2025_3_20_DataWrangling_mer0127

Madeline Redd

2025 - 03 - 19

Seamless Data Wrangling

Manipulating data, Adding new columns, Working with large messy data set

Installing and loading in the tidyverse package. Attaches core tidyverse packages. Tidyverse allows easier work for large data sets.

R script Setup Code Explained

1. include = FALSE prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
2. echo = FALSE prevents code, but not the results from appearing in the finished file. This is a useful way to embed figures.
3. message = FALSE prevents messages that are generated by code from appearing in the finished file.
4. warning = FALSE prevents warnings that are generated by code from appearing in the finished.
5. fig.cap = "..." adds a caption to graphical results.

```
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     1.0.2
## -- 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
```

More info and cheat sheets can be found here: [Tidyverse] (https://tidyr.tidyverse.org/index.html) Lets demonstrate some of the most useful functionality of some tidyverse functions. Note that this tutorial does not cover everything and self-learning may be required for specific functionality.

Goals to learn from the Assignment:

Data wrangling & manipulation
– mutate()
– select()
– filter()
– the pipe %>%

– summarise()
– group_by()
– joining
– pivotting
– Integration with plotting

Loading in the file "Bull_richness.csv". This contains information crops Corn and Soy that includes all fungi in the Phylum Ascomycota. In this assignment the goal is to look at how Fungicide variable impacts richness.

```
microbiome.fungi <- read.csv("Project_Data_Files/Bull_richness.csv")
                                        #hid results to clean up output in pdf

head(microbiome.fungi)        #Shows column names and the first 6 rows of data
str(microbiome.fungi)
```

Select Function `select ()`

Example of function use broken down: newdataframe <- select(data, column_name_wanted, column_name_wanted, column_range_first : column_range_last, column_name_wanted)

```
microbiome.fungi2 <- select(microbiome.fungi, SampleID, Crop,
                            Compartment:Fungicide, richness)
str(microbiome.fungi2)
```

```
## 'data.frame':    287 obs. of  10 variables:
##  $ SampleID   : chr  "Corn2017LeafObjective2Collection1T1R1CAH2" "Corn2017LeafObjective2Collection1T
##  $ Crop       : chr  "Corn" "Corn" "Corn" "Corn" ...
##  $ Compartment: chr  "Leaf" "Leaf" "Leaf" "Leaf" ...
##  $ DateSampled: chr  "6/26/17" "6/26/17" "6/26/17" "6/26/17" ...
##  $ GrowthStage: chr  "V6" "V6" "V6" "V6" ...
##  $ Treatment  : chr  "Conv." "Conv." "Conv." "Conv." ...
##  $ Rep        : chr  "R1" "R1" "R1" "R1" ...
##  $ Sample     : chr  "A" "B" "C" "A" ...
##  $ Fungicide  : chr  "C" "C" "C" "F" ...
##  $ richness   : int  9 6 5 7 4 2 3 8 4 4 ...
```

Filter Function `filter()`

Example of Function head(filter(data, column_name == "Value"))

Similar coding to `subset()` function

```
head(filter(microbiome.fungi2, Treatment == "Conv."))
```

```
##                                   SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1CCB3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R1FAC3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R1FBD3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R1FCE3 Corn        Leaf     6/26/17
```

```
##   GrowthStage Treatment Rep Sample Fungicide richness
## 1          V6     Conv.  R1      A         C        9
## 2          V6     Conv.  R1      B         C        6
## 3          V6     Conv.  R1      C         C        5
## 4          V6     Conv.  R1      A         F        7
## 5          V6     Conv.  R1      B         F        4
## 6          V6     Conv.  R1      C         F        2
```

```
head(filter(microbiome.fungi2, Treatment == "Conv." & Fungicide == "C"))
```

```
##                                   SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1CCB3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R2CAF3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R2CBG3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R2CCH3 Corn        Leaf     6/26/17
##   GrowthStage Treatment Rep Sample Fungicide richness
## 1          V6     Conv.  R1      A         C        9
## 2          V6     Conv.  R1      B         C        6
## 3          V6     Conv.  R1      C         C        5
## 4          V6     Conv.  R2      A         C        3
## 5          V6     Conv.  R2      B         C        8
## 6          V6     Conv.  R2      C         C        4
```

```
                                        #variable value & variable value

head(filter(microbiome.fungi2, Sample == "A" | Sample == "B"))
```

```
##                                   SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1FAC3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R1FBD3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R2CAF3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R2CBG3 Corn        Leaf     6/26/17
##   GrowthStage Treatment Rep Sample Fungicide richness
## 1          V6     Conv.  R1      A         C        9
## 2          V6     Conv.  R1      B         C        6
## 3          V6     Conv.  R1      A         F        7
## 4          V6     Conv.  R1      B         F        4
## 5          V6     Conv.  R2      A         C        3
## 6          V6     Conv.  R2      B         C        8
```

```
                                        #variable value OR variable value
```

Mutate Function `mutate()`

Example of Function mutate(data, new_column = any_function(old_column_to_manipulate))

```
microbiome.fungi2$logRich <- log(microbiome.fungi2$richness)
                          #Creating a new column called logRich using $ symbol

head(mutate(microbiome.fungi2, logRich = log(richness)))
```

```
##                                   SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1CCB3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R1FAC3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R1FBD3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R1FCE3 Corn        Leaf     6/26/17
##   GrowthStage Treatment Rep Sample Fungicide richness   logRich
## 1          V6     Conv.  R1      A         C        9 2.1972246
## 2          V6     Conv.  R1      B         C        6 1.7917595
## 3          V6     Conv.  R1      C         C        5 1.6094379
## 4          V6     Conv.  R1      A         F        7 1.9459101
## 5          V6     Conv.  R1      B         F        4 1.3862944
## 6          V6     Conv.  R1      C         F        2 0.6931472
```

```
head(mutate(microbiome.fungi2, Crop_Treatment = paste(Crop, Treatment)))
```

```
##                                   SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1CCB3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R1FAC3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R1FBD3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R1FCE3 Corn        Leaf     6/26/17
##   GrowthStage Treatment Rep Sample Fungicide richness   logRich Crop_Treatment
## 1          V6     Conv.  R1      A         C        9 2.1972246     Corn Conv.
## 2          V6     Conv.  R1      B         C        6 1.7917595     Corn Conv.
## 3          V6     Conv.  R1      C         C        5 1.6094379     Corn Conv.
## 4          V6     Conv.  R1      A         F        7 1.9459101     Corn Conv.
## 5          V6     Conv.  R1      B         F        4 1.3862944     Corn Conv.
## 6          V6     Conv.  R1      C         F        2 0.6931472     Corn Conv.
```

```
                        #Creating a new column that combines Crop and Treatment
```

Pipe Function %>%

Allows to combine multiple functions together to wrangle the data into a specific form for a new data frame. The data from the previous step is transferred to the next step.

```
select(microbiome.fungi, SampleID, Crop, Compartment:Fungicide, richness)
                                            #hid results to clean up document
```

```
microbiome.fungi %>%    #DATA
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
                                                        #Selecting out columns
```

```
  filter(Treatment == "Conv.") %>%
                              #Filtering out the data for a specific requirement
  mutate(logRich = log(richness)) %>%
                                  #Creating a new column for the log(richness)
  head()
```

```
##                                    SampleID Crop Compartment DateSampled
## 1 Corn2017LeafObjective2Collection1T1R1CAH2 Corn        Leaf     6/26/17
## 2 Corn2017LeafObjective2Collection1T1R1CBA3 Corn        Leaf     6/26/17
## 3 Corn2017LeafObjective2Collection1T1R1CCB3 Corn        Leaf     6/26/17
## 4 Corn2017LeafObjective2Collection1T1R1FAC3 Corn        Leaf     6/26/17
## 5 Corn2017LeafObjective2Collection1T1R1FBD3 Corn        Leaf     6/26/17
## 6 Corn2017LeafObjective2Collection1T1R1FCE3 Corn        Leaf     6/26/17
##   GrowthStage Treatment Rep Sample Fungicide richness   logRich
## 1          V6     Conv.  R1      A         C        9 2.1972246
## 2          V6     Conv.  R1      B         C        6 1.7917595
## 3          V6     Conv.  R1      C         C        5 1.6094379
## 4          V6     Conv.  R1      A         F        7 1.9459101
## 5          V6     Conv.  R1      B         F        4 1.3862944
## 6          V6     Conv.  R1      C         F        2 0.6931472
```

Summarise Function `summarise()`

Example of Function summarise(new_column_name = math_function(old_column))

The `summarise()` function allows for calculations like mean, standard deviation, and standard error.

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
                                                          #Selecting columns
  filter(Treatment == "Conv.") %>%
                          #Subsetting to only include the conventional treatment
  mutate(logRich = log(richness)) %>%
                                     #Creating a new column of the log richness
  summarise(Mean.rich = mean(logRich))  #Calculating overall mean log richness
```

```
##   Mean.rich
## 1  2.304395
```

Adding more lines of code to perform more calculations.

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
                                                  #Selecting Specific Columns
  filter(Treatment == "Conv.") %>%
  #Filtering the data where Treatment equals Conventional Treatment ("Conv.")
  mutate(logRich = log(richness)) %>%
                                #Again adding a new column of the log richness
  summarise(Mean.rich = mean(logRich),
                #Calculating the Mean, Standard Deviation, and Standard Error
            n = n(),
            sd.dev = sd(logRich)) %>%
  mutate(std.err = sd.dev/sqrt(n))
```

```
##   Mean.rich   n    sd.dev   std.err
## 1  2.304395 144 0.7024667 0.0585389
```

Group_by Function group_by()

Example of Group_by Function

dataframe %>% group_by (column, column)

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
  group_by(Treatment, Fungicide) %>%
                                #Groups the data by treatment and fungicide
  mutate(logRich = log(richness)) %>%
  summarise(Mean.rich = mean(logRich),
            n = n(),
            sd.dev = sd(logRich)) %>%
  mutate(std.err = sd.dev/sqrt(n))
```

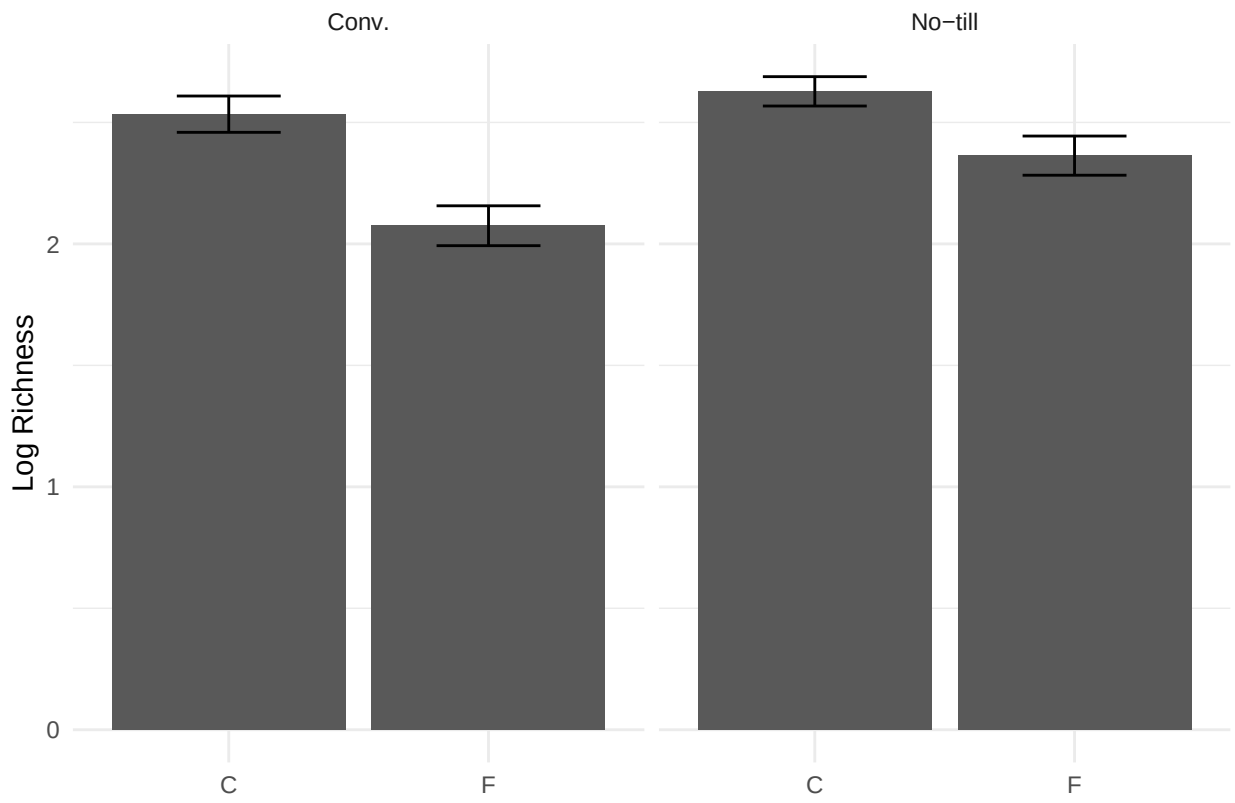
```
## `summarise()` has grouped output by 'Treatment'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 6
## # Groups:   Treatment [2]
##   Treatment Fungicide Mean.rich     n sd.dev std.err
##   <chr>     <chr>         <dbl> <int>  <dbl>   <dbl>
## 1 Conv.     C              2.53    72  0.635  0.0748
## 2 Conv.     F              2.07    72  0.696  0.0820
## 3 No-till   C              2.63    72  0.513  0.0604
## 4 No-till   F              2.36    71  0.680  0.0807
```

Connecting to Plotting Great for directly plotting into ggplot function. This will be great for project data and final project.

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
  group_by(Treatment, Fungicide) %>%
  mutate(logRich = log(richness)) %>%
  summarise(Mean.rich = mean(logRich),
            n = n(),
            sd.dev = sd(logRich)) %>%
  mutate(std.err = sd.dev/sqrt(n)) %>%
  ggplot(aes(x = Fungicide, y = Mean.rich)) +
        #Adding GGPLOT function, treated like normal, but not dataframe input
  geom_bar(stat="identity") +
  geom_errorbar( aes(x=Fungicide, ymin=Mean.rich-std.err,
                     ymax=Mean.rich+std.err), width=0.4) +
  theme_minimal() +
  xlab("") +
  ylab("Log Richness") +
  facet_wrap(~Treatment)
```

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



Joining Functions [Dplyr] (https://dplyr.tidyverse.org/reference/mutate-joins.html)

-`left_join()`: Keep all rows of X and add matching rows from Y. Any rows in Y that don't match X are excluded. -`right_join()`: reverse of left_join() -`inner_join()`: only keep rows that are common to both X AND Y, remove everything else. -`full_join()`: Keep any columns that are in either X or Y

Sample_n Function `sample_n()`

```
Richness <- microbiome.fungi %>%
  select(SampleID, richness)

Metadata <- microbiome.fungi %>%
  select(SampleID, Fungicide, Crop, Compartment, GrowthStage, Treatment,
         Rep, Sample)

head(Metadata)
```

```
##                                  SampleID Fungicide Crop Compartment
## 1 Corn2017LeafObjective2Collection1T1R1CAH2         C Corn        Leaf
## 2 Corn2017LeafObjective2Collection1T1R1CBA3         C Corn        Leaf
## 3 Corn2017LeafObjective2Collection1T1R1CCB3         C Corn        Leaf
## 4 Corn2017LeafObjective2Collection1T1R1FAC3         F Corn        Leaf
```

```
## 5 Corn2017LeafObjective2Collection1T1R1FBD3         F Corn        Leaf
## 6 Corn2017LeafObjective2Collection1T1R1FCE3         F Corn        Leaf
##   GrowthStage Treatment Rep Sample
## 1          V6     Conv.  R1      A
## 2          V6     Conv.  R1      B
## 3          V6     Conv.  R1      C
## 4          V6     Conv.  R1      A
## 5          V6     Conv.  R1      B
## 6          V6     Conv.  R1      C
```

```
head(Richness)
```

```
##                                    SampleID richness
## 1 Corn2017LeafObjective2Collection1T1R1CAH2        9
## 2 Corn2017LeafObjective2Collection1T1R1CBA3        6
## 3 Corn2017LeafObjective2Collection1T1R1CCB3        5
## 4 Corn2017LeafObjective2Collection1T1R1FAC3        7
## 5 Corn2017LeafObjective2Collection1T1R1FBD3        4
## 6 Corn2017LeafObjective2Collection1T1R1FCE3        2
```

```
head(left_join(Metadata, Richness, by = "SampleID"))
```

```
##                                    SampleID Fungicide Crop Compartment
## 1 Corn2017LeafObjective2Collection1T1R1CAH2         C Corn        Leaf
## 2 Corn2017LeafObjective2Collection1T1R1CBA3         C Corn        Leaf
## 3 Corn2017LeafObjective2Collection1T1R1CCB3         C Corn        Leaf
## 4 Corn2017LeafObjective2Collection1T1R1FAC3         F Corn        Leaf
## 5 Corn2017LeafObjective2Collection1T1R1FBD3         F Corn        Leaf
## 6 Corn2017LeafObjective2Collection1T1R1FCE3         F Corn        Leaf
##   GrowthStage Treatment Rep Sample richness
## 1          V6     Conv.  R1      A        9
## 2          V6     Conv.  R1      B        6
## 3          V6     Conv.  R1      C        5
## 4          V6     Conv.  R1      A        7
## 5          V6     Conv.  R1      B        4
## 6          V6     Conv.  R1      C        2
```

```
                    #Adding the richness data to the metadata based on on
                                          #the common column of sampleID
```

Pivoting Function

Pivoting is also useful for converting from wide to long format and back again. Functions called `pivot_longer()` and `pivot_wider()`

[Tidyverse] (https://tidyr.tidyverse.org/reference/pivot_wider.html)

Example of Function

`pivot_wider()`

Wide Format: sets the values within the fungicide column into column names

names_from and values_from: A pair of arguments describing which column (or columns) to get the name of the output column (names_from), and which column (or columns) to get the cell values from (values_from).

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
  group_by(Treatment, Fungicide) %>%
  summarise(Mean = mean(richness)) %>%
  pivot_wider(names_from = Fungicide, values_from = Mean) #Pivot to wide format
```

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

```
## # A tibble: 2 x 3
## # Groups:   Treatment [2]
##   Treatment     C     F
##   <chr>     <dbl> <dbl>
## 1 Conv.      14.6  9.75
## 2 No-till    15.4 13.1
```

Calculating the difference between the fungicide and control.

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
  group_by(Treatment, Fungicide) %>%
  summarise(Mean = mean(richness)) %>%
  pivot_wider(names_from = Fungicide, values_from = Mean) %>%
  mutate(diff.fungicide = C - F)
```

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

```
## # A tibble: 2 x 4
## # Groups:   Treatment [2]
##   Treatment     C     F diff.fungicide
##   <chr>     <dbl> <dbl>          <dbl>
## 1 Conv.      14.6  9.75           4.89
## 2 No-till    15.4 13.1            2.32
```

Now plotting the calculated data.

```
microbiome.fungi %>%
  select(SampleID, Crop, Compartment:Fungicide, richness) %>%
  group_by(Treatment, Fungicide) %>%
  summarise(Mean = mean(richness)) %>%
  pivot_wider(names_from = Fungicide, values_from = Mean) %>%
  mutate(diff.fungicide = C - F) %>%
  ggplot(aes(x = Treatment, y = diff.fungicide)) +
  geom_col() +
  theme_minimal() +
  xlab("") +
  ylab("Difference in average species richness")
```

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