

CodingChallenge5

Mamata K C

2025-03-20

Contents

1. 3 pts. Download two .csv files from Canvas called DiversityData.csv and Metadata.csv, and read them into R using relative file paths. 2
2. 4 pts. Join the two dataframes together by the common column 'Code'. Name the resulting dataframe alpha. 5
3. 4 pts. 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. 8
4. 4. Pts. 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. 11
5. 4. Pts. Calculate the difference between the soybean column, the soil column, and the difference between the cotton column and the soil column 11
6. 4 pts. Connecting it to plots 12
7. 2 pts. Commit and push a gfm .md file to GitHub inside a directory called Coding Challenge 5. Provide me a link to your github written as a clickable link in your .pdf or .docx 13

Data wrangling – 25 pts

PLEASE READ THIS BEFORE CONTINUING

This assignment will help you practice integrating some of the tidyverse functions into your R scripts. It will also involve some more practice with GitHub. You may collaborate with a partner to enhance your learning experience. Please ensure the following:

- Collaboration: If you work with a partner, include both names on the final submission by editing the YAML header.
- Submission: Only one person should submit the assignment to Canvas in a Word document or .pdf file generated through R markdown. Additionally, you should provide a link to your GitHub, where the assignment should be viewable by rendering it as a GitHub-flavored markdown file.
- Setup: It is also assumed you already have a GitHub repository for this class.
- Time: This should take you no longer than the class period to complete.

1. 3 pts. Download two .csv files from Canvas called Diversity-Data.csv and Metadata.csv, and read them into R using relative file paths.

```
DiversityData <- read.csv("DiversityData.csv") #loading data in R
DiversityData
```

##	Code	shannon	invsimpson	simpson	richness
## 1	S01_13	6.624921	210.72795	0.9952545	3319
## 2	S02_16	6.612413	206.86664	0.9951660	3079
## 3	S03_19	6.660853	213.01843	0.9953056	3935
## 4	S04_22	6.660671	204.69080	0.9951146	3922
## 5	S05_25	6.610965	200.25523	0.9950064	3196
## 6	S06_28	6.650812	199.32110	0.9949830	3481
## 7	S61_32	6.570679	200.23177	0.9950058	3250
## 8	S62_35	6.492227	171.27965	0.9941616	3170
## 9	S63_38	6.610986	192.08535	0.9947940	3657
## 10	S64_41	6.472259	163.99814	0.9939024	3177
## 11	S65_44	6.508824	181.69248	0.9944962	2985
## 12	S66_47	6.482495	176.90684	0.9943473	2770
## 13	S121_51	6.276073	126.56259	0.9920988	3040
## 14	S122_54	6.461118	152.98152	0.9934633	3192
## 15	S123_57	6.334648	138.92556	0.9928019	2673
## 16	S124_60	6.461988	171.13732	0.9941567	3180
## 17	S125_63	6.501973	172.97532	0.9942188	3320
## 18	S126_66	6.354387	142.61016	0.9929879	2773
## 19	S181_70	6.299381	142.64506	0.9929896	2806
## 20	S182_74	6.340644	145.48656	0.9931265	3047
## 21	S183_78	6.282807	150.39829	0.9933510	2190
## 22	S184_82	6.268316	141.14138	0.9929149	2488
## 23	S186_90	6.289000	140.45260	0.9928802	2684
## 24	C01_11	6.618126	220.66218	0.9954682	3076
## 25	C02_14	6.627206	211.03921	0.9952615	3180
## 26	C03_17	6.616958	216.06631	0.9953718	2938
## 27	C04_20	6.626465	215.93901	0.9953691	3371
## 28	C05_23	6.642822	211.08960	0.9952627	3435
## 29	C06_26	6.679131	216.31351	0.9953771	3629
## 30	C61_30	6.454741	170.03639	0.9941189	2767
## 31	C62_33	6.484032	172.35279	0.9941979	3377
## 32	C63_36	6.517958	173.41489	0.9942335	3804
## 33	C64_39	6.476069	167.13138	0.9940167	3204
## 34	C65_42	6.569722	197.01186	0.9949242	3250
## 35	C66_45	6.482145	172.96394	0.9942184	3009
## 36	C121_49	5.944568	71.55607	0.9860249	2779
## 37	C122_52	6.187755	96.43939	0.9896308	3193
## 38	C123_55	6.129460	81.26646	0.9876948	2859
## 39	C124_58	6.028523	75.49726	0.9867545	2950
## 40	C125_61	6.148179	98.94468	0.9898933	3018
## 41	C126_64	6.347332	150.05708	0.9933359	2946
## 42	C181_68	6.301392	132.36230	0.9924450	3266
## 43	C182_72	6.000205	83.90929	0.9880824	2969

## 44	C183_76	5.981284	82.44127	0.9878702	2636
## 45	C184_80	5.578566	50.73174	0.9802885	2043
## 46	C185_84	6.064655	87.82732	0.9886140	3113
## 47	SB01_12	6.644864	216.86110	0.9953888	3203
## 48	SB02_15	6.615662	211.32573	0.9952680	3055
## 49	SB03_18	6.693987	230.45439	0.9956607	3595
## 50	SB04_21	6.647502	234.80343	0.9957411	3253
## 51	SB05_24	6.605749	198.57265	0.9949641	3187
## 52	SB06_27	6.640696	215.26494	0.9953546	3190
## 53	SB61_31	6.044229	89.13912	0.9887816	2371
## 54	SB62_34	6.437589	154.21624	0.9935156	3248
## 55	SB63_37	6.194632	83.11681	0.9879687	2976
## 56	SB64_40	6.117393	87.20257	0.9885324	3006
## 57	SB65_43	5.439798	29.48338	0.9660826	2809
## 58	SB66_46	6.195816	108.22394	0.9907599	2680
## 59	SB121_50	4.393341	12.39587	0.9193280	2508
## 60	SB122_53	5.630929	52.97931	0.9811247	2403
## 61	SB123_56	5.579523	48.59842	0.9794232	2752
## 62	SB124_59	5.406651	34.08685	0.9706632	2946
## 63	SB125_62	5.863941	63.33020	0.9842097	3165
## 64	SB126_65	5.738025	57.88780	0.9827252	2705
## 65	SB181_69	5.671024	57.37726	0.9825715	2642
## 66	SB182_73	5.489406	43.16854	0.9768350	2773
## 67	SB183_77	5.713960	60.47882	0.9834653	2454
## 68	SB184_81	5.467076	44.06798	0.9773078	2365
## 69	SB185_85	5.729473	55.95864	0.9821297	2789
## 70	SB186_89	5.556356	54.34527	0.9815991	2050

```
Metadata <- read.csv("Metadata.csv",na.strings ="na") #loading data in R
Metadata
```

##	Code	Crop	Time_Point	Replicate	Water_Imbibed
## 1	S01_13	Soil	0	1	NA
## 2	S02_16	Soil	0	2	NA
## 3	S03_19	Soil	0	3	NA
## 4	S04_22	Soil	0	4	NA
## 5	S05_25	Soil	0	5	NA
## 6	S06_28	Soil	0	6	NA
## 7	S61_32	Soil	6	1	NA
## 8	S62_35	Soil	6	2	NA
## 9	S63_38	Soil	6	3	NA
## 10	S64_41	Soil	6	4	NA
## 11	S65_44	Soil	6	5	NA
## 12	S66_47	Soil	6	6	NA
## 13	S121_51	Soil	12	1	NA
## 14	S122_54	Soil	12	2	NA
## 15	S123_57	Soil	12	3	NA
## 16	S124_60	Soil	12	4	NA
## 17	S125_63	Soil	12	5	NA
## 18	S126_66	Soil	12	6	NA
## 19	S181_70	Soil	18	1	NA
## 20	S182_74	Soil	18	2	NA
## 21	S183_78	Soil	18	3	NA
## 22	S184_82	Soil	18	4	NA

## 23	S186_90	Soil	18	6	NA
## 24	C01_11	Cotton	0	1	0.0042
## 25	C02_14	Cotton	0	2	0.0091
## 26	C03_17	Cotton	0	3	0.0013
## 27	C04_20	Cotton	0	4	0.0087
## 28	C05_23	Cotton	0	5	0.0075
## 29	C06_26	Cotton	0	6	0.0046
## 30	C61_30	Cotton	6	1	0.0580
## 31	C62_33	Cotton	6	2	0.0440
## 32	C63_36	Cotton	6	3	0.0569
## 33	C64_39	Cotton	6	4	0.0841
## 34	C65_42	Cotton	6	5	0.0535
## 35	C66_45	Cotton	6	6	0.0029
## 36	C121_49	Cotton	12	1	0.0651
## 37	C122_52	Cotton	12	2	0.0527
## 38	C123_55	Cotton	12	3	0.0675
## 39	C124_58	Cotton	12	4	0.0545
## 40	C125_61	Cotton	12	5	0.0623
## 41	C126_64	Cotton	12	6	0.0021
## 42	C181_68	Cotton	18	1	0.0034
## 43	C182_72	Cotton	18	2	0.0632
## 44	C183_76	Cotton	18	3	0.0514
## 45	C184_80	Cotton	18	4	0.0577
## 46	C185_84	Cotton	18	5	0.0554
## 47	SB01_12	Soybean	0	1	0.1664
## 48	SB02_15	Soybean	0	2	0.0942
## 49	SB03_18	Soybean	0	3	0.1248
## 50	SB04_21	Soybean	0	4	0.1150
## 51	SB05_24	Soybean	0	5	0.0993
## 52	SB06_27	Soybean	0	6	0.1005
## 53	SB61_31	Soybean	6	1	0.2308
## 54	SB62_34	Soybean	6	2	0.2603
## 55	SB63_37	Soybean	6	3	0.2111
## 56	SB64_40	Soybean	6	4	0.2808
## 57	SB65_43	Soybean	6	5	0.2712
## 58	SB66_46	Soybean	6	6	0.2887
## 59	SB121_50	Soybean	12	1	0.2822
## 60	SB122_53	Soybean	12	2	0.2557
## 61	SB123_56	Soybean	12	3	0.2982
## 62	SB124_59	Soybean	12	4	0.2489
## 63	SB125_62	Soybean	12	5	0.2573
## 64	SB126_65	Soybean	12	6	0.2285
## 65	SB181_69	Soybean	18	1	0.2528
## 66	SB182_73	Soybean	18	2	0.2706
## 67	SB183_77	Soybean	18	3	0.3196
## 68	SB184_81	Soybean	18	4	0.2437
## 69	SB185_85	Soybean	18	5	0.2461
## 70	SB186_89	Soybean	18	6	0.3010

2. 4 pts. Join the two dataframes together by the common column 'Code'. Name the resulting dataframe alpha.

```
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
```

```
library(ggplot2)
alpha <- left_join(DiversityData, Metadata, by = "Code") #combining two dataframes together by common c
alpha
```

	Code	shannon	invsimpson	simpson	richness	Crop	Time_Point	Replicate
## 1	S01_13	6.624921	210.72795	0.9952545	3319	Soil	0	1
## 2	S02_16	6.612413	206.86664	0.9951660	3079	Soil	0	2
## 3	S03_19	6.660853	213.01843	0.9953056	3935	Soil	0	3
## 4	S04_22	6.660671	204.69080	0.9951146	3922	Soil	0	4
## 5	S05_25	6.610965	200.25523	0.9950064	3196	Soil	0	5
## 6	S06_28	6.650812	199.32110	0.9949830	3481	Soil	0	6
## 7	S61_32	6.570679	200.23177	0.9950058	3250	Soil	6	1
## 8	S62_35	6.492227	171.27965	0.9941616	3170	Soil	6	2
## 9	S63_38	6.610986	192.08535	0.9947940	3657	Soil	6	3
## 10	S64_41	6.472259	163.99814	0.9939024	3177	Soil	6	4
## 11	S65_44	6.508824	181.69248	0.9944962	2985	Soil	6	5
## 12	S66_47	6.482495	176.90684	0.9943473	2770	Soil	6	6
## 13	S121_51	6.276073	126.56259	0.9920988	3040	Soil	12	1
## 14	S122_54	6.461118	152.98152	0.9934633	3192	Soil	12	2
## 15	S123_57	6.334648	138.92556	0.9928019	2673	Soil	12	3
## 16	S124_60	6.461988	171.13732	0.9941567	3180	Soil	12	4
## 17	S125_63	6.501973	172.97532	0.9942188	3320	Soil	12	5
## 18	S126_66	6.354387	142.61016	0.9929879	2773	Soil	12	6
## 19	S181_70	6.299381	142.64506	0.9929896	2806	Soil	18	1
## 20	S182_74	6.340644	145.48656	0.9931265	3047	Soil	18	2
## 21	S183_78	6.282807	150.39829	0.9933510	2190	Soil	18	3
## 22	S184_82	6.268316	141.14138	0.9929149	2488	Soil	18	4
## 23	S186_90	6.289000	140.45260	0.9928802	2684	Soil	18	6
## 24	C01_11	6.618126	220.66218	0.9954682	3076	Cotton	0	1
## 25	C02_14	6.627206	211.03921	0.9952615	3180	Cotton	0	2
## 26	C03_17	6.616958	216.06631	0.9953718	2938	Cotton	0	3
## 27	C04_20	6.626465	215.93901	0.9953691	3371	Cotton	0	4
## 28	C05_23	6.642822	211.08960	0.9952627	3435	Cotton	0	5
## 29	C06_26	6.679131	216.31351	0.9953771	3629	Cotton	0	6
## 30	C61_30	6.454741	170.03639	0.9941189	2767	Cotton	6	1

## 31	C62_33	6.484032	172.35279	0.9941979	3377	Cotton	6	2
## 32	C63_36	6.517958	173.41489	0.9942335	3804	Cotton	6	3
## 33	C64_39	6.476069	167.13138	0.9940167	3204	Cotton	6	4
## 34	C65_42	6.569722	197.01186	0.9949242	3250	Cotton	6	5
## 35	C66_45	6.482145	172.96394	0.9942184	3009	Cotton	6	6
## 36	C121_49	5.944568	71.55607	0.9860249	2779	Cotton	12	1
## 37	C122_52	6.187755	96.43939	0.9896308	3193	Cotton	12	2
## 38	C123_55	6.129460	81.26646	0.9876948	2859	Cotton	12	3
## 39	C124_58	6.028523	75.49726	0.9867545	2950	Cotton	12	4
## 40	C125_61	6.148179	98.94468	0.9898933	3018	Cotton	12	5
## 41	C126_64	6.347332	150.05708	0.9933359	2946	Cotton	12	6
## 42	C181_68	6.301392	132.36230	0.9924450	3266	Cotton	18	1
## 43	C182_72	6.000205	83.90929	0.9880824	2969	Cotton	18	2
## 44	C183_76	5.981284	82.44127	0.9878702	2636	Cotton	18	3
## 45	C184_80	5.578566	50.73174	0.9802885	2043	Cotton	18	4
## 46	C185_84	6.064655	87.82732	0.9886140	3113	Cotton	18	5
## 47	SB01_12	6.644864	216.86110	0.9953888	3203	Soybean	0	1
## 48	SB02_15	6.615662	211.32573	0.9952680	3055	Soybean	0	2
## 49	SB03_18	6.693987	230.45439	0.9956607	3595	Soybean	0	3
## 50	SB04_21	6.647502	234.80343	0.9957411	3253	Soybean	0	4
## 51	SB05_24	6.605749	198.57265	0.9949641	3187	Soybean	0	5
## 52	SB06_27	6.640696	215.26494	0.9953546	3190	Soybean	0	6
## 53	SB61_31	6.044229	89.13912	0.9887816	2371	Soybean	6	1
## 54	SB62_34	6.437589	154.21624	0.9935156	3248	Soybean	6	2
## 55	SB63_37	6.194632	83.11681	0.9879687	2976	Soybean	6	3
## 56	SB64_40	6.117393	87.20257	0.9885324	3006	Soybean	6	4
## 57	SB65_43	5.439798	29.48338	0.9660826	2809	Soybean	6	5
## 58	SB66_46	6.195816	108.22394	0.9907599	2680	Soybean	6	6
## 59	SB121_50	4.393341	12.39587	0.9193280	2508	Soybean	12	1
## 60	SB122_53	5.630929	52.97931	0.9811247	2403	Soybean	12	2
## 61	SB123_56	5.579523	48.59842	0.9794232	2752	Soybean	12	3
## 62	SB124_59	5.406651	34.08685	0.9706632	2946	Soybean	12	4
## 63	SB125_62	5.863941	63.33020	0.9842097	3165	Soybean	12	5
## 64	SB126_65	5.738025	57.88780	0.9827252	2705	Soybean	12	6
## 65	SB181_69	5.671024	57.37726	0.9825715	2642	Soybean	18	1
## 66	SB182_73	5.489406	43.16854	0.9768350	2773	Soybean	18	2
## 67	SB183_77	5.713960	60.47882	0.9834653	2454	Soybean	18	3
## 68	SB184_81	5.467076	44.06798	0.9773078	2365	Soybean	18	4
## 69	SB185_85	5.729473	55.95864	0.9821297	2789	Soybean	18	5
## 70	SB186_89	5.556356	54.34527	0.9815991	2050	Soybean	18	6
##	Water_Imbibed							
## 1	NA							
## 2	NA							
## 3	NA							
## 4	NA							
## 5	NA							
## 6	NA							
## 7	NA							
## 8	NA							
## 9	NA							
## 10	NA							
## 11	NA							
## 12	NA							
## 13	NA							

## 14	NA
## 15	NA
## 16	NA
## 17	NA
## 18	NA
## 19	NA
## 20	NA
## 21	NA
## 22	NA
## 23	NA
## 24	0.0042
## 25	0.0091
## 26	0.0013
## 27	0.0087
## 28	0.0075
## 29	0.0046
## 30	0.0580
## 31	0.0440
## 32	0.0569
## 33	0.0841
## 34	0.0535
## 35	0.0029
## 36	0.0651
## 37	0.0527
## 38	0.0675
## 39	0.0545
## 40	0.0623
## 41	0.0021
## 42	0.0034
## 43	0.0632
## 44	0.0514
## 45	0.0577
## 46	0.0554
## 47	0.1664
## 48	0.0942
## 49	0.1248
## 50	0.1150
## 51	0.0993
## 52	0.1005
## 53	0.2308
## 54	0.2603
## 55	0.2111
## 56	0.2808
## 57	0.2712
## 58	0.2887
## 59	0.2822
## 60	0.2557
## 61	0.2982
## 62	0.2489
## 63	0.2573
## 64	0.2285
## 65	0.2528
## 66	0.2706
## 67	0.3196

```
## 68      0.2437
## 69      0.2461
## 70      0.3010
```

3. 4 pts. 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 <- alpha %>%
mutate(logRich = log(richness)) %>% #adding a column named logRich containing log values of richness c
mutate(alpha, Pielousevennessindex = shannon/logRich) #adding a column Pielousevennessindex by calculat

alpha_even
```

##	Code	shannon	invsimpson	simpson	richness	Crop	Time_Point	Replicate
## 1	S01_13	6.624921	210.72795	0.9952545	3319	Soil	0	1
## 2	S02_16	6.612413	206.86664	0.9951660	3079	Soil	0	2
## 3	S03_19	6.660853	213.01843	0.9953056	3935	Soil	0	3
## 4	S04_22	6.660671	204.69080	0.9951146	3922	Soil	0	4
## 5	S05_25	6.610965	200.25523	0.9950064	3196	Soil	0	5
## 6	S06_28	6.650812	199.32110	0.9949830	3481	Soil	0	6
## 7	S61_32	6.570679	200.23177	0.9950058	3250	Soil	6	1
## 8	S62_35	6.492227	171.27965	0.9941616	3170	Soil	6	2
## 9	S63_38	6.610986	192.08535	0.9947940	3657	Soil	6	3
## 10	S64_41	6.472259	163.99814	0.9939024	3177	Soil	6	4
## 11	S65_44	6.508824	181.69248	0.9944962	2985	Soil	6	5
## 12	S66_47	6.482495	176.90684	0.9943473	2770	Soil	6	6
## 13	S121_51	6.276073	126.56259	0.9920988	3040	Soil	12	1
## 14	S122_54	6.461118	152.98152	0.9934633	3192	Soil	12	2
## 15	S123_57	6.334648	138.92556	0.9928019	2673	Soil	12	3
## 16	S124_60	6.461988	171.13732	0.9941567	3180	Soil	12	4
## 17	S125_63	6.501973	172.97532	0.9942188	3320	Soil	12	5
## 18	S126_66	6.354387	142.61016	0.9929879	2773	Soil	12	6
## 19	S181_70	6.299381	142.64506	0.9929896	2806	Soil	18	1
## 20	S182_74	6.340644	145.48656	0.9931265	3047	Soil	18	2
## 21	S183_78	6.282807	150.39829	0.9933510	2190	Soil	18	3
## 22	S184_82	6.268316	141.14138	0.9929149	2488	Soil	18	4
## 23	S186_90	6.289000	140.45260	0.9928802	2684	Soil	18	6
## 24	C01_11	6.618126	220.66218	0.9954682	3076	Cotton	0	1
## 25	C02_14	6.627206	211.03921	0.9952615	3180	Cotton	0	2
## 26	C03_17	6.616958	216.06631	0.9953718	2938	Cotton	0	3
## 27	C04_20	6.626465	215.93901	0.9953691	3371	Cotton	0	4
## 28	C05_23	6.642822	211.08960	0.9952627	3435	Cotton	0	5
## 29	C06_26	6.679131	216.31351	0.9953771	3629	Cotton	0	6
## 30	C61_30	6.454741	170.03639	0.9941189	2767	Cotton	6	1
## 31	C62_33	6.484032	172.35279	0.9941979	3377	Cotton	6	2
## 32	C63_36	6.517958	173.41489	0.9942335	3804	Cotton	6	3
## 33	C64_39	6.476069	167.13138	0.9940167	3204	Cotton	6	4

## 34	C65_42	6.569722	197.01186	0.9949242	3250	Cotton	6	5
## 35	C66_45	6.482145	172.96394	0.9942184	3009	Cotton	6	6
## 36	C121_49	5.944568	71.55607	0.9860249	2779	Cotton	12	1
## 37	C122_52	6.187755	96.43939	0.9896308	3193	Cotton	12	2
## 38	C123_55	6.129460	81.26646	0.9876948	2859	Cotton	12	3
## 39	C124_58	6.028523	75.49726	0.9867545	2950	Cotton	12	4
## 40	C125_61	6.148179	98.94468	0.9898933	3018	Cotton	12	5
## 41	C126_64	6.347332	150.05708	0.9933359	2946	Cotton	12	6
## 42	C181_68	6.301392	132.36230	0.9924450	3266	Cotton	18	1
## 43	C182_72	6.000205	83.90929	0.9880824	2969	Cotton	18	2
## 44	C183_76	5.981284	82.44127	0.9878702	2636	Cotton	18	3
## 45	C184_80	5.578566	50.73174	0.9802885	2043	Cotton	18	4
## 46	C185_84	6.064655	87.82732	0.9886140	3113	Cotton	18	5
## 47	SB01_12	6.644864	216.86110	0.9953888	3203	Soybean	0	1
## 48	SB02_15	6.615662	211.32573	0.9952680	3055	Soybean	0	2
## 49	SB03_18	6.693987	230.45439	0.9956607	3595	Soybean	0	3
## 50	SB04_21	6.647502	234.80343	0.9957411	3253	Soybean	0	4
## 51	SB05_24	6.605749	198.57265	0.9949641	3187	Soybean	0	5
## 52	SB06_27	6.640696	215.26494	0.9953546	3190	Soybean	0	6
## 53	SB61_31	6.044229	89.13912	0.9887816	2371	Soybean	6	1
## 54	SB62_34	6.437589	154.21624	0.9935156	3248	Soybean	6	2
## 55	SB63_37	6.194632	83.11681	0.9879687	2976	Soybean	6	3
## 56	SB64_40	6.117393	87.20257	0.9885324	3006	Soybean	6	4
## 57	SB65_43	5.439798	29.48338	0.9660826	2809	Soybean	6	5
## 58	SB66_46	6.195816	108.22394	0.9907599	2680	Soybean	6	6
## 59	SB121_50	4.393341	12.39587	0.9193280	2508	Soybean	12	1
## 60	SB122_53	5.630929	52.97931	0.9811247	2403	Soybean	12	2
## 61	SB123_56	5.579523	48.59842	0.9794232	2752	Soybean	12	3
## 62	SB124_59	5.406651	34.08685	0.9706632	2946	Soybean	12	4
## 63	SB125_62	5.863941	63.33020	0.9842097	3165	Soybean	12	5
## 64	SB126_65	5.738025	57.88780	0.9827252	2705	Soybean	12	6
## 65	SB181_69	5.671024	57.37726	0.9825715	2642	Soybean	18	1
## 66	SB182_73	5.489406	43.16854	0.9768350	2773	Soybean	18	2
## 67	SB183_77	5.713960	60.47882	0.9834653	2454	Soybean	18	3
## 68	SB184_81	5.467076	44.06798	0.9773078	2365	Soybean	18	4
## 69	SB185_85	5.729473	55.95864	0.9821297	2789	Soybean	18	5
## 70	SB186_89	5.556356	54.34527	0.9815991	2050	Soybean	18	6
##	Water_Imbibed	logRich	Pielousevennessindex					
## 1	NA	8.107419		0.8171431				
## 2	NA	8.032360		0.8232216				
## 3	NA	8.277666		0.8046776				
## 4	NA	8.274357		0.8049774				
## 5	NA	8.069655		0.8192376				
## 6	NA	8.155075		0.8155427				
## 7	NA	8.086410		0.8125582				
## 8	NA	8.061487		0.8053387				
## 9	NA	8.204398		0.8057856				
## 10	NA	8.063693		0.8026420				
## 11	NA	8.001355		0.8134652				
## 12	NA	7.926603		0.8178151				
## 13	NA	8.019613		0.7825905				
## 14	NA	8.068403		0.8007927				
## 15	NA	7.890957		0.8027732				
## 16	NA	8.064636		0.8012745				

## 17	NA	8.107720	0.8019483
## 18	NA	7.927685	0.8015438
## 19	NA	7.939515	0.7934213
## 20	NA	8.021913	0.7904154
## 21	NA	7.691657	0.8168340
## 22	NA	7.819234	0.8016534
## 23	NA	7.895063	0.7965737
## 24	0.0042	8.031385	0.8240330
## 25	0.0091	8.064636	0.8217613
## 26	0.0013	7.985484	0.8286233
## 27	0.0087	8.122965	0.8157692
## 28	0.0075	8.141772	0.8158938
## 29	0.0046	8.196712	0.8148549
## 30	0.0580	7.925519	0.8144250
## 31	0.0440	8.124743	0.7980600
## 32	0.0569	8.243808	0.7906489
## 33	0.0841	8.072155	0.8022726
## 34	0.0535	8.086410	0.8124399
## 35	0.0029	8.009363	0.8093209
## 36	0.0651	7.929846	0.7496447
## 37	0.0527	8.068716	0.7668822
## 38	0.0675	7.958227	0.7702042
## 39	0.0545	7.989560	0.7545500
## 40	0.0623	8.012350	0.7673379
## 41	0.0021	7.988204	0.7945881
## 42	0.0034	8.091321	0.7787840
## 43	0.0632	7.995980	0.7504026
## 44	0.0514	7.877018	0.7593336
## 45	0.0577	7.622175	0.7318864
## 46	0.0554	8.043342	0.7539969
## 47	0.1664	8.071843	0.8232153
## 48	0.0942	8.024535	0.8244294
## 49	0.1248	8.187299	0.8176063
## 50	0.1150	8.087333	0.8219646
## 51	0.0993	8.066835	0.8188774
## 52	0.1005	8.067776	0.8231136
## 53	0.2308	7.771067	0.7777862
## 54	0.2603	8.085795	0.7961603
## 55	0.2111	7.998335	0.7744902
## 56	0.2808	8.008366	0.7638754
## 57	0.2712	7.940584	0.6850627
## 58	0.2887	7.893572	0.7849191
## 59	0.2822	7.827241	0.5612885
## 60	0.2557	7.784473	0.7233538
## 61	0.2982	7.920083	0.7044778
## 62	0.2489	7.988204	0.6768294
## 63	0.2573	8.059908	0.7275444
## 64	0.2285	7.902857	0.7260697
## 65	0.2528	7.879291	0.7197378
## 66	0.2706	7.927685	0.6924349
## 67	0.3196	7.805475	0.7320451
## 68	0.2437	7.768533	0.7037462
## 69	0.2461	7.933438	0.7221929
## 70	0.3010	7.625595	0.7286456

4. 4. Pts. 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 <- alpha_even %>%
  group_by(Crop, Time_Point) %>%      #grouping the data by Crop and Time point
  summarise(Mean = mean(Pielousevennessindex), #calculating mean of evenness by group
            n = n(), #counting number of rows based on group
            sd.dev = sd(Pielousevennessindex)) %>% #calculating standard deviation of the evenness by
  mutate(std.err = sd.dev/sqrt(n)) #calculating standard error by group
```

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

```
alpha_average
```

```
## # A tibble: 12 x 6
## # Groups:   Crop [3]
##   Crop    Time_Point Mean      n sd.dev std.err
##   <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. 4. Pts. 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.

- e. Name the resulting dataframe `alpha_average2`

```
alpha_average2 <- alpha_average %>%  
  select(Time_Point, Crop, Mean) %>% #selecting the relevant columns  
  pivot_wider(names_from = Crop, values_from = Mean) %>% #transforming data from longer format to wide  
  mutate(diff.cotton.even = Soil - Cotton) %>% #calculating difference in mean between soil and cotton  
  mutate(diff.soybean.even = Soil - Soybean) #calculating difference in mean between soil and soybean  
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
```

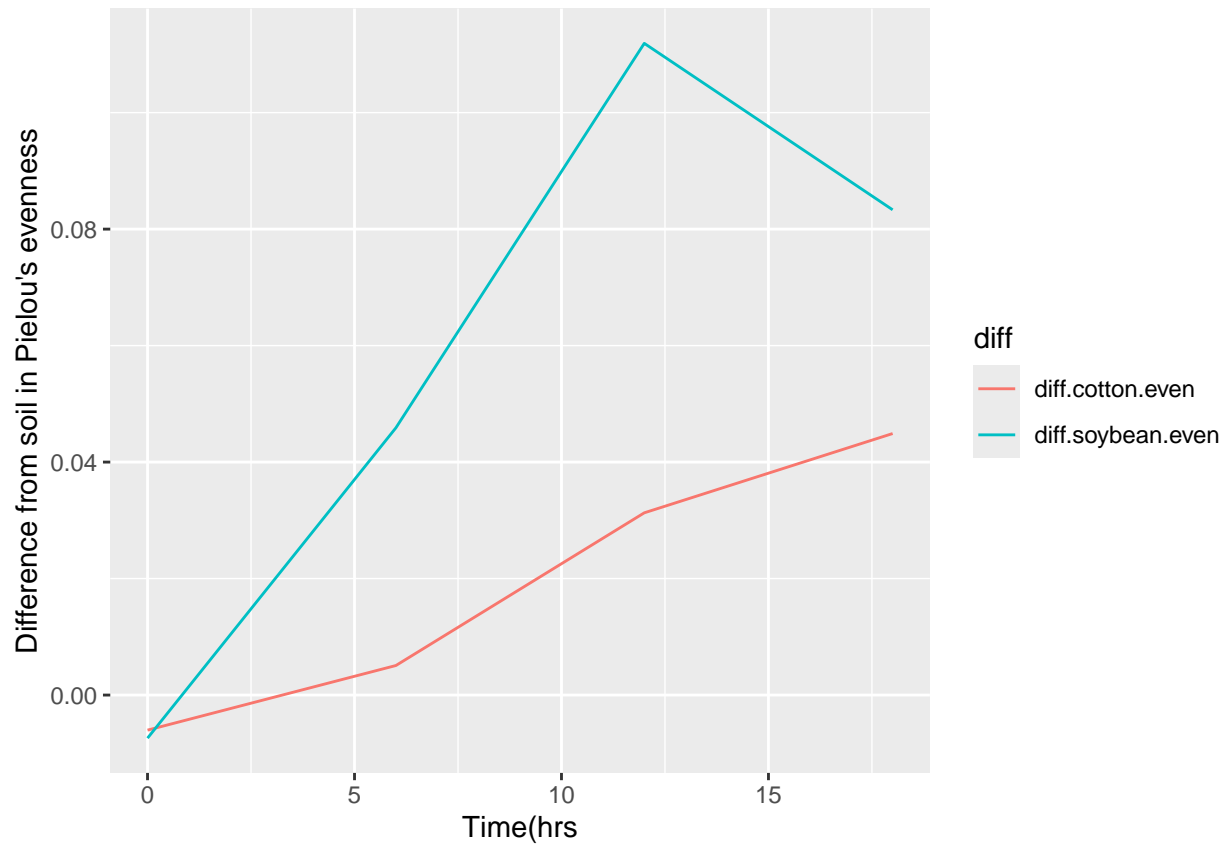
6. 4 pts. Connecting it to plots

- 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`.
- This might be challenging, so I'll give you a break. The code is below.

```
pivot_longer(c(diff.cotton.even, diff.soybean.even), names_to = "diff")
```

- 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) %>% #selecting relevant columns  
  pivot_longer(c(diff.cotton.even, diff.soybean.even), names_to = "diff") %>% #transforming wide format to long  
  ggplot(aes(x= Time_Point, y = value, color = diff))+ #creating a line graph with time point as x axis  
  geom_line()+  
  xlab("Time(hrs)") + #labeling x axis  
  ylab("Difference from soil in Pielou's evenness") #labeling y axis
```



7. 2 pts. Commit and push a gfm .md file to GitHub inside a directory called Coding Challenge 5. Provide me a link to your github written as a clickable link in your .pdf or .docx

Link to my GitHub