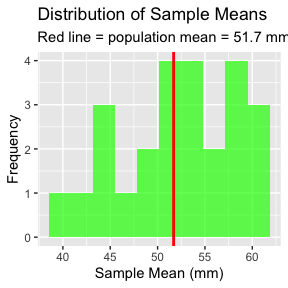
Min. 1st Qu. Median Mean 3rd Qu. Max.   
 33.00 48.57 52.00 51.47 54.38 66.50

## Run this code to take the samples

# Example usage:  
means\_result <- generate\_sample\_means(num\_samples = 25, sample\_size = 5)

## Run this code to see the plot

means\_plot <- ggplot(means\_result, aes(x = sample\_means)) +  
 geom\_histogram(bins = 10, fill = "green", alpha = 0.7) +  
 geom\_vline(xintercept = toolik\_stats$mean\_length, color = "red", linewidth = 1) +  
 labs(title = paste("Distribution of Sample Means"),  
 subtitle = paste("Red line = population mean =", round(toolik\_stats$mean\_length, 1), "mm"),  
 x = "Sample Mean (mm)",  
 y = "Frequency")   
means\_plot



## Run this code to see the dataframe

means\_result # to access the dataframe of sample means

# A tibble: 25 × 2  
 sample\_number sample\_means  
 <int> <dbl>  
 1 1 61.5  
 2 2 54.3  
 3 3 57   
 4 4 57.8  
 5 5 59.7  
 6 6 52.8  
 7 7 43.7  
 8 8 49.4  
 9 9 53.2  
10 10 51.7  
# ℹ 15 more rows

## Standard Error

The standard error of the mean tells us how much we expect sample means to vary:

# Calculate the standard error from our samples  
sample\_se <- sd(means\_result$sample\_means)  
sample\_se

[1] 5.960243

# Calculate the theoretical standard error  
theoretical\_se <- toolik\_stats$sd\_length / sqrt(toolik\_stats$n)  
theoretical\_se

[1] 0.8337564

# Display both values  
cat("Standard Error from samples:", round(sample\_se, 2), "mm\n")

Standard Error from samples: 5.96 mm

cat("Standard Error from populaiton:", round(theoretical\_se, 2), "mm\n")

Standard Error from populaiton: 0.83 mm

|  |
| --- |
| Activity 3 |
| Let’s see how sample size affects the standard error. Run the code below to take samples of size 5 and size 20.  # Function to take multiple samples of a given size take\_samples <- function(sample\_size, n\_samples) {  replicate(n\_samples, {  sculpin\_df %>%  filter(lake == "Toolik") %>%  sample\_n(sample\_size) %>%  summarize(mean\_length = mean(total\_length\_mm, na.rm = TRUE)) %>%  pull(mean\_length)  }) }  # Take 10 samples of size 5 and calculate SE # take\_samples(size of sample, number of samples) sample\_means\_low <- take\_samples(5, 20) se\_low <- sd(sample\_means\_low)  # Take samples of size 20 and calculate SE sample\_means\_high <- take\_samples(25, 20) se\_high <- sd(sample\_means\_high)  # Print results cat("SE with low number and 20 replicates:", round(se\_low, 2), "mm\n")  SE with low number and 20 replicates: 5.56 mm  cat("SE with high number and 20 replicates:", round(se\_high, 2), "mm\n")  SE with high number and 20 replicates: 1.5 mm  # How does SE change as sample size increases? |

# Part 3: Confidence Intervals

## Calculating Confidence Intervals

A confidence interval gives us a range of values that likely contains the true population mean. For a 95% confidence interval:

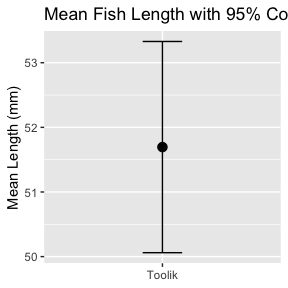
Let’s calculate the 95% confidence interval for the mean length of Toolik Lake fish:

# Calculate 95% confidence interval using z (normal distribution)  
ci\_lower <- toolik\_stats$mean\_length - 1.96 \* toolik\_stats$se\_length  
ci\_upper <- toolik\_stats$mean\_length + 1.96 \* toolik\_stats$se\_length  
  
# Display the confidence interval  
cat("95% Confidence Interval:",   
 round(ci\_lower, 1), "to", round(ci\_upper, 1), "mm\n")

95% Confidence Interval: 50.1 to 53.3 mm

## Visualizing Confidence Intervals

# Create a data frame with our statistics  
toolik\_ci <- data.frame(  
 lake = "Toolik",  
 mean = toolik\_stats$mean\_length,  
 lower = ci\_lower,  
 upper = ci\_upper  
)  
  
# Plot the mean with error bars showing the 95% CI  
ggplot(toolik\_ci, aes(x = lake, y = mean)) +  
 geom\_point(size = 3) +  
 geom\_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +  
 labs(title = "Mean Fish Length with 95% Confidence Interval",  
 x = NULL,  
 y = "Mean Length (mm)")



|  |
| --- |
| Activity 4 |
| Calculate and visualize the 95% confidence interval for another lake of your choice.  # Choose another lake other\_lake <- "E 01" # You can change this to any lake in the dataset  # Calculate statistics for your chosen lake other\_lake\_stats <- sculpin\_df %>%  filter(lake == other\_lake) %>%  summarize(  mean\_length = mean(total\_length\_mm, na.rm = TRUE),  sd\_length = sd(total\_length\_mm, na.rm = TRUE),  n = sum(!is.na(total\_length\_mm)),  se\_length = sd\_length / sqrt(n)  )  # Display the statistics other\_lake\_stats  # A tibble: 1 × 4  mean\_length sd\_length n se\_length  <dbl> <dbl> <int> <dbl> 1 58.2 15.3 79 1.72  # Calculate 95% confidence interval other\_ci\_lower <- other\_lake\_stats$mean\_length - 1.96 \* other\_lake\_stats$se\_length other\_ci\_upper <- other\_lake\_stats$mean\_length + 1.96 \* other\_lake\_stats$se\_length  # Display the confidence interval  cat("\n95% Confidence Interval for", other\_lake, ":",   round(other\_ci\_lower, 1), "to", round(other\_ci\_upper, 1), "mm\n")  95% Confidence Interval for E 01 : 54.8 to 61.6 mm  # Create a visualization # Add your code here to create a similar errorbar plot |

# Part 4: Student’s t-Distribution

When our sample size is small (n < 30), we should use the Student’s t-distribution instead of the normal distribution to calculate confidence intervals:

# Calculate 95% CI using t-distribution  
n <- toolik\_stats$n  
t\_critical <- qt(0.975, df = n - 1) # Two-tailed, so 0.975 for 95% CI  
  
t\_ci\_lower <- toolik\_stats$mean\_length - t\_critical \* toolik\_stats$se\_length  
t\_ci\_upper <- toolik\_stats$mean\_length + t\_critical \* toolik\_stats$se\_length  
  
# Compare with z-based CI  
cat("95% CI using z:", round(ci\_lower, 1), "to", round(ci\_upper, 1), "mm\n")

95% CI using z: 50.1 to 53.3 mm

cat("95% CI using t:", round(t\_ci\_lower, 1), "to", round(t\_ci\_upper, 1), "mm\n")

95% CI using t: 50.1 to 53.3 mm

|  |
| --- |
| Activity 5 |
| Let’s see how t-critical values change with sample size. The smaller the sample size, the larger the t-critical value.  # Create a table of t-critical values for different sample sizes sample\_sizes <- c(5, 10, 30, 100) t\_critical\_values <- qt(0.975, df = sample\_sizes - 1)  # Create a data frame t\_critical\_df <- data.frame(  sample\_size = sample\_sizes,  t\_critical = t\_critical\_values )  # Display the table t\_critical\_df  sample\_size t\_critical 1 5 2.776445 2 10 2.262157 3 30 2.045230 4 100 1.984217  # The t-critical value for 95% confidence is 1.96 # Which sample sizes have t-critical values close to 1.96? |

# Part 5: Testing Hypotheses

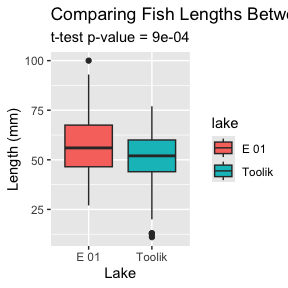
## Two-Sample t-test

Now let’s compare fish lengths between two lakes:

# Get data for Toolik and E 01 lakes  
# what is this doing?  
# look in ENV tab - it is making a vector that is running a t-test  
toolik\_fish <- sculpin\_df %>%   
 filter(lake == "Toolik", !is.na(total\_length\_mm)) %>%  
 pull(total\_length\_mm)  
  
e01\_fish <- sculpin\_df %>%   
 filter(lake == "E 01", !is.na(total\_length\_mm)) %>%  
 pull(total\_length\_mm)  
  
# Perform a two-sample t-test  
t\_test\_result <- t.test(toolik\_fish, e01\_fish)  
  
# Display the result  
t\_test\_result

Welch Two Sample t-test  
  
data: toolik\_fish and e01\_fish  
t = -3.4051, df = 116.36, p-value = 0.0009082  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -10.313036 -2.727921  
sample estimates:  
mean of x mean of y   
 51.69471 58.21519

# Create a boxplot comparison  
sculpin\_df %>%  
 filter(lake %in% c("Toolik", "E 01"), !is.na(total\_length\_mm)) %>%  
 ggplot(aes(x = lake, y = total\_length\_mm, fill = lake)) +  
 geom\_boxplot() +  
 labs(title = "Comparing Fish Lengths Between Lakes",  
 subtitle = paste("t-test p-value =", round(t\_test\_result$p.value, 4)),  
 x = "Lake",  
 y = "Length (mm)")



|  |
| --- |
| Activity 6 |
| Run a t-test to compare fish lengths between two different lakes of your choice.  # Choose two lakes to compare lake1 <- "S 06" # You can change this lake2 <- "NE 12" # You can change this  # Get the data for both lakes lake1\_fish <- sculpin\_df %>%   filter(lake == lake1, !is.na(total\_length\_mm)) %>%  pull(total\_length\_mm)  lake2\_fish <- sculpin\_df %>%   filter(lake == lake2, !is.na(total\_length\_mm)) %>%  pull(total\_length\_mm)  # Run the t-test my\_t\_test <- t.test(lake1\_fish, lake2\_fish)  # Display the result my\_t\_test  Welch Two Sample t-test  data: lake1\_fish and lake2\_fish t = 2.8345, df = 309.86, p-value = 0.004891 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval:  1.280581 7.094166 sample estimates: mean of x mean of y   54.01515 49.82778  # Interpret your results: # - What is the p-value? # - Is there a significant difference between the lakes at α = 0.05? # - Which lake has longer fish on average? |

# Part 6: Guided Challenges if time

Now it’s your turn to apply what you’ve learned. Work with your partner to complete these challenges:

|  |
| --- |
| Challenge 1 |
| Create boxplots of fish lengths for two lakes of your choice. Include the sample size in the labels. |

# Your code here for Challenge 1

|  |
| --- |
| Challenge 2 |
| Calculate and compare the standard errors for fish lengths in three different lakes. Which lake has the smallest standard error and why? |

# Your code here for Challenge 2

|  |
| --- |
| Challenge 3 |
| Create a bar plot showing the mean fish lengths for all lakes with at least 20 fish, and add error bars showing the 95% confidence intervals. |

# Your code here for Challenge 3

# Reflection Questions

After completing the activities, discuss these questions with your group:

1. How does sample size affect our confidence in estimating the population mean?
2. Why is the t-distribution more appropriate than the normal distribution when working with small samples?
3. When comparing two populations, what can we learn from looking at confidence intervals versus performing a t-test?
4. How would you explain the concept of statistical significance to someone who has never taken a statistics course?