Data Wrangling Coding Challenge

Dustyn Lewis

2025-03-25

Load the required libraries

```
library(tidyverse)
```

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

```
getwd()
```

```
## [1] "C:/Users/Dutal/OneDrive/Desktop/Classes/PLPA 6820/Coding Challenge 5/Coding-Challenge-5"
```

1) Read in the CSV files

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

2) Join the two data frames by common column "code"

```
alpha <- left_join(diversity, metadata, by = "Code")
```

3) Calculate Pielou's eveness index ($Shannon/\log(Richness)$)

```
alpha_even <- alpha %>%
  mutate(even = shannon/log(richness))
```

4) Summarize mean and standard error of evenness grouped by *Crop* over *Time_Point*

```r
alpha_average <- alpha_even %>%
  group_by(Crop, Time_Point) %>%
  summarize(
    mean.even = mean(even, na.rm = TRUE),
    n=n(),
    sd.even = sd(even, na.rm = TRUE),
    se.even = sd.even/sqrt(n) ) %>%
  print()
```

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

```
## # A tibble: 12 x 6
## # Groups:   Crop [3]
##    Crop    Time_Point mean.even     n sd.even se.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
##  7 Soil            12     0.798     6 0.00782 0.00319
##  8 Soil            18     0.800     5 0.0104  0.00465
##  9 Soybean          0     0.822     6 0.00270 0.00110
## 10 Soybean          6     0.764     6 0.0400  0.0163
## 11 Soybean         12     0.687     6 0.0643  0.0263
## 12 Soybean         18     0.716     6 0.0153  0.00626
```

5) Calculate differences between crops (Soil - Cotton, Soil - Soybean) for evenness

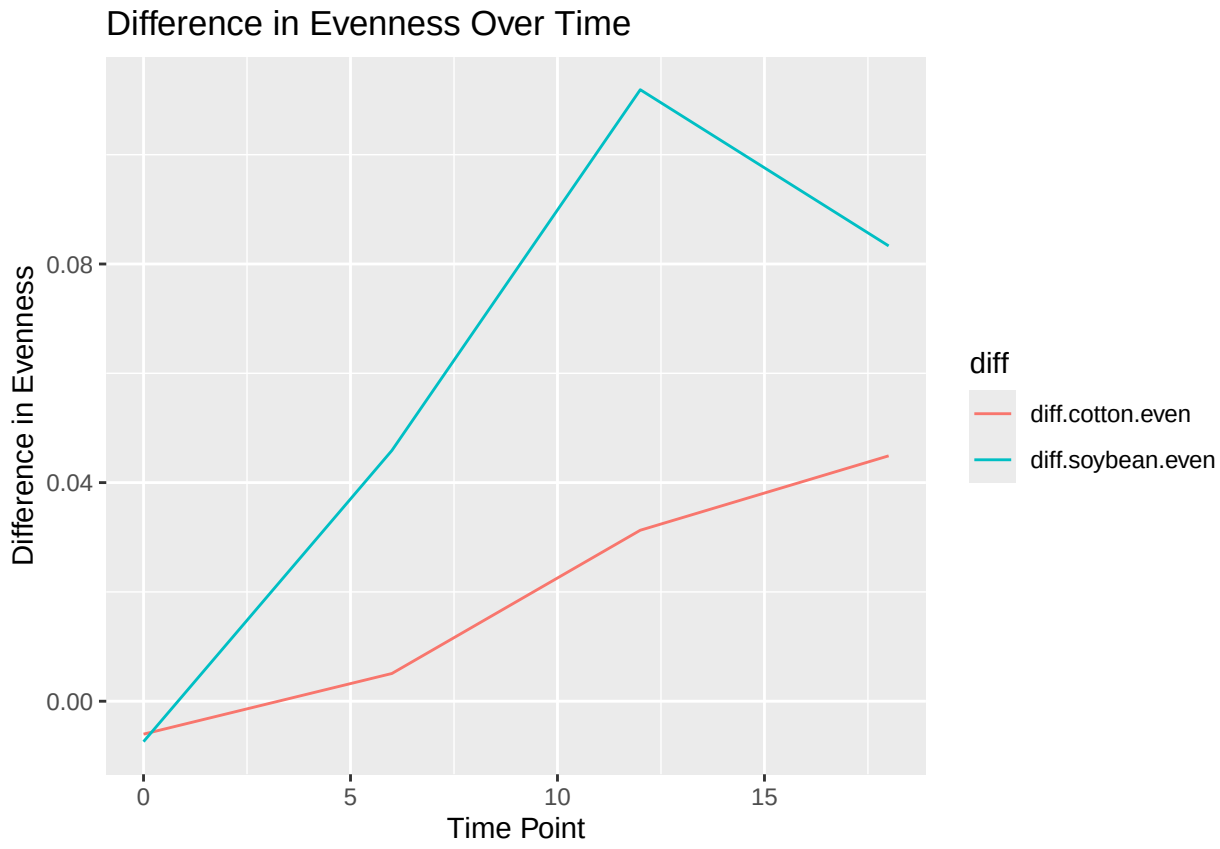
```r
alpha_average2 <- alpha_average %>%
  select(Crop:mean.even) %>%
  pivot_wider(names_from = Crop, values_from = mean.even) %>%
  mutate(diff.cotton.even = Soil - Cotton, diff.soybean.even = Soil - Soybean) %>%
  print()
```

```
## # A tibble: 4 x 6
##   Time_Point Cotton  Soil Soybean diff.cotton.even diff.soybean.even
##        <int>  <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
```

6) Creating plots to compare differences over *Time_Point*

```r
alpha_long <- alpha_average2 %>%
  select(Time_Point, diff.cotton.even, diff.soybean.even) %>%
  pivot_longer( cols = c(diff.cotton.even, diff.soybean.even),
                names_to = "diff", values_to = "values" )
```

```r
ggplot(alpha_long, aes(x = Time_Point, y = values, color = diff)) +
  geom_line() +
  labs(
    title = "Difference in Evenness Over Time",
    x = "Time Point",
    y = "Difference in Evenness"
    )
```



7) Here is a link to GitHub: https://github.com/Dustyn-T-Lewis/Coding-Challenge-5