# Data Wrangling coding challenge 5

lihui xiang

2025-03-20

###1.Download two .csv files from Canvas called DiversityData.csv and Metadata.csv, and read them into R using relative file paths.

```
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 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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
```

```
DiversityData <- read.csv("DiversityData.csv")
Metadata <- read.csv("Metadata.csv")
str(DiversityData)
```

```
## '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 ...
```

```
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: chr  "na" "na" "na" "na" ...
```

###2.Join the two dataframes together by the common column 'Code'. Name the resulting dataframe alpha.

```
alpha <- left_join(DiversityData, Metadata, by = "Code") ##as thses two data all have Code line and sam
str(alpha)
```

```
## 'data.frame':    70 obs. of  9 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 ...
##  $ 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: chr  "na" "na" "na" "na" ...
```

###3.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. a. Using mutate, create a new column to calculate Pielou's evenness index. b. Name the resulting dataframe alpha_even.

```
alpha_even <- alpha %>%
  mutate(Pielou_evenness = shannon / log(richness)) ##calculate Pielou_evenness and save it to alpha_eve
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 Pielou_evenness
## 1            na       0.8171431
## 2            na       0.8232216
## 3            na       0.8046776
## 4            na       0.8049774
## 5            na       0.8192376
## 6            na       0.8155427
```

###4.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. a. Start with the alpha_even dataframe b. Group the data: group the data by Crop and Time_Point. c. Summarize the data: Calculate the mean, count, standard deviation, and standard error for the even variable within each group. d. Name the resulting dataframe alpha_average

```
alpha_average <- alpha_even %>%  ###add the pipe %>% for multiple functions together
  group_by(Crop, Time_Point) %>% ###group the data by Crop and Time_Point
  summarise(
    mean_evenness = mean(Pielou_evenness, na.rm = TRUE), ###Calculate the mean
    count = n(), ###calculate count
    sd_evenness = sd(Pielou_evenness, na.rm = TRUE), ###standard deviation
    se_evenness = sd(Pielou_evenness, na.rm = TRUE) / sqrt(n()) ###standard error
  )
```

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

```
print(alpha_average)
```

```
## # A tibble: 12 x 6
## # Groups:   Crop [3]
##    Crop    Time_Point mean_evenness count sd_evenness se_evenness
##    <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 the difference between the soybean column, the soil column, and the difference between the cotton column and the soil column a. Start with the alpha_average dataframe b. Select relevant columns: select the columns Time_Point, Crop, and mean.even. c. 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. d. 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. e. Name the resulting dataframe alpha_average2

```
alpha_average2 <- alpha_average %>%
  select(Time_Point, Crop, mean_evenness) %>% ###select the columns Time_Point, Crop, and mean.even.
  pivot_wider(names_from = Crop, values_from = mean_evenness) %>%###transform the data from long to wid
  mutate(
    diff.cotton.even = Soil - Cotton, ###calculate the difference between the cotton and soil
    diff.soybean.even = Soil - Soybean ###calculate the difference between the Soil and Soybean
  )

print(alpha_average2)
```

```
## # 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.Connecting it to plots a. Start with the alpha_average2 dataframe b. Select relevant columns: select the columns Time_Point, diff.cotton.even, and diff.soybean.even. c. 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. i. This might be challenging, so I’ll give you a break. The code is below.

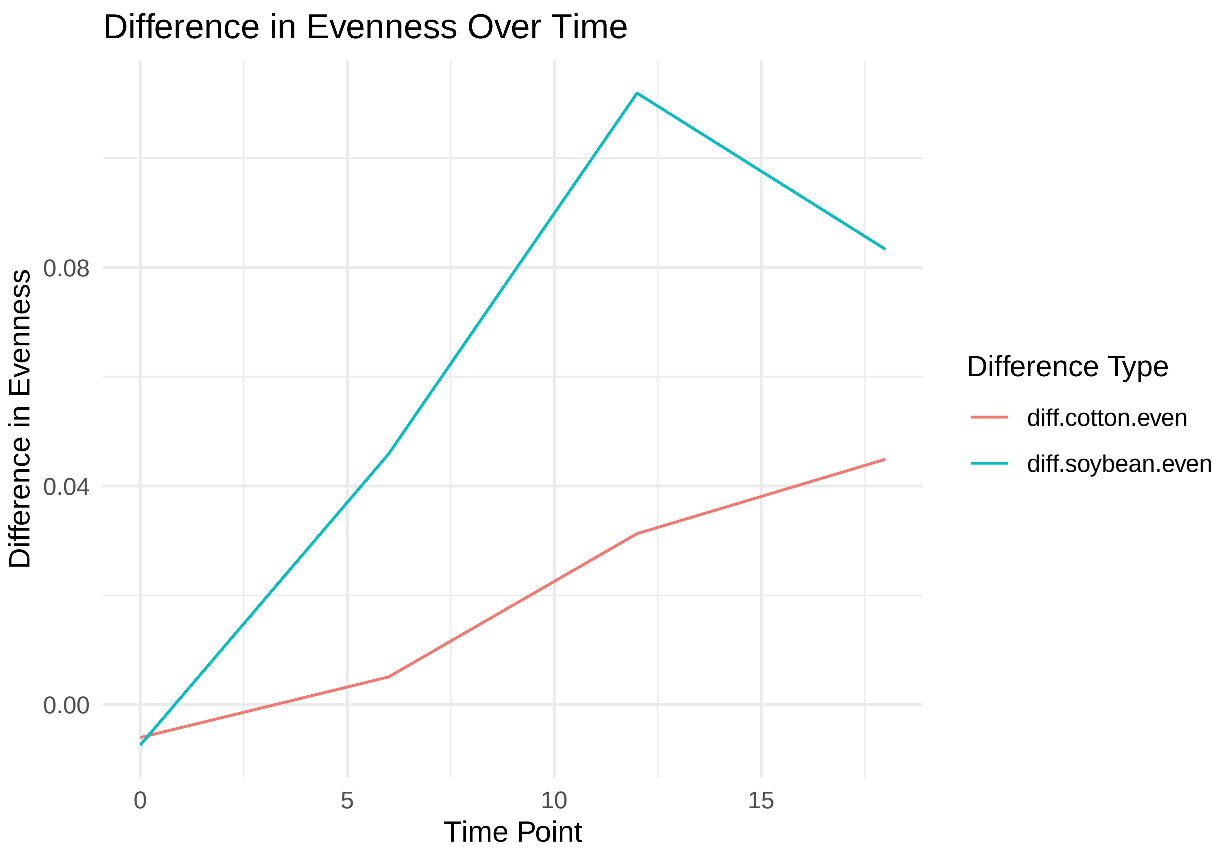
```
library(ggplot2)
library(dplyr)
alpha_long <- alpha_average2 %>%
  select(Time_Point, diff.cotton.even, diff.soybean.even) %>% ###select the columns we want
  pivot_longer(                                               ###transform the data from long to wide fo
    cols = c(diff.cotton.even, diff.soybean.even),            ###creating a new column named diff that
    names_to = "diff",
    values_to = "values"
  )

print(alpha_long)
```

```
## # A tibble: 8 x 3
##   Time_Point diff                values
##        <int> <chr>                <dbl>
## 1          0 diff.cotton.even  -0.00602
## 2          0 diff.soybean.even -0.00740
## 3          6 diff.cotton.even   0.00507
## 4          6 diff.soybean.even  0.0459
## 5         12 diff.cotton.even   0.0313
## 6         12 diff.soybean.even  0.112
## 7         18 diff.cotton.even   0.0449
## 8         18 diff.soybean.even  0.0833
```

###Create the plot

```
ggplot(alpha_long, aes(x = Time_Point, y = values, color = diff)) +
  geom_line() +      ####add line plotr
  labs(
    title = "Difference in Evenness Over Time",
    x = "Time Point",
    y = "Difference in Evenness",
    color = "Difference Type"
  ) +
  theme_minimal()
```


Difference in Evenness Over Time
Difference in Evenness
0.08
0.04
0.00
0
5
10
15
Time Point
Difference Type
diff.cotton.even
diff.soybean.even