Jaden\_King

[clickable link to .docx](https://github.com/kingjad33/PLPA6820Class/blob/main/Code%20Challenge%205/DataWrangle_CodeChal.docx)

requires tidyverse package

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

read data in

metadata <- read.csv("metadata.csv")  
diversity <- read.csv("DiversityData.csv")

connect the data frames by column code

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

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

Calculate Pielou’s evenness index: Pielou’s evenness is an ecological parameter calculated by the Shannon diversity index (column Shannon) divided by the log of the richness column. - Using mutate, create a new column to calculate Pielou’s evenness index. - Name the resulting dataframe alpha\_even.

alpha\_even <- data.frame(mutate(alpha, Peven = shannon/(log(richness))))  
  
head(alpha\_even)

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

Using tidyverse language of functions and the pipe, use the summarise function and tell me the mean and standard error evenness grouped by crop over time. - Start with the alpha\_even dataframe - Group the data: group the data by Crop and Time\_Point. - Summarize the data: Calculate the mean, count, standard deviation, and standard error for the even variable within each group. - Name the resulting dataframe alpha\_average

alpha\_average<- data.frame(alpha\_even %>%  
 group\_by(Crop, Time\_Point) %>%  
 summarize(Mean.even = mean(Peven),  
 n=n(),  
 sd.dev = sd(Peven))%>%  
 mutate(std.err = sd.dev/sqrt(n)))

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

head(alpha\_average)

## Crop Time\_Point Mean.even n sd.dev std.err  
## 1 Cotton 0 0.8201559 6 0.005564427 0.002271668  
## 2 Cotton 6 0.8045279 6 0.009198591 0.003755309  
## 3 Cotton 12 0.7672012 6 0.015668209 0.006396520  
## 4 Cotton 18 0.7548807 5 0.016887655 0.007552389  
## 5 Soil 0 0.8141333 6 0.007654244 0.003124832  
## 6 Soil 6 0.8096008 6 0.005870261 0.002396524

Calculate the difference between the soybean column, the soil column, and the difference between the cotton column and the soil column - Start with the alpha\_average dataframe - Select relevant columns: select the columns Time\_Point, Crop, and mean.even. - Reshape the data: Use the pivot\_wider function to transform the data from long to wide format, creating new columns for each Crop with values from mean.even. - Calculate differences: Create new columns named diff.cotton.even and diff.soybean.even by calculating the difference between Soil and Cotton, and Soil and Soybean, respectively. - Name the resulting dataframe alpha\_average2

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

## Time\_Point Cotton Soil Soybean diff.cotton.even diff.soybean.even  
## 1 0 0.8201559 0.8141333 0.8215344 -0.006022574 -0.007401079  
## 2 6 0.8045279 0.8096008 0.7637156 0.005072906 0.045885155  
## 3 12 0.7672012 0.7984872 0.6865940 0.031285973 0.111893212  
## 4 18 0.7548807 0.7997796 0.7164671 0.044898885 0.083312497

Connecting it to plots - Start with the alpha\_average2 dataframe - Select relevant columns: select the columns Time\_Point, diff.cotton.even, and diff.soybean.even. - Reshape the data: Use the pivot\_longer function to transform the data from wide to long format, creating a new column named diff that contains the values from diff.cotton.even and diff.soybean.even - Create the plot: Use ggplot and geom\_line() with ‘Time\_Point’ on the x-axis, the column ‘values’ on the y-axis, and different colors for each ‘diff’ category. The column named ‘values’ come from the pivot\_longer. The resulting plot should look like the one to the right.

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() +  
 theme\_classic() +  
 xlab("Time (hrs)") +  
 ylab("Difference from soil in Pielou's evenness")

