Coding Challenge: Data Wragling

Prativa Chhetri and Karamjit Kaur Baryah

2025-03-20

Table of Contents

# Question 1.

metadata <- read.csv("Metadata.csv", na.strings = "na")  
diversity <- read.csv("DiversityData.csv")  
str(metadata)

## 'data.frame': 70 obs. of 5 variables:  
## $ Code : chr "S01\_13" "S02\_16" "S03\_19" "S04\_22" ...  
## $ Crop : chr "Soil" "Soil" "Soil" "Soil" ...  
## $ Time\_Point : int 0 0 0 0 0 0 6 6 6 6 ...  
## $ Replicate : int 1 2 3 4 5 6 1 2 3 4 ...  
## $ Water\_Imbibed: num NA NA NA NA NA NA NA NA NA NA ...

str(diversity)

## 'data.frame': 70 obs. of 5 variables:  
## $ Code : chr "S01\_13" "S02\_16" "S03\_19" "S04\_22" ...  
## $ shannon : num 6.62 6.61 6.66 6.66 6.61 ...  
## $ invsimpson: num 211 207 213 205 200 ...  
## $ simpson : num 0.995 0.995 0.995 0.995 0.995 ...  
## $ richness : int 3319 3079 3935 3922 3196 3481 3250 3170 3657 3177 ...

# Question 2.

alpha <- left\_join(diversity, metadata, by = "Code")  
head(alpha)

## Code shannon invsimpson simpson richness Crop Time\_Point Replicate  
## 1 S01\_13 6.624921 210.7279 0.9952545 3319 Soil 0 1  
## 2 S02\_16 6.612413 206.8666 0.9951660 3079 Soil 0 2  
## 3 S03\_19 6.660853 213.0184 0.9953056 3935 Soil 0 3  
## 4 S04\_22 6.660671 204.6908 0.9951146 3922 Soil 0 4  
## 5 S05\_25 6.610965 200.2552 0.9950064 3196 Soil 0 5  
## 6 S06\_28 6.650812 199.3211 0.9949830 3481 Soil 0 6  
## Water\_Imbibed  
## 1 NA  
## 2 NA  
## 3 NA  
## 4 NA  
## 5 NA  
## 6 NA

# Question 3

alpha\_even <- alpha %>%  
 mutate(even = shannon / log(richness))  
head(alpha\_even)

## Code shannon invsimpson simpson richness Crop Time\_Point Replicate  
## 1 S01\_13 6.624921 210.7279 0.9952545 3319 Soil 0 1  
## 2 S02\_16 6.612413 206.8666 0.9951660 3079 Soil 0 2  
## 3 S03\_19 6.660853 213.0184 0.9953056 3935 Soil 0 3  
## 4 S04\_22 6.660671 204.6908 0.9951146 3922 Soil 0 4  
## 5 S05\_25 6.610965 200.2552 0.9950064 3196 Soil 0 5  
## 6 S06\_28 6.650812 199.3211 0.9949830 3481 Soil 0 6  
## Water\_Imbibed even  
## 1 NA 0.8171431  
## 2 NA 0.8232216  
## 3 NA 0.8046776  
## 4 NA 0.8049774  
## 5 NA 0.8192376  
## 6 NA 0.8155427

# Question 4

alpha\_average <- alpha\_even %>%  
 group\_by(Crop, Time\_Point) %>%   
 summarise(  
 mean.even = mean(even, na.rm = TRUE),  
 n = n(),   
 sd.even = sd(even, na.rm = TRUE)   
 ) %>%  
 mutate(std.err.even = sd.even / sqrt(n))

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

head(alpha\_average)

## # A tibble: 6 × 6  
## # Groups: Crop [2]  
## Crop Time\_Point mean.even n sd.even std.err.even  
## <chr> <int> <dbl> <int> <dbl> <dbl>  
## 1 Cotton 0 0.820 6 0.00556 0.00227  
## 2 Cotton 6 0.805 6 0.00920 0.00376  
## 3 Cotton 12 0.767 6 0.0157 0.00640  
## 4 Cotton 18 0.755 5 0.0169 0.00755  
## 5 Soil 0 0.814 6 0.00765 0.00312  
## 6 Soil 6 0.810 6 0.00587 0.00240

# Question 5.

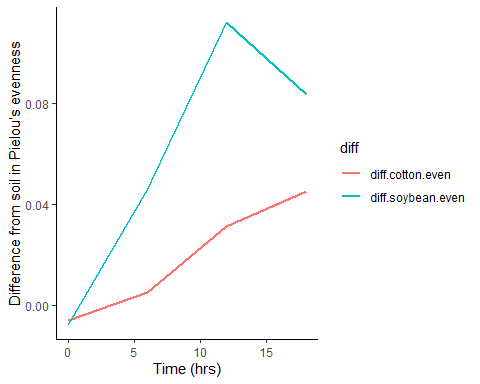
alpha\_average2 <- alpha\_average %>%  
 select(Time\_Point, Crop, mean.even) %>%  
 pivot\_wider(names\_from = Crop, values\_from = mean.even) %>%   
 mutate(  
 diff.cotton.even = Soil - Cotton,   
 diff.soybean.even = Soil - Soybean  
 )  
str(alpha\_average2)

## tibble [4 × 6] (S3: tbl\_df/tbl/data.frame)  
## $ Time\_Point : int [1:4] 0 6 12 18  
## $ Cotton : num [1:4] 0.82 0.805 0.767 0.755  
## $ Soil : num [1:4] 0.814 0.81 0.798 0.8  
## $ Soybean : num [1:4] 0.822 0.764 0.687 0.716  
## $ diff.cotton.even : num [1:4] -0.00602 0.00507 0.03129 0.0449  
## $ diff.soybean.even: num [1:4] -0.0074 0.0459 0.1119 0.0833

# Question 6.

alpha\_average2 %>%  
 select(Time\_Point, diff.cotton.even, diff.soybean.even) %>%  
 pivot\_longer(c(diff.cotton.even, diff.soybean.even), names\_to = "diff") %>%  
 ggplot(aes(x = Time\_Point, y = value, color = diff)) +  
 geom\_line(size = 1) +   
 theme\_classic() +  
 labs(  
 x = "Time (hrs)",  
 y = "Difference from soil in Pielou’s evenness",  
 color = "diff"  
 )

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# Question 7.

[GitHub for Coding Challenge 5] (<https://github.com/PrativaC/Mycotoxin>)