Lecture 03

Bill Perry

# **Lecture 2: Review of data and graphing**

* We covered
* How to design a well-organized project
* How to implement good naming conventions
  + Controlled vocabulary
  + Including units in names
* Create and use metadata effectively
* Build tidy, well-structured spreadsheets
* Understand data repositories
* Create effective visualizations with ggplot2

These are variables - do you know what they mean?

TGW - yep its a thing

ODO - what do you think it is?

NO3 - what is it? Are you sure? Why might you get in legal trouble if you used this?



# **Lecture 3:** Descriptive Statistics and Uncerrtainty in R and Tidyverse

## The objectives:

* Understand why statistics is vital in biology
* Distinguish between different types of biological variables
* Learn about accuracy, precision, and bias in measurements
* Calculate and interpret measures of central tendency (mean, median, geometric mean)
* Calculate and interpret measures of spread (standard deviation, variance, IQR)
* Understand data transformations for skewed distributions
* Visualize descriptive statistics for our data
* Learn how to handle uncertainty in our data

We’ll use a dataset on grayling fish from two different lakes to explore these concepts..





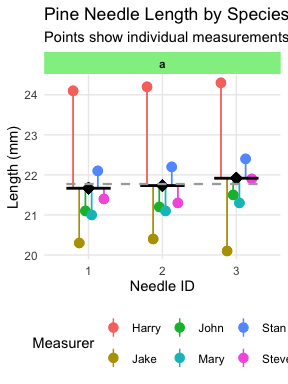
# Lecture 3: Why Statistics is Vital in Biology

Biology is fundamentally different from fields like physics in that:

* Most biological phenomena are **probabilistic** rather than **deterministic**
  + Responses occur with some characteristic probability, not with certainty
* All biological material varies, which is essential for evolution (recall Darwin’s postulates):
  + Variation exists within populations
  + Some variation is heritable
  + Some heritable variation affects survival/reproduction
* Environmental conditions (in nature, lab, or greenhouse) always vary
* Measurements include error
* Multiple unmeasured causal factors influence nearly all biological systems

Statistics helps us understand biological processes in this variable world by:

1. Condensing variation into summary form (Descriptive statistics)
2. Testing whether observations are consistent with predictions (Inferential statistics)



|  |
| --- |
| Practice Exercise 1: Can you do this for the pine data we have collected? |
| 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 read in the file # How do you examine the data - what are the ways you think and lets try it! |

# Lecture 3: Populations and Samples

Before we dive into descriptive statistics, let’s clarify some fundamental concepts:

* **Population**: The entire group of things under consideration; the group for which answers obtained from measurements and statistical analysis are pertinent.
* **Sample**: A subset of the population that is actually measured.
* **Sample unit**: The individual thing drawn from the population.

Types of populations: -

* **Observational population**: Usually finite but may be very large (e.g., head width of all corn earworms in a field) -
* **Experimental population**: Often conceptually infinite (e.g., all possible goldenrod plants that could receive a specific fertilizer treatment)

Sampling involves

* **inference** - generalizing from what is observed in the sample to what is present in the population.
* Valid inference requires **random sampling**.



# Lecture 3: Parameters vs. Statistics

It’s important to distinguish between:

* **Parameters**: True numerical values for a population (usually denoted by Greek letters)
* **Statistics**: Estimates of parameters based on samples (usually denoted by Roman letters)

For example:

* Population mean (μ) is estimated by sample mean (Y̅)
* Population standard deviation (σ) is estimated by sample standard deviation (s)

The standard deviation formula above includes n-1 in the denominator (rather than n) to provide an unbiased estimate of the population parameter.



# Lecture 3: Kinds of Biological Variables

Understanding the type of variable you’re working with is essential for selecting appropriate statistics:

### Measurement or Quantitative Variables

* **Continuous**: Any value between extremes of scale is possible (e.g., mass, length)
* **Discrete (meristic)**: Only fixed values (usually integers) between extremes are possible (e.g., bristle number, egg count)

### Rank Variables (Ordinal)

* Assign only order, not quantity
* Nothing implied about relative distance between values

### Categorical Variables (Qualitative)

* No quantitative information (e.g., male/female, living/dead)
* Some are simplifications of quantitative variables (e.g., color instead of wavelength)

asdfsd

# Lecture 3: Kinds of Biological Variables

Derived Variables

* **Percentages, Proportions**: Ratio of some component to total
* **Ratios**: Relation of two variables
* **Rates**: Quantity per unit (time, mass, etc.)
* **Indices**: More complex derived variables (e.g., condition index)

Let’s explore our grayling fish dataset and identify the types of variables it contains.

asf

|  |
| --- |
| Practice Exercise 2: Can you do this for the pine data we have collected? |
| Let’s examine the different data and determine what they are?  # Write your code here to read in the file # How do you examine the data - what are the ways you think and lets try it!  # Load the grayling data grayling\_df <- read\_csv("data/gray\_I3\_I8.csv")  Rows: 168 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.  # Take a look at the first few rows head(grayling\_df)  # A tibble: 6 × 5  site lake species length\_mm mass\_g  <dbl> <chr> <chr> <dbl> <dbl> 1 113 I3 arctic grayling 266 135 2 113 I3 arctic grayling 290 185 3 113 I3 arctic grayling 262 145 4 113 I3 arctic grayling 275 160 5 113 I3 arctic grayling 240 105 6 113 I3 arctic grayling 265 145  # Get a summary of the dataset glimpse(grayling\_df)  Rows: 168 Columns: 5 $ site <dbl> 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, … $ lake <chr> "I3", "I3", "I3", "I3", "I3", "I3", "I3", "I3", "I3", "I3", … $ species <chr> "arctic grayling", "arctic grayling", "arctic grayling", "ar… $ length\_mm <dbl> 266, 290, 262, 275, 240, 265, 265, 253, 246, 203, 289, 239, … $ mass\_g <dbl> 135, 185, 145, 160, 105, 145, 150, 130, 130, 71, 179, 108, 1… |

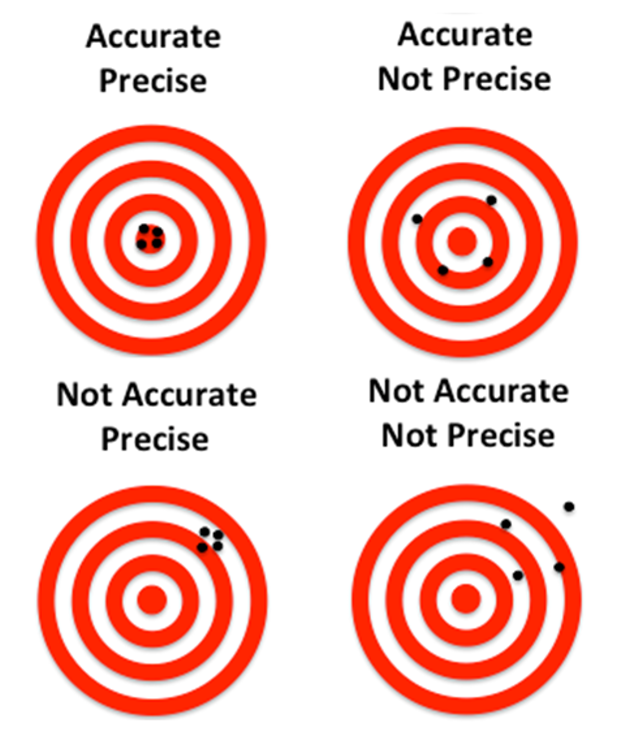
# Lecture 3: Accuracy, Precision, and Bias

When taking biological measurements, understanding measurement quality is essential:

* **Accuracy**: Closeness of measured value to true value
* **Precision**: Closeness of repeated measurements to each other (repeatability)
* **Bias**: Systematic departure from the true value

Accuracy is a function of both precision and bias. For statisticians, bias is usually a more serious problem than low precision because:

* It’s harder to detect (true value usually unknown)
* Low precision can be compensated for by increased sample size



asdfsd

|  |
| --- |
| Practice Exercise 1: What are potential sources of error in pine needles or fish? |
| For our grayling data, potential sources of measurement error might include:   * Precision issues:   + Variations in how fish are measured (e.g., slightly bent fish) * Bias issues:   + Systematic underestimation of length if measurements aren’t taken from the true tip of the snout to the end of the tail * Accuracy issues? what could they be? |

# Lecture 3: Measures of Central Tendency

The two most common measures of central tendency are the **mean** and the **median**.

The Arithmetic Mean The arithmetic mean is the average of a set of measurements:

Where:

* represents each individual measurement
* is the total number of observations

stuff

# Calculate mean length of all fish  
mean\_length <- mean(grayling\_df$length\_mm)  
cat("Mean length of all fish:", round(mean\_length, 1), "mm\n")

Mean length of all fish: 324.5 mm

# Calculate mean by lake  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(mean\_length = mean(length\_mm, na.rm=TRUE)) %>%  
 kable(caption = "Mean length by lake", digits = 1)

Mean length by lake

| lake | mean\_length |
| --- | --- |
| I3 | 265.6 |
| I8 | 362.6 |

# Lecture 3: Measures of Central Tendency

The Median

* The median is the middle value of a sorted dataset.
* If there is an even number of observations, it’s the average of the two middle values.

# Calculate median length of all fish  
median\_length <- median(grayling\_df$length\_mm)  
cat("Median length of all fish:", median\_length, "mm\n")

Median length of all fish: 324.5 mm

# Calculate median by lake  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(median\_length = median(length\_mm)) %>%  
 kable(caption = "Median length by lake", digits = 1)

Median length by lake

| lake | median\_length |
| --- | --- |
| I3 | 266 |
| I8 | 373 |

# Lecture 3: Measures of Spread

The spread of a distribution tells us how variable the measurements are.

### Variance and Standard Deviation

The variance is

The standard deviation is the square root of variance

* measures how far observations typically are from the mean and are in the units of the mean:

# Calculate standard deviation of fish length  
var\_length <- var(grayling\_df$length\_mm)  
sd\_length <- sd(grayling\_df$length\_mm)  
  
  
cat("Variance of length:", round(var\_length, 1), "mm²\n")

Variance of length: 4225.9 mm²

cat("Standard deviation of length:", round(sd\_length, 1), "mm\n")

Standard deviation of length: 65 mm

# Calculate by lake  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 var\_length = var(length\_mm),   
 sd\_length = sd(length\_mm)  
   
 ) %>%  
 kable(caption = "Standard deviation and variance by lake", digits = 1)

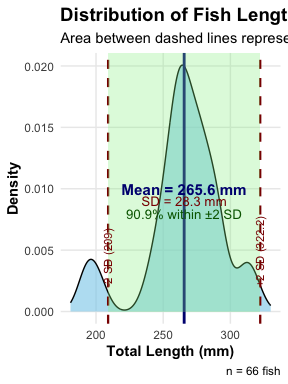
Standard deviation and variance by lake

| lake | var\_length | sd\_length |
| --- | --- | --- |
| I3 | 801.1 | 28.3 |
| I8 | 2739.4 | 52.3 |

# Lecture 3: Measures of Spread

The area under the curve of a bell shaped curve within + and - 2 Standard deviations on each side includes about 95% of the data

# Read the data  
fish\_data <- read.csv("data/gray\_I3\_I8.csv")  
  
# Filter data for I8 lake  
i3\_data <- fish\_data %>%   
 filter(lake == "I3")  
  
# Calculate statistics  
mean\_length <- mean(i3\_data$length\_mm)  
sd\_length <- sd(i3\_data$length\_mm)  
  
# Calculate the bounds for standard deviations  
minus\_2sd <- mean\_length - (2 \* sd\_length)  
plus\_2sd <- mean\_length + (2 \* sd\_length)  
  
# Calculate percentage of data within 2 SD  
percent\_within\_2sd <- 100 \* mean(  
 i3\_data$length\_mm >= minus\_2sd &   
 i3\_data$length\_mm <= plus\_2sd  
)  
  
# Create the plot  
ggplot(i3\_data, aes(x = length\_mm)) +  
 # Add density curve  
 geom\_density(fill = "skyblue", alpha = 0.6) +  
   
 # Add vertical line for mean  
 geom\_vline(xintercept = mean\_length, color = "navy", linewidth = 1) +  
   
 # Add vertical lines for +/- 2 SD  
 geom\_vline(xintercept = minus\_2sd, color = "darkred", linewidth = 0.8, linetype = "dashed") +  
 geom\_vline(xintercept = plus\_2sd, color = "darkred", linewidth = 0.8, linetype = "dashed") +  
   
 # Highlight area within 2 SD  
 annotate("rect",   
 xmin = minus\_2sd, xmax = plus\_2sd,   
 ymin = 0, ymax = Inf,   
 fill = "lightgreen", alpha = 0.3) +  
   
 # Add annotations  
 annotate("text",   
 x = mean\_length, y = 0.010,  
 label = paste0("Mean = ", round(mean\_length, 1), " mm"),  
 color = "navy", fontface = "bold", size = 4) +  
   
 annotate("text",   
 x = mean\_length, y = 0.009,  
 label = paste0("SD = ", round(sd\_length, 1), " mm"),  
 color = "darkred", size = 3.5) +  
   
 annotate("text",   
 x = mean\_length, y = 0.008,  
 label = paste0(round(percent\_within\_2sd, 1), "% within ±2 SD"),  
 color = "darkgreen", size = 3.5) +  
   
 # Add labels for SD boundaries  
 annotate("text",   
 x = minus\_2sd, y = 0.002,  
 label = paste0("-2 SD (", round(minus\_2sd, 1), ")"),  
 color = "darkred", angle = 90, hjust = 0, size = 3) +  
   
 annotate("text",   
 x = plus\_2sd, y = 0.002,  
 label = paste0("+2 SD (", round(plus\_2sd, 1), ")"),  
 color = "darkred", angle = 90, hjust = 0, size = 3) +  
   
 # Add title and labels  
 labs(  
 title = "Distribution of Fish Lengths in i3 Lake",  
 subtitle = "Area between dashed lines represents ±2 standard deviations from the mean",  
 x = "Total Length (mm)",  
 y = "Density",  
 caption = paste0("n = ", nrow(i3\_data), " fish")  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(face = "bold", size = 14),  
 plot.subtitle = element\_text(size = 11),  
 axis.title = element\_text(face = "bold"),  
 panel.grid.minor = element\_blank()  
 ) +  
 # Set x-axis limits to show the full range plus a bit of padding  
 xlim(min(i3\_data$length\_mm) - 10, max(i3\_data$length\_mm) + 10)



# Print statistics to console  
cat("i3 Lake Fish Length Summary:\n")

i3 Lake Fish Length Summary:

cat("Number of fish:", nrow(i3\_data), "\n")

Number of fish: 66

cat("Mean length:", round(mean\_length, 2), "mm\n")

Mean length: 265.61 mm

cat("Standard Deviation:", round(sd\_length, 2), "mm\n")

Standard Deviation: 28.3 mm

cat("Range for ±2 SD:", round(minus\_2sd, 2), "to", round(plus\_2sd, 2), "mm\n")

Range for ±2 SD: 209 to 322.21 mm

cat("Percentage within ±2 SD:", round(percent\_within\_2sd, 2), "%\n")

Percentage within ±2 SD: 90.91 %

# Lecture 3: Coefficient of Variation

The coefficient of variation (CV) expresses the standard deviation as a percentage of the mean:

This is useful for comparing the variability of measurements with different units or vastly different scales.

stuff

# Calculate coefficient of variation  
cv\_length <- sd\_length / mean\_length \* 100  
cat("Coefficient of variation:", round(cv\_length, 1), "%\n")

Coefficient of variation: 10.7 %

# Calculate by lake  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 cv\_length = sd(length\_mm) / mean(length\_mm) \* 100  
 ) %>%  
 kable(caption = "Coefficient of variation by lake", digits = 1)

Coefficient of variation by lake

| lake | cv\_length |
| --- | --- |
| I3 | 10.7 |
| I8 | 14.4 |

# Lecture 3: Interquartile Range

The interquartile range (IQR) is the range of the middle 50% of the data:

Where is the first quartile (25th percentile) and is the third quartile (75th percentile).

stuff

# Calculate quartiles and IQR  
q1\_length <- quantile(grayling\_df$length\_mm, 0.25)  
q3\_length <- quantile(grayling\_df$length\_mm, 0.75)  
iqr\_length <- IQR(grayling\_df$length\_mm)  
  
cat("First quartile:", q1\_length, "mm\n")

First quartile: 270.75 mm

cat("Third quartile:", q3\_length, "mm\n")

Third quartile: 377 mm

cat("Interquartile range:", iqr\_length, "mm\n")

Interquartile range: 106.25 mm

# Calculate by lake  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 q1 = quantile(length\_mm, 0.25),  
 q3 = quantile(length\_mm, 0.75),  
 iqr = IQR(length\_mm)  
 ) %>%  
 kable(caption = "Quartiles and IQR by lake", digits = 1)

Quartiles and IQR by lake

| lake | q1 | q3 | iqr |
| --- | --- | --- | --- |
| I3 | 256 | 280 | 24 |
| I8 | 340 | 401 | 61 |

# Lecture 3: Data Transformations for Skewed Distributions

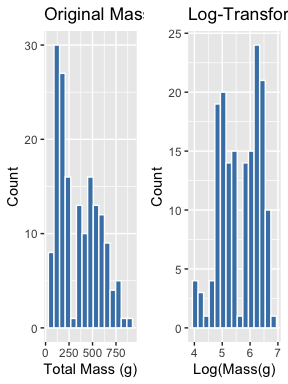
Biological data are often skewed (asymmetrical), which can make the arithmetic mean less representative of central tendency. Data transformations can help address this issue.

### Logarithmic Transformation

The logarithmic transformation is one of the most common for right-skewed biological data:

When data are log-normally distributed, the geometric mean often provides a better measure of central tendency than the arithmetic mean.

# Add log-transformed length to our dataset  
grayling\_df <- grayling\_df %>%  
 mutate(log\_mass = log(mass\_g))  
  
# Compare original and log-transformed distributions  
p1 <- ggplot(grayling\_df, aes(x = mass\_g)) +  
 geom\_histogram(bins = 15, fill = "steelblue", color = "white") +  
 labs(  
 title = "Original Mass Distribution",  
 x = "Total Mass (g)",  
 y = "Count"  
 )  
  
p2 <- ggplot(grayling\_df, aes(x = log\_mass)) +  
 geom\_histogram(bins = 15, fill = "steelblue", color = "white") +  
 labs(  
 title = "Log-Transformed Mass Distribution",  
 x = "Log(Mass(g)",  
 y = "Count"  
 )  
  
# Display side by side  
gridExtra::grid.arrange(p1, p2, ncol = 2)



# Compare means on original and transformed scales  
mean\_log\_mass <- mean(grayling\_df$log\_mass)  
back\_transformed\_mean <- exp(mean\_log\_mass)  
  
cat("Arithmetic mean of original data:", round(mean\_length, 1), "mm\n")

Arithmetic mean of original data: 265.6 mm

cat("Geometric mean (back-transformed mean of logs):", round(back\_transformed\_mean, 1), "mm\n")

Geometric mean (back-transformed mean of logs): NA mm

# Lecture 3: When to Use Transformations

* **Log transformation**: When data are right-skewed or follow multiplicative rather than additive processes
* **Square root transformation**: For count data or data where variance increases with the mean
* **Inverse transformation**: For strongly right-skewed data
* **Arcsine square root transformation**: For proportions or percentages (though logistic regression is often preferred now)

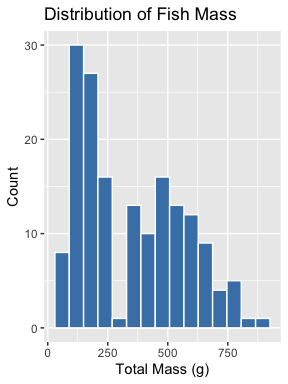
s

# Lecture 3: Visualizing Distributions

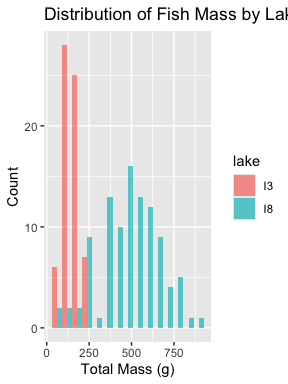
Histograms

Histograms show the frequency distribution of our data.

# Create a histogram  
ggplot(grayling\_df, aes(x = mass\_g)) +  
 geom\_histogram(bins = 15, fill = "steelblue", color = "white") +  
 labs(  
 title = "Distribution of Fish Mass",  
 x = "Total Mass (g)",  
 y = "Count"  
 )



# Histograms by lake  
ggplot(grayling\_df, aes(x = mass\_g, fill = lake)) +  
 geom\_histogram(bins = 15, position = "dodge", alpha = 0.7) +  
 labs(  
 title = "Distribution of Fish Mass by Lake",  
 x = "Total Mass (g)",  
 y = "Count"  
 )

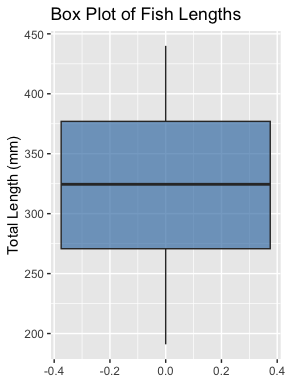


# Lecture 3: Visualizing Distributions

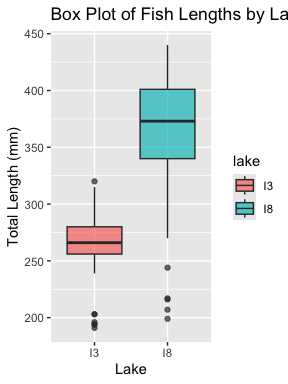
Box Plots

Box plots show the median, quartiles, and potential outliers.

# Create a box plot  
ggplot(grayling\_df, aes(y = length\_mm)) +  
 geom\_boxplot(fill = "steelblue", alpha = 0.7) +  
 labs(  
 title = "Box Plot of Fish Lengths",  
 y = "Total Length (mm)"  
 )



# Box plot by lake  
ggplot(grayling\_df, aes(x = lake, y = length\_mm, fill = lake)) +  
 geom\_boxplot(alpha = 0.7) +  
 labs(  
 title = "Box Plot of Fish Lengths by Lake",  
 x = "Lake",  
 y = "Total Length (mm)"  
 )



# Lecture 3: Comparing Mean vs. Median

The mean and median measure different aspects of a distribution:

Mean: Center of gravity of the distribution

Median: Middle value of the data

When a distribution is symmetric, the mean and median are similar. When it’s skewed or has outliers, they can differ significantly.

stuff

# Calculate summary statistics by lake  
stats\_by\_lake <- grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 mean = mean(length\_mm),  
 median = median(length\_mm),  
 sd = sd(length\_mm),  
 iqr = IQR(length\_mm),  
 skewness = moments::skewness(length\_mm)  
 )  
  
# Display the results  
kable(stats\_by\_lake, caption = "Comparison of Mean and Median by Lake", digits = 1)

Comparison of Mean and Median by Lake

| lake | mean | median | sd | iqr | skewness |
| --- | --- | --- | --- | --- | --- |
| I3 | 265.6 | 266 | 28.3 | 24 | -0.9 |
| I8 | 362.6 | 373 | 52.3 | 61 | -1.1 |

# Lecture 3: Comparing Mean vs. Mediany

The mean and median measure different aspects of a distribution:

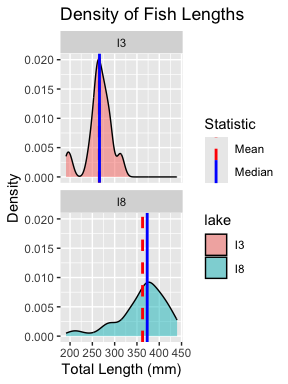
Mean: Center of gravity of the distribution

Median: Middle value of the data

When a distribution is symmetric, the mean and median are similar. When it’s skewed or has outliers, they can differ significantly.

stuff

# Create a density plot with vertical lines for mean and median  
ggplot(grayling\_df, aes(x = length\_mm, fill = lake)) +  
 geom\_density(alpha = 0.5) +  
 geom\_vline(data = stats\_by\_lake,   
 aes(xintercept = mean, color = "Mean"),   
 linetype = "dashed", linewidth = 1) +  
 geom\_vline(data = stats\_by\_lake,   
 aes(xintercept = median, color = "Median"),   
 linetype = "solid", linewidth = 1) +  
 scale\_color\_manual(values = c("Mean" = "red", "Median" = "blue")) +  
 facet\_wrap(~ lake, ncol = 1) +  
 labs(  
 title = "Density of Fish Lengths",  
 x = "Total Length (mm)",  
 y = "Density",  
 color = "Statistic"  
 )



# Lecture 3: Standard Deviation vs. Interquartile Range

The standard deviation and interquartile range both measure spread, but:

Standard deviation: Sensitive to outliers

Interquartile range: Robust against outliers

When the data is approximately normal, the IQR ≈ 1.35 × standard deviation.

# Calculate the ratio of IQR to SD for our data  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 sd = sd(length\_mm),  
 iqr = IQR(length\_mm),  
 ratio\_iqr\_sd = IQR(length\_mm) / sd(length\_mm)  
 ) %>%  
 kable(caption = "Comparison of SD and IQR by Lake", digits = 2)

Comparison of SD and IQR by Lake

| lake | sd | iqr | ratio\_iqr\_sd |
| --- | --- | --- | --- |
| I3 | 28.30 | 24 | 0.85 |
| I8 | 52.34 | 61 | 1.17 |

# Lecture 3: Standard Deviation vs. Interquartile Range

Percentiles are values that divide a dataset into 100 equal parts.

The 25th percentile is the first quartile (Q1)

The 50th percentile is the median

The 75th percentile is the third quartile (Q3)

The IQR is the difference between Q3 and Q1.

stuff

# Calculate percentiles  
percentiles <- quantile(grayling\_df$length\_mm,   
 probs = c(0.1, 0.25, 0.5, 0.75, 0.9))  
  
# Display the percentiles  
kable(  
 data.frame(  
 Percentile = c("10th", "25th (Q1)", "50th (Median)", "75th (Q3)", "90th"),  
 Value = percentiles  
 ),  
 caption = "Key Percentiles of Fish Length (mm)",  
 digits = 1  
)

Key Percentiles of Fish Length (mm)

|  | Percentile | Value |
| --- | --- | --- |
| 10% | 10th | 251.1 |
| 25% | 25th (Q1) | 270.8 |
| 50% | 50th (Median) | 324.5 |
| 75% | 75th (Q3) | 377.0 |
| 90% | 90th | 408.6 |

# Lecture 3: Handling Missing Values

Let’s examine how missing values affect our descriptive statistics by looking at the mass variable, which has some missing data.

# Check for missing values in mass  
sum(is.na(grayling\_df$mass\_g))

[1] 2

# Calculate descriptive statistics with and without handling missing values  
# Without handling (will produce NA results)  
cat("Mean mass without handling NAs:", mean(grayling\_df$mass\_g), "g\n")

Mean mass without handling NAs: NA g

# With handling missing values  
cat("Mean mass with na.rm=TRUE:", mean(grayling\_df$mass\_g, na.rm = TRUE), "g\n")

Mean mass with na.rm=TRUE: 351.2289 g

# Calculate descriptive statistics by lake, properly handling NAs  
grayling\_df %>%  
 group\_by(lake) %>%  
 summarise(  
 mean\_mass = mean(mass\_g, na.rm = TRUE),  
 median\_mass = median(mass\_g, na.rm = TRUE),  
 sd\_mass = sd(mass\_g, na.rm = TRUE),  
 n\_missing = sum(is.na(mass\_g))  
 ) %>%  
 kable(caption = "Mass Statistics by Lake (Handling Missing Values)", digits = 1)

Mass Statistics by Lake (Handling Missing Values)

| lake | mean\_mass | median\_mass | sd\_mass | n\_missing |
| --- | --- | --- | --- | --- |
| I3 | 150.5 | 147 | 42.2 | 0 |
| I8 | 483.7 | 490 | 176.5 | 2 |

# Lecture 3: Best Practices for Handling Missing Values:

1. Always check for missing values in your data before calculating statistics.
2. Use na.rm = TRUE when calculating summary statistics to handle missing values.
3. Report the number of missing values along with your statistics.
4. Consider whether the missing values are random or might introduce bias.

# **Sampling from a Population**

Now that we have estimates of the sample we need to relate that to the population

In reality, we rarely know the true population parameters. When studying fish in lakes I3 and I8:

* The **population** includes all grayling fish in each lake
* The true population mean (μ) and standard deviation (σ) are unknown
* Our dataset is a **sample** from this population
* We use the sample mean (x̄) to estimate μ
* Sampling introduces random variation in our estimates

Let’s demonstrate how different samples from the same population can give different estimates.

If we could sample all fish in the lake, we would know the true mean length. But that’s usually impossible in ecology!

# **Demonstrating Sampling Variation**

Let’s take several random samples from Lake I3 and see how the sample means vary:

# Filter for Lake I3  
i3\_data <- grayling\_df %>% filter(lake == "I3")  
  
# Function to take a random sample and calculate the mean  
sample\_mean <- function(data, sample\_size) {  
 sample\_data <- sample\_n(data, sample\_size)  
 return(mean(sample\_data$length\_mm))  
}  
  
# Take 10 different samples of size 15 from Lake I3  
set.seed(123) # For reproducibility  
sample\_size <- 15  
sample\_means <- replicate(10, sample\_mean(i3\_data, sample\_size))  
  
# Create a data frame with sample numbers and means  
samples\_df <- data.frame(  
 sample\_number = 1:10,  
 sample\_mean = sample\_means  
)  
  
# Display the sample means  
samples\_df

sample\_number sample\_mean  
1 1 269.9333  
2 2 260.6000  
3 3 255.2000  
4 4 263.4000  
5 5 275.3333  
6 6 279.2667  
7 7 263.7333  
8 8 273.6000  
9 9 264.8000  
10 10 269.8667

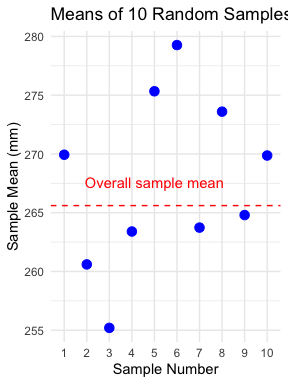
# Calculate the mean and standard deviation of the sample means  
mean(sample\_means)

[1] 267.5733

sd(sample\_means)

[1] 7.346063

# Plot the different sample means  
ggplot(samples\_df, aes(x = factor(sample\_number), y = sample\_mean)) +  
 geom\_point(size = 3, color = "blue") +  
 geom\_hline(yintercept = mean(i3\_data$length\_mm),   
 linetype = "dashed", color = "red") +  
 annotate("text", x = 5, y = mean(i3\_data$length\_mm) + 2,   
 label = "Overall sample mean", color = "red") +  
 labs(title = "Means of 10 Random Samples from Lake I3",  
 x = "Sample Number",  
 y = "Sample Mean (mm)") +  
 theme\_minimal()



Notice how each sample’s mean differs from the overall mean. This demonstrates sampling variation.

# **Standard Error: Quantifying Uncertainty**

The **standard error of the mean (SEM)** measures the precision of a sample mean as an estimate of the population mean.

Formula:

Where: - s is the sample standard deviation - n is the sample size

The standard error tells us: - How much uncertainty is in our estimate - How much sample means are expected to vary - How close our sample mean is likely to be to the true population mean

**Remember:** - Standard deviation (s) describes the variability in the individual data points - Standard error (SE) describes the variability in the sample mean itself - As sample size increases, SE decreases (more precise estimate)

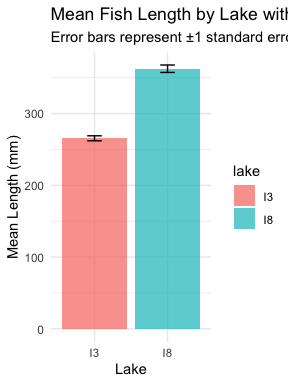
# **Standard Error for Our Grayling Data**

Let’s calculate and visualize the standard error for both lakes:

# Calculate mean, SD, and SE for each lake  
grayling\_stats <- grayling\_df %>%  
 group\_by(lake) %>%  
 summarize(  
 mean\_length = mean(length\_mm),  
 sd\_length = sd(length\_mm),  
 n = n(),  
 se\_length = sd\_length / sqrt(n)  
 )  
  
# Display the statistics  
grayling\_stats

# A tibble: 2 × 5  
 lake mean\_length sd\_length n se\_length  
 <chr> <dbl> <dbl> <int> <dbl>  
1 I3 266. 28.3 66 3.48  
2 I8 363. 52.3 102 5.18

# Create a bar plot with error bars representing ±1 SE  
ggplot(grayling\_stats, aes(x = lake, y = mean\_length, fill = lake)) +  
 geom\_bar(stat = "identity", alpha = 0.7) +  
 geom\_errorbar(aes(ymin = mean\_length - se\_length,   
 ymax = mean\_length + se\_length),  
 width = 0.2) +  
 labs(title = "Mean Fish Length by Lake with Standard Error",  
 subtitle = "Error bars represent ±1 standard error",  
 x = "Lake",  
 y = "Mean Length (mm)") +  
 theme\_minimal()



# **Sampling Distribution of the Mean**

The **sampling distribution of the mean** is the theoretical distribution of all possible sample means of a given sample size from a population.

Important properties: 1. It is centered at the population mean (μ) 2. Its standard deviation is the standard error (σ/√n) 3. For large sample sizes, it approaches a normal distribution (Central Limit Theorem)

The larger the sample size: - The narrower the sampling distribution - The smaller the standard error - The more precise our estimate of the population mean

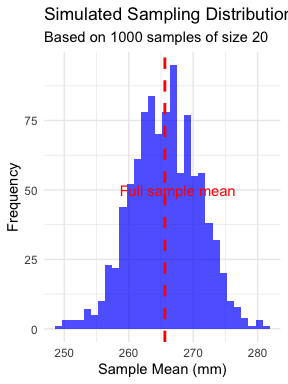
Let’s simulate the sampling distribution for Lake I3 fish data.

# **Simulating the Sampling Distribution**

Let’s simulate taking many samples from Lake I3 to visualize the sampling distribution:

# Filter for Lake I3  
i3\_data <- grayling\_df %>% filter(lake == "I3")  
  
# Number of samples to simulate  
num\_simulations <- 1000  
sample\_size <- 20  
  
# Simulate many samples and calculate means  
set.seed(456) # For reproducibility  
simulated\_means <- replicate(num\_simulations, sample\_mean(i3\_data, sample\_size))  
  
# Calculate the mean and standard deviation of the simulated means  
mean\_of\_means <- mean(simulated\_means)  
sd\_of\_means <- sd(simulated\_means)  
  
# Create a data frame with the simulated means  
simulated\_df <- data.frame(sample\_mean = simulated\_means)  
  
# Plot the sampling distribution  
ggplot(simulated\_df, aes(x = sample\_mean)) +  
 geom\_histogram(bins = 30, fill = "blue", alpha = 0.7) +  
 geom\_vline(xintercept = mean(i3\_data$length\_mm),   
 linetype = "dashed", color = "red", size = 1) +  
 annotate("text", x = mean(i3\_data$length\_mm) + 2, y = 50,   
 label = "Full sample mean", color = "red") +  
 labs(title = "Simulated Sampling Distribution of the Mean",  
 subtitle = paste("Based on", num\_simulations, "samples of size", sample\_size),  
 x = "Sample Mean (mm)",  
 y = "Frequency") +  
 theme\_minimal()

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



Notice that the simulated sampling distribution:

1. Is approximately normally distributed
2. Is centered around the overall sample mean
3. Has a spread that is related to the standard error

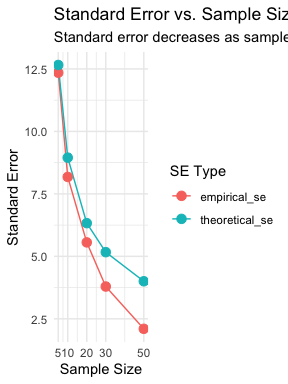
# **Standard Error and Sample Size**

Let’s see how the standard error changes with different sample sizes:

# Define a range of sample sizes to test  
sample\_sizes <- c(5, 10, 20, 30, 50)  
  
# For each sample size, simulate the sampling distribution and calculate SE  
results <- data.frame()  
  
for (size in sample\_sizes) {  
 # Simulate many sample means for this sample size  
 simulated\_means <- replicate(500, sample\_mean(i3\_data, size))  
   
 # Calculate the standard deviation of the sampling distribution (empirical SE)  
 empirical\_se <- sd(simulated\_means)  
   
 # Calculate the theoretical SE  
 theoretical\_se <- sd(i3\_data$length\_mm) / sqrt(size)  
   
 # Add to results  
 results <- rbind(results, data.frame(  
 sample\_size = size,  
 empirical\_se = empirical\_se,  
 theoretical\_se = theoretical\_se  
 ))  
}  
  
# Display the results  
results

sample\_size empirical\_se theoretical\_se  
1 5 12.349407 12.657835  
2 10 8.178270 8.950441  
3 20 5.558957 6.328918  
4 30 3.792177 5.167540  
5 50 2.099744 4.002759

# Plot how SE changes with sample size  
results\_long <- pivot\_longer(results,   
 cols = c(empirical\_se, theoretical\_se),  
 names\_to = "se\_type",   
 values\_to = "standard\_error")  
  
ggplot(results\_long, aes(x = sample\_size, y = standard\_error, color = se\_type)) +  
 geom\_line() +  
 geom\_point(size = 3) +  
 scale\_x\_continuous(breaks = sample\_sizes) +  
 labs(title = "Standard Error vs. Sample Size",  
 subtitle = "Standard error decreases as sample size increases",  
 x = "Sample Size",  
 y = "Standard Error",  
 color = "SE Type") +  
 theme\_minimal()



# **Confidence Intervals**

A **confidence interval** is a range of values that is likely to contain the true population parameter.

The 95% confidence interval for the mean is approximately:

This “2 SE rule of thumb” means: - The interval extends 2 standard errors below and above the sample mean - About 95% of such intervals constructed from different samples would contain the true population mean

Confidence intervals provide a way to express the precision of our estimates.

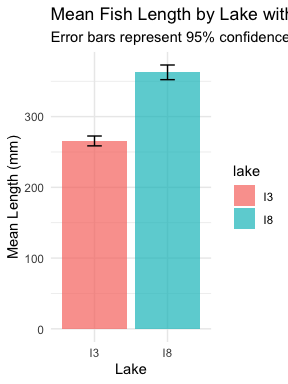
# **Calculating Confidence Intervals for Grayling Data**

Let’s calculate and visualize the 95% confidence intervals for the mean fish length in each lake:

# Calculate 95% confidence intervals  
grayling\_ci <- grayling\_df %>%  
 group\_by(lake) %>%  
 summarize(  
 mean\_length = mean(length\_mm),  
 sd\_length = sd(length\_mm),  
 n = n(),  
 se\_length = sd\_length / sqrt(n),  
 ci\_lower = mean\_length - 2 \* se\_length,  
 ci\_upper = mean\_length + 2 \* se\_length  
 )  
  
# Display the confidence intervals  
grayling\_ci

# A tibble: 2 × 7  
 lake mean\_length sd\_length n se\_length ci\_lower ci\_upper  
 <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
1 I3 266. 28.3 66 3.48 259. 273.  
2 I8 363. 52.3 102 5.18 352. 373.

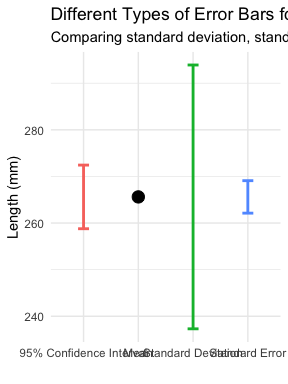
# Plot with confidence intervals  
ggplot(grayling\_ci, aes(x = lake, y = mean\_length, fill = lake)) +  
 geom\_bar(stat = "identity", alpha = 0.7) +  
 geom\_errorbar(aes(ymin = ci\_lower, ymax = ci\_upper),  
 width = 0.2) +  
 labs(title = "Mean Fish Length by Lake with 95% Confidence Intervals",  
 subtitle = "Error bars represent 95% confidence intervals",  
 x = "Lake",  
 y = "Mean Length (mm)") +  
 theme\_minimal()



# **Different Types of Error Bars**

Let’s compare different ways of displaying uncertainty in our estimates:

# Calculate statistics for different types of error bars  
grayling\_error\_bars <- grayling\_df %>%  
 group\_by(lake) %>%  
 summarize(  
 mean\_length = mean(length\_mm),  
 sd\_length = sd(length\_mm),  
 n = n(),  
 se\_length = sd\_length / sqrt(n),  
 ci\_lower = mean\_length - 1.96 \* se\_length,  
 ci\_upper = mean\_length + 1.96 \* se\_length,  
 one\_sd\_lower = mean\_length - sd\_length,  
 one\_sd\_upper = mean\_length + sd\_length  
 )  
  
# Create a data frame for plotting different error types  
lake\_i3 <- grayling\_error\_bars %>% filter(lake == "I3")  
  
error\_types <- data.frame(  
 error\_type = c("Standard Deviation", "Standard Error", "95% Confidence Interval"),  
 lower = c(lake\_i3$one\_sd\_lower,   
 lake\_i3$mean\_length - lake\_i3$se\_length,   
 lake\_i3$ci\_lower),  
 upper = c(lake\_i3$one\_sd\_upper,   
 lake\_i3$mean\_length + lake\_i3$se\_length,   
 lake\_i3$ci\_upper)  
)  
  
# Plot the comparison  
ggplot() +  
 geom\_point(data = lake\_i3, aes(x = "Mean", y = mean\_length), size = 4) +  
 geom\_errorbar(data = error\_types,   
 aes(x = error\_type, ymin = lower, ymax = upper, color = error\_type),  
 width = 0.2, size = 1) +  
 labs(title = "Different Types of Error Bars for Lake I3",  
 subtitle = "Comparing standard deviation, standard error, and 95% confidence interval",  
 x = "",  
 y = "Length (mm)",  
 color = "Error Bar Type") +  
 theme\_minimal() +  
 theme(legend.position = "none")



# **Key Takeaways**

* The **standard error** measures the precision of a sample statistic as an estimate of a population parameter
* The standard error of the mean decreases as sample size increases:
* The **sampling distribution** shows the variation in sample statistics that would be expected due to random sampling
* **Confidence intervals** provide a range of plausible values for the population parameter
* Larger sample sizes provide more precise estimates (narrower confidence intervals)
* When reporting results, always include a measure of precision (SE or

# **For Further Practice**

* Try calculating the standard error and confidence intervals for other variables in the dataset
* Experiment with different sample sizes to see how they affect the precision of estimates
* Compare the means of the two lakes using confidence intervals - do they overlap?
* Consider how these concepts extend to other statistics beyond the mean

# Lecture 3: Conclusion

In this lecture, we’ve explored:

* Why statistics is essential in biology
* Types of biological variables and their properties
* Accuracy, precision, and bias in measurements
* Measures of central tendency (mean, median, geometric mean)
* Measures of spread (standard deviation, variance, and interquartile range)
* Data transformations for skewed distributions
* Visualization techniques for understanding distributions
* Handling missing values

These tools form the foundation of statistical analysis and will be essential as we move forward to more complex statistical methods.