Week 3:
Coding in R I: syntax, variables, arrays, and dataframes; for loops, apply

Group work (2–3 students)

Objectives

- Writing basic R syntax
- Creating and working with variables, vectors, and arrays
- Building and exploring data frames
- Writing for loops
- Using the `apply` family of functions

Part 1: Basic Syntax and Variables

1. Create a variable called `temp_C` and assign it the value of 18.5 (degrees Celsius).
2. Convert `temp_C` to Fahrenheit and store it in a new variable called `temp_F`.
3. Write a single line of code that prints:
   `"The water temperature is XX °C (YY °F)"`, where `XX` and `YY` are replaced with your variables.

Part 2: Vectors and Arrays

1. Create a vector called `species_counts` with the following values:
   - Bluegill = 12, Bass = 7, Sunfish = 21, Carp = 3
2. Write code to:
   - Find the total number of fish counted.
   - Find the species with the highest count.
3. Create a 3×3 array (matrix) that represents chlorophyll concentrations (µg/L) measured at 3 depths (surface, mid, bottom) on 3 different days. Use arbitrary numbers but make them realistic (e.g., 1–50).
4. Calculate the average chlorophyll concentration at each depth across days.

Part 3: Data Frames

You sampled dissolved oxygen (mg/L) and temperature (°C) in 5 lakes.

Lake	Temp_C	DO_mgL
Mendota	22.4	8.3
Wingra	25.1	6.7
Monona	23.7	7.5
Waubesa	24.6	7.9
Kegonsa	26.0	6.2

1. Enter this data into a data frame called `lakes`.
2. Calculate the mean temperature and mean dissolved oxygen across all lakes.
3. Add a new column called `Temp_F` with values converted to Fahrenheit.
4. [BONUS] install package <LakeMetabolizer>. Add new column for the equilibrium concentration of oxygen in water. Add a second new column of dissolved oxygen % saturation. Sort the dataframe in order of DO % saturation using the `order()` function.

Part 4: For Loops

1. Write a for loop that prints the square of each number from 1 to 10.
2. Suppose you want to model exponential population growth with $N_t=N_0e^{rt}$
   - Let $N_0=10$, $r=0.3$, and simulate 10 time steps.
   - Use a for loop to calculate population size at each time step and store results in a vector called `pop`.

You collected **phosphorus concentration data (µg/L)** from 5 lakes. Each lake was sampled 4 times. The data are stored as a **list**, where each element is a vector of values for one lake.

3. Create a list called `phosphorus` with 5 elements (one per lake), each containing 4 numeric values. Use made-up numbers, but keep them realistic (e.g., 5–40 µg/L).
4. Write a **for loop** that:
   - Iterates through each lake in the list.
   - Calculates the **mean phosphorus concentration** for that lake.
   - Stores these means in a new numeric vector called `lake_means`.
   - Prints a message for each lake, e.g., `"Lake1 mean phosphorus = 18.75 µg/L"`.
5. At the end, print the vector of means (`lake_means`).

Part 5: Apply Functions

1. Revisit your chlorophyll **array** from Part 2. Use `apply()` to calculate:
   - The mean concentration for each depth (rows).
   - The mean concentration for each day (columns).
2. Revisit your **lakes** data frame. Use `apply()` to calculate the range (max – min) of each numeric column.

3. Compare your `for` loop population growth results with using `sapply()` or `lapply()`. Which feels easier or cleaner?

Deliverables

- A single R script that runs start-to-finish without errors.
- Comment your code so another ecologist could follow your logic.
- Submit GitHub repo URL
