Coding Challenge 5

Muhtarin Khayer Brohee

2025-03-20

Table of Contents

[Click here to view my GitHub repository](https://github.com/mzb0226/PLPA_6820/tree/main/Coding_Challenge_5)

## **Loading tidyverse**

# Load tidyverse for data wrangling and visualization  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

## **Loading datasets**

# Read the datasets using relative paths  
diversity <- read\_csv("DiversityData.csv", show\_col\_types = FALSE)  
metadata <- read\_csv("Metadata.csv", show\_col\_types = FALSE)

## **Joining data using left\_join**

# Join datasets by 'Code'  
alpha <- left\_join(diversity, metadata, by = "Code")  
  
# Check the first few rows  
head(alpha)

## # A tibble: 6 × 9  
## Code shannon invsimpson simpson richness Crop Time\_Point Replicate  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>  
## 1 S01\_13 6.62 211. 0.995 3319 Soil 0 1  
## 2 S02\_16 6.61 207. 0.995 3079 Soil 0 2  
## 3 S03\_19 6.66 213. 0.995 3935 Soil 0 3  
## 4 S04\_22 6.66 205. 0.995 3922 Soil 0 4  
## 5 S05\_25 6.61 200. 0.995 3196 Soil 0 5  
## 6 S06\_28 6.65 199. 0.995 3481 Soil 0 6  
## # ℹ 1 more variable: Water\_Imbibed <chr>

# Check column names first  
colnames(alpha)

## [1] "Code" "shannon" "invsimpson" "simpson"   
## [5] "richness" "Crop" "Time\_Point" "Replicate"   
## [9] "Water\_Imbibed"

## **Mutate() function and pipe output of one into input of another (%>%)**

# Create a new column for Evenness  
alpha\_even <- alpha %>%  
 mutate(Evenness = shannon / log(richness))

## **Summarise data to find sd, mean, standard error**

# Display first few rows  
head(alpha\_even)

## # A tibble: 6 × 10  
## Code shannon invsimpson simpson richness Crop Time\_Point Replicate  
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>  
## 1 S01\_13 6.62 211. 0.995 3319 Soil 0 1  
## 2 S02\_16 6.61 207. 0.995 3079 Soil 0 2  
## 3 S03\_19 6.66 213. 0.995 3935 Soil 0 3  
## 4 S04\_22 6.66 205. 0.995 3922 Soil 0 4  
## 5 S05\_25 6.61 200. 0.995 3196 Soil 0 5  
## 6 S06\_28 6.65 199. 0.995 3481 Soil 0 6  
## # ℹ 2 more variables: Water\_Imbibed <chr>, Evenness <dbl>

# Summarise and pipe  
alpha\_average <- alpha\_even %>%  
 group\_by(Crop, Time\_Point) %>%  
 summarise(  
 mean\_even = mean(Evenness, na.rm = TRUE), # Mean Evenness  
 count = n(), # Sample count  
 sd\_even = sd(Evenness, na.rm = TRUE), # Standard deviation  
 se\_even = sd\_even / sqrt(count) # Standard error  
 )

## `summarise()` has grouped output by 'Crop'. You can override using the  
## `.groups` argument.

# Display results  
head

## function (x, ...)   
## UseMethod("head")  
## <bytecode: 0x00000243e18462a0>  
## <environment: namespace:utils>

## **Converting data from long format to wide (Pivot)**

alpha\_average2 <- alpha\_average %>%  
 select(Time\_Point, Crop, mean\_even) %>%  
 pivot\_wider(names\_from = Crop, values\_from = mean\_even) %>%  
 mutate(  
 diff\_cotton\_even = Cotton - Soil,  
 diff\_soybean\_even = Soybean - Soil  
 )  
  
# Display the transformed data  
head(alpha\_average2)

## # A tibble: 4 × 6  
## Time\_Point Cotton Soil Soybean diff\_cotton\_even diff\_soybean\_even  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 0.820 0.814 0.822 0.00602 0.00740  
## 2 6 0.805 0.810 0.764 -0.00507 -0.0459   
## 3 12 0.767 0.798 0.687 -0.0313 -0.112   
## 4 18 0.755 0.800 0.716 -0.0449 -0.0833

## **Piping altogether to create a plot**

# Reshape data to long format for ggplot  
alpha\_long <- alpha\_average2 %>%  
 pivot\_longer(c(diff\_cotton\_even, diff\_soybean\_even), names\_to = "diff")  
  
# Generate the line plot  
plot <- ggplot(alpha\_long, aes(x = Time\_Point, y = value, color = diff)) +  
 geom\_line() +  
 theme\_minimal() +  
 labs(x = "Time Point", y = "Difference in Evenness", color = "Crop Type")  
 print(plot)

