Name: _____ LA Initials: [ ]

Lab 8: Transforming Data

Learning Objectives

By the end of this lab, students will be able to:

- Use `filter()` to subset rows of a data frame based on logical conditions.
- Use `select()` to subset columns of a data frame by name.
- Create new variables using `mutate()`, including transformations of existing variables.
- Reshape data from wide to long format using `pivot_longer()`, and from long to wide format using `pivot_wider()`.
- Join two data frames together using `left_join()`, `inner_join()`, or `full_join()`.
- Combine multiple data manipulation steps into a single pipeline using the pipe operator (`|>`).

Getting started

Before you begin the lab activities, review the steps for setting up a project for a lab activity:

- Weekly Lab Setup

Then, complete the following steps to make sure you are fully set up.

1. **Get the Lab Worksheet.**

   Pick up a physical copy of the lab worksheet, or print one if you are working outside of class.
   Download Lab Worksheet (PDF, if needed)

2. **Open Posit Cloud and start Lab 8.**

3. **Install required packages**

   - tidyverse
   - palmerpenguins
   - lterdatasampler

4. **Create an R Script**

5. **Load the packages**

> **i** Checkpoint
>
> At this point, you should have:
>
> - These instructions open in a web browser.
> - Your Lab 8 project open in Posit Cloud in another browser window.
> - The required packages installed and loaded without errors
> - The Lab 8 worksheet in front of you.
>
> Do not continue until all of the above steps are working correctly.

Overview

Real data are rarely ready for analysis. Before you can calculate a summary statistic, fit a model, or create a figure, you almost always need to reorganize the dataset. That means selecting only the rows you need, keeping only relevant

variables, creating new variables that better represent your biological question, and restructuring the data into a format appropriate for analysis or visualization.

In this lab, you will focus on core data wrangling skills: subsetting rows and columns, creating new variables, reshaping data between wide and long formats, and joining multiple datasets together. These tasks are foundational. If you cannot reliably manipulate data, you cannot conduct reproducible analyses.

We will use functions from the **tidyverse** exclusively. While many of these tasks can be done with base R, the tidyverse provides a consistent grammar for data manipulation. Functions such as `filter()`, `select()`, `mutate()`, `pivot_longer()`, `pivot_wider()`, and the various `*_join()` functions are designed to work together and integrate cleanly with the pipe operator (`|>`). This allows you to express a sequence of data transformations in a clear, readable workflow.

The goal is not just to learn individual functions, but to think in terms of pipelines. You should be able to start with a raw dataset and move step-by-step—subsetting, transforming, reshaping, and combining—until the data are structured appropriately for the question you want to answer. This lab builds the practical skills required for all subsequent statistical analysis and visualization in this course.

The Problem

Today's task is to prepare data for a biologically meaningful analysis using the `nwt_pikas` dataset.

This dataset contains field measurements from North American pikas (*Ochotona princeps*) collected at multiple alpine sites. For each sampling event, researchers measured **glucocorticoid metabolites (GCMs)** from fecal samples. GCM concentration is a commonly used noninvasive indicator of physiological stress. Elevated glucocorticoid levels can reflect environmental stressors such as temperature extremes, predation risk, habitat quality, or resource limitation. In alpine systems—where pikas are sensitive to heat—GCMs may provide insight into how individuals respond to environmental conditions.

For this lab, we will focus on two sites: **West Knoll 3** and **West Knoll 6**. These sites are geographically close to one another. If local environmental conditions are driving physiological stress, we might expect GCM concentrations at these two sites to vary in similar ways over time. In statistical terms, we are asking:

> Are GCM measurements at West Knoll 3 and West Knoll 6 correlated?

To evaluate this, we will need to: - Subset the data to include only these two sites. - Reshape the data so that GCM values from each site appear in separate columns. - Create a scatterplot with one site on the x-axis and the other on the y-axis.

One environmental mechanism that may drive variation in GCM concentration is air temperature. If temperature fluctuates over time and influences physiological stress in pikas, then GCM values at West Knoll 3 and West Knoll 6 may rise and fall together because both sites are responding to the same temperature conditions. In this case, temperature serves as a shared environmental driver that could explain why measurements at the two sites appear correlated.

To examine this possibility, we will incorporate daily weather data from a nearby meteorological station (White et al., Niwot Ridge LTER; https://portal.edirepository.org/nis/mapbrowse?packageid=knb-lter-nwt.314.4). We will join these weather data to the pika dataset by date. This will allow us to examine:

- How GCM concentration at each site varies with temperature.
- Whether temperature helps explain any correlation between sites.

Completing this analysis will require filtering, selecting, mutating, reshaping, joining, and piping multiple steps together—exactly the data preparation skills emphasized in this lab.

Task Outline

1. Import the weather data

- In the lab project, run the provided script: `import-weather-data.R`.
- This script imports the weather dataset needed for the lab.

2. Prepare the weather table (`dt1`)

After running the script, you should see a data frame named `dt1` in your Environment.

1. Keep only the needed columns:
   - `date`
   - `airtemp_avg_gapfilled`
2. Convert `date` from character (`chr`) to Date format using `mutate()` and `as.Date()`.

3. Prepare the pika data (`nwt_pikas`)

Work from the `nwt_pikas` dataset and apply the following steps in a single pipeline.

a. Subset to the two sites of interest

- Filter to include only **West Knoll 3** and **West Knoll 6**.

b. Keep only high-elevation sites

- Filter to include only sites with **elevation > 3300 m**.

c. Convert GCM units

- Convert glucocorticoid metabolite concentration from **picograms/gram** to **nanograms/gram** by dividing by **1000**.
- Store the converted values in a new variable.

d. Keep only the required columns

- Select:
  - `date`
  - `station`
  - your new concentration variable

e. Reshape to wide format

- Use `pivot_wider()` so that each station becomes its own column, with one row per date.

f. Remove incomplete paired observations

- Identify dates where one site has `NA` while the other has a value.
- Filter those dates out so that both sites have data on every remaining date.

4. Scatterplot: site vs site

- Make a scatterplot with:
  - **West Knoll 3** on one axis
  - **West Knoll 6** on the other axis

5. Join pika + weather data

- Join the transformed pika dataset with the transformed weather dataset using:
  - `inner_join()`
- Join key:
  - `date` (in both tables)

6. Scatterplot: GCM vs temperature

- Make a scatterplot showing:
  - **West Knoll 3 GCM concentration** (y or x)
  - **air temperature** (`airtemp_avg_gapfilled`) (the other axis)

Tips and Tools

Core Functions You Will Likely Need

From **dplyr**: - `filter()` — subset rows based on logical conditions
- `select()` — choose specific columns
- `mutate()` — create or modify variables
- `inner_join()` — join two data frames by a key column

From **tidyr**: - `pivot_wider()` — reshape data from long to wide format

From **base R**: - `as.Date()` — convert character strings to Date format

From **ggplot2**: - `ggplot()`
- `geom_point()`

Using the Native Pipe Operator (|>)

The pipe operator passes the result of one step directly into the next step.
Instead of creating many intermediate objects, you build a sequence of transformations in order.

Without a pipe:

```
step1 <- filter(nwt_pikas, station == "West Knoll 3")
step2 <- select(step1, date, station, gcm_concentration)
```

With a pipe:

```
nwt_pikas |>
  filter(station == "West Knoll 3") |>
  select(date, station, gcm_concentration)
```

How it works:

- The object on the left of |> becomes the first argument of the function on the right.
- Each line performs one transformation.
- The output of one step flows directly into the next.

This allows you to:

- Read the code from top to bottom.
- Express a complete workflow in a single, reproducible pipeline.
- Avoid creating unnecessary intermediate objects.

For this lab, most of your data preparation should be written as pipelines.

What to turn in

Show your script and graphs to an LA for completion credit.

Create your two graphs and upload them to the D2L assignment.