---

title: "slash veg complete"

author: "Peter Smalldige"

date: "`r Sys.Date()`"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

<!-- README

File Description: The purpose of this file is to:

1. read the raw vegetation data and make it complete

2. convert the field measurements to per/acre values

3. calculate the number of stems for each clump at a point

exported data files include:

xxx (line 237)

xxx (line 353)

xxx (line 530)

xxx

-->

### Import and cleaning of slash wall regeneation & plot data - 2023

###Initialize Libraries

```{r }

#library(ggplot2)

#library(ggpubr)

library(expss)

library(readr)

library(readxl)

library(writexl)

library(skimr)

library(tidyverse) # includes: tidyr, dplyr, haven, readr, readxl, forcats, stringer, ggplot2. See all with tidyverse\_packages()

library(knitr)

library(janitor)

#install.packages("openxlsx")

library(openxlsx)

# install.packages("shinyaddins")

# library(shinyaddins)

```

#

# ALT plus hypen = <- (within chunk)

# ctrl plus shift plus m = %>%

# ctrl + ALT + I = insert chunk

# ALT + - = insert assignment operator

# filter rows, select columns

# RMD comment = <!-- comment -->

#

#

<!--

-->

<!-- -->

### code 01

### Create file path input and the output folder (code 01)

file paths, input and output

```{r}

#file\_path\_input01 <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\regen\_per\_acre\_all\_alpha\_tidy\_trmt\_coded 2019-2022 (with read me).xlsx"

file\_path\_2023\_veg<- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\summary output\\complete\_full\_2023(readme).xlsx"

file\_path\_2019 <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\regen.data.2019.xlsx"

# line 93

file\_path\_2020 <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\regen.data.2020\_cleaned.xlsx"

# line 143

file\_path\_2021 <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\regen.data.2021\_cleaned.xlsx"

# line 179

file\_path\_2022 <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\slash.wall.data.2022.11february2025.xlsx"

# line 221

#

# slash.wall.data.2022.11february2025.xlsx has worksheets for

# site-plot = non cafri plot data, all data has been entered

# growth-height = seedling height data from cafri and non-cafri sites

# regen = regeneration data from cafri and non-cafri sites, but not yet 6\_6 as of 11feb2025

# site-traits = cafri plot data. only 6\_9 as of 11feb2025

file\_path\_points <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\location\_by\_point.xlsx"

#plot data for the 4" to 12" seedling data

file\_path\_2019\_plot <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\plot-data-2019.csv"

file\_path\_2020\_plot <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\plot\_calc\_2020\_untidy.csv"

file\_path\_2021\_plot <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\slash.wall.data.2021.30october2022.xlsx"

file\_path\_2022\_plot <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\old files but do not discard\\plot\_data\_2022\_untidy.csv"

# does not include any CAFRI data

file\_path\_2023\_plot <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\slash.wall.data.2023.23july2024(maybe FINAL).xlsx"

output\_folder <- "C:\\Users\\pjs23\\Documents\\R\\slash-wall-vegetation\\summary output\\"

```

In 2019 - 2022, seedling regen data were recorded in the following strata:

Site/plot sheet: counts of seedlings 4" to 12", sometimes as categorical

Regen Tally sheet:

12" to 4.5ft (B) = formerly seed01

4.5ft - 9 ft (C1) = formerly seed02

9ft to 1"(C2) [C1 + C2 = D], formerly seed03

1 - 4", formerly sapl

in 2023 and beyond, seedling regen data were recorded in the following strata:

Regen tally sheet:

(A = 01) 4" to 12"

(B = 02) 12" to 60" (the break point at 60" vs. 54" was to align with SILVAH protocols)

(D = 03) 60" to 1" dbh

(P = 04) 1" to 4" dbh

To standardize labels:

A = 4" to 12" radius = 3.7 ft; count in 2019, 2023, and 2024 otherwise categorical 0 = 0; 1 = 1to5; 2 = 6 to 15; 3 = 16 to 30; 4 = 31+

B = 12" to 4.5/5 ft radius = 6 ft

C1 = 4.5 ft to 9 ft radius = 6 ft

C2 = 9ft to 1" dbh radius = 6 ft

D = 4.5/5 ft to 1" radius = 6 ft

P = 1" to 4" (saplings) radius thru 2022 = 6 ft; then 16 ft

Radii

4" - 12" = 3.7 ft

12" to 4.5/5ft = 6 ft

plot data for seedlings 4" to 12" (recorded as 01 in 2023)

folder = R\\slash-wall-vegetation\\old files but do not discard

2019 file = plot-data-2019.csv

2020 file = plot\_calc\_2020\_untidy.csv

2021 file = plot\_data\_2021\_untidy.csv

2022 file = plot\_data\_2022\_untidy.csv

2023 files =

slash.wall.data.2023.23july2024(maybe FINAL)

slash.wall.data.2023\_1\_feb\_2024\_regen(north, south, rectangle, gasline)\_final

input point attributes, "point by location"

```{r}

point\_labels <- read\_excel(file\_path\_points, sheet = "point\_by\_location") %>%

rename(trmt = wall) %>%

select(-harvest.1) %>%

mutate(

height = 10,

height = if\_else(location == "c", 0, height) #if location = c, then height is set to 0, otherwise it remains as "height"

)%>%

mutate(

point = as.character(point),

harvest = case\_when(

harvest == "b" ~ "sta\_rd",

harvest == "boot" ~ "sta\_rd",

harvest == "cgl" ~ "gas\_line",

harvest == "circle1" ~ "circle",

harvest == "circle2" ~ "circle",

harvest == "circle3" ~ "circle",

harvest == "cricle1" ~ "circle",

harvest == "crp" ~ "red\_pine",

harvest == "cwb" ~ "sta\_rd",

harvest == "gasline" ~ "gas\_line",

harvest == "gl" ~ "gas\_line",

harvest == "recksouth" ~ "recknagel",

harvest == "rect1" ~ "recknagel",

harvest == "rect2" ~ "recknagel",

harvest == "rect3" ~ "recknagel",

harvest == "rect4" ~ "recknagel",

harvest == "rectangle1" ~ "recknagel",

harvest == "rectangle2" ~ "recknagel",

harvest == "rectangle3" ~ "recknagel",

harvest == "rectangle4" ~ "recknagel",

harvest == "rp" ~ "red\_pine",

harvest == "redpine" ~ "red\_pine",

harvest == "stationrd" ~ "sta\_rd",

harvest == "w" ~ "sta\_rd",

harvest == "wb" ~ "sta\_rd",

harvest == "wedge" ~ "sta\_rd",

harvest == "old" ~ "new",

TRUE ~ harvest # Keeps other values unchanged

)

)

```

2019 data

```{r eval=T}

raw2019 <- read\_excel(file\_path\_2019, sheet = "seedling", skip = 0 ) |>

select(! c(top4, other, radius.01, radius.other)) |>

mutate(

season = "2019",

sup.seed03 = as.numeric(sup.seed03),

sup.sapl = as.numeric(sup.sapl),

point = as.character(point)

) |>

mutate(origin = case\_when(

origin %in% c("s","u") ~ "single",

origin %in% c("t", "r") ~ "clump",

TRUE ~ origin # Keeps other values unchanged

)) |>

select(harvest, season, point, location, spp, origin, everything()) %>%

mutate(

harvest = case\_when(

harvest == "cgl" ~ "gasline",

harvest == "crp" ~ "redpine",

harvest == "cwb" ~ "stationrd",

harvest == "gl" ~ "gasline",

harvest == "rp" ~ "redpine",

TRUE ~ harvest # Keeps other values unchanged

)

) %>%

mutate(

location = case\_when(

location == "control" ~ "c",

location == "interior" ~ "i",

location == "perimeter" ~ "p",

TRUE ~ harvest # Keeps other values unchanged

)

)

#separate the clumps to create a variable count for calculation of tpa and also to calculate ramet (number sprouts / clump)

# The first "pivot\_longer" was to replicate rows that had multiple data values. In some years, multiple clumps were

# recorded on the same row on the tally sheet.

#

# The "pivot\_wider" recreates the tally sheet format to allow the data to be made complete.

clump2019 <- raw2019 |>

filter(origin == "clump") |>

mutate(

count = 1,

numb.clumps = rowSums(across(c(starts\_with("sup"), starts\_with("exp")), ~ if\_else(. > 0, 1, 0)), na.rm = TRUE)

)

#

# pivot\_longer(cols = matches("^(su|ex)"), names\_to = "variable", values\_to = "count") |>

# filter(count > 0) %>%

# mutate(

# presence =1

# ) %>%

# pivot\_wider(

# names\_from = variable,

# values\_from = presence,

# values\_fn = list # This ensures we create lists of count values for each variable

# ) %>%

# unnest(cols = starts\_with("exp") | starts\_with("sup")) # Unnest any lists of count values to separate rows

single2019 <- raw2019 %>%

filter(origin == "single") %>%

mutate(

count = 1,

numb.clumps = 0

)

regen2019 <- bind\_rows(clump2019, single2019) %>%

mutate(across(where(is.numeric), ~replace\_na(., 0))) %>%

mutate(across(starts\_with("su"), ~replace\_na(., 0))) %>% # Replace NA with 0 for "su" variables, the "supressed" variables

mutate(across(starts\_with("ex"), ~replace\_na(., 0))) %>% # Replace NA with 0 for "ex" variables, the "exposed" variables

full\_join(point\_labels) %>%

filter(harvest %in% c("boot", "gasline", "redpine", "stationrd", "wedge")) %>%

#filter(is.na(spp)) %>%

mutate(

season = if\_else(is.na(spp), "2019", season),

origin = if\_else(is.na(spp), "single", origin),

date = if\_else(is.na(spp), "000", date),

spp = if\_else(is.na(spp), "001", spp)

) %>%

mutate(across(

where(is.numeric), # Select all numeric columns

~ if\_else(is.na(spp), 0, .x) # Set to 0 when spp is NA, otherwise keep original values

))

```

2020 data

```{r}

raw2020 <- read\_excel(file\_path\_2020, sheet = "regen", skip = 0 ) |>

select(! c(top4, Comments)) |>

mutate(

season = as.character(season),

spp = as.character(spp),

sup.seed03 = as.numeric(sup.seed03),

sup.sapl = as.numeric(sup.sapl)

) |>

mutate(origin = case\_when(

origin %in% c("s","u") ~ "single",

origin %in% c("t", "r") ~ "clump",

TRUE ~ origin # Keeps other values unchanged

)) |>

select(harvest, season, point, location, spp, origin, everything())

# in 2020, clump data were located on the same row,

#separate the clumps to create a variable count for calculation of tpa and also to calculate ramet (number sprouts / clump)

clump2020 <- raw2020 |>

filter(origin == "clump") %>%

mutate(

count = 1,

numb.clumps = rowSums(across(c(starts\_with("sup"), starts\_with("exp")), ~ if\_else(. > 0, 1, 0)), na.rm = TRUE)

)

# pivot\_longer(cols = matches("^(su|ex)"), names\_to = "variable", values\_to = "count") |>

# filter(count > 0) %>%

# mutate(

# presence =1

# ) %>%

# pivot\_wider(

# names\_from = variable,

# values\_from = presence,

# values\_fn = list # This ensures we create lists of count values for each variable

# ) %>%

# unnest(cols = starts\_with("exp") | starts\_with("sup")) # Unnest any lists of count values to separate rows

single2020 <- raw2020 %>%

filter(origin == "single") %>%

mutate(

count = 1,

numb.clumps = 0

)

regen2020 <- bind\_rows(clump2020, single2020) %>%

mutate(point = as.character(point)) %>%

mutate(across(where(is.numeric), ~replace\_na(., 0))) %>%

mutate(across(starts\_with("su"), ~replace\_na(., 0))) %>% # Replace NA with 0 for "su" variables

mutate(across(starts\_with("ex"), ~replace\_na(., 0))) # Replace NA with 0 for "ex" variables

```

2021 data

```{r}

raw2021 <- read\_excel(file\_path\_2021, sheet = "regen", skip = 2 ) |>

select(! c(top4)) |>

mutate(

season = as.character(season),

location = as.character(location),

spp = as.character(spp)

) |>

mutate(origin = case\_when(

origin %in% c("s","u") ~ "single",

origin %in% c("v", "r") ~ "clump",

TRUE ~ origin # Keeps other values unchanged

)) |>

mutate(

sup.seed03 = as.numeric(sup.seed03),

sup.sapl = as.numeric(sup.sapl)

) %>%

select(harvest, season, point, location, spp, origin, everything())

#separate the clumps to create a variable count for calculation of tpa and also to calculate ramet (number sprouts / clump)

clump2021 <- raw2021 |>

filter(origin == "clump") |>

mutate(

count = 1,

numb.clumps = rowSums(across(c(starts\_with("sup"), starts\_with("exp")), ~ if\_else(. > 0, 1, 0)), na.rm = TRUE)

)

# pivot\_longer(cols = matches("^(su|ex)"), names\_to = "variable", values\_to = "count") |>

# filter(count > 0) %>%

# mutate(

# presence =1,

# location = as.character(location)

# ) %>%

# pivot\_wider(

# names\_from = variable,

# values\_from = presence,

# values\_fn = list # This ensures we create lists of count values for each variable

# ) %>%

# unnest(cols = starts\_with("exp") | starts\_with("sup")) # Unnest any lists of count values to separate rows

single2021 <- raw2021 %>%

filter(origin == "single") %>%

mutate(

count = 1,

numb.clumps = 0

)

regen2021 <- bind\_rows(clump2021, single2021) %>%

mutate(

location = as.character(location),

point = as.character(point)

) %>%

mutate(across(where(is.numeric), ~replace\_na(., 0))) %>%

mutate(across(starts\_with("su"), ~replace\_na(., 0))) %>% # Replace NA with 0 for "su" variables

mutate(across(starts\_with("ex"), ~replace\_na(., 0))) # Replace NA with 0 for "ex" variables

```

2022 data

```{r}

raw2022 <- read\_excel(file\_path\_2022, sheet = "regen", skip = 0 ) |>

select(! c(top4, comments)) |>

mutate(

season = as.character(season),

location = as.character(location),

spp = as.character(spp),

point = as.character(point)

) |>

mutate(origin = case\_when(

origin %in% c("s","u") ~ "single",

origin %in% c("v", "r") ~ "clump",

TRUE ~ origin # Keeps other values unchanged

)) |>

mutate(

sup.seed02 = as.numeric(sup.seed03),

sup.seed03 = as.numeric(sup.seed03),

sup.sapl = as.numeric(sup.sapl)

) %>%

select(harvest, season, point, location, spp, origin, everything())

#separate the clumps to create a variable count for calculation of tpa and also to calculate ramet (number sprouts / clump)

clump2022 <- raw2022 |>

filter(origin == "clump") |>

mutate(

count = 1,

numb.clumps = rowSums(across(c(starts\_with("sup"), starts\_with("exp")), ~ if\_else(. > 0, 1, 0)), na.rm = TRUE)

)

# pivot\_longer(cols = matches("^(su|ex)"), names\_to = "variable", values\_to = "count") |>

# filter(count > 0) %>%

# mutate(

# presence =1

# ) %>%

# pivot\_wider(

# names\_from = variable,

# values\_from = presence,

# values\_fn = list # This ensures we create lists of count values for each variable

# ) %>%

# unnest(cols = starts\_with("exp") | starts\_with("sup")) # Unnest any lists of count values to separate rows

single2022 <- raw2022 %>%

filter(origin == "single") %>%

mutate(

count = 1,

numb.clumps = 0

)

regen2022 <- bind\_rows(clump2022, single2022) %>%

mutate(

location = as.character(location)

) %>%

mutate(across(where(is.numeric), ~replace\_na(., 0))) %>%

mutate(across(starts\_with("su"), ~replace\_na(., 0))) %>% # Replace NA with 0 for "su" variables

mutate(across(starts\_with("ex"), ~replace\_na(., 0))) # Replace NA with 0 for "ex" variables

```

2023 veg data

```{r}

# values are already on a per acre basis

regen2023 <- read\_excel(file\_path\_2023\_veg, sheet = "Sheet1", skip = 0 ) %>%

rename("harvest" = "stand",

supA = sup.seed01ac,

supB = sup.seed02ac,

supD = sup.seed03ac,

supP = sup.saplac,

expA = exp.seed01ac,

expB = exp.seed02ac,

expD = exp.seed03ac,

expP = exp.saplac

) %>%

mutate(type = "blank") %>%

mutate(type = case\_when(

spp %in% c("ba", "bc", "bih", "bl", "co", "he", "rm", "ro", "sa",

"sb", "sh", "sm", "wa", "wo", "wp", "yb", "yp") ~ "commercial",

spp %in% c("ab", "eh", "pc", "stm", "toh" ) ~ "interfere",

spp %in% c("app", "bta", "haw", "brs", "dws", "la", "ns", "pb", "qa", "rp", "scp", "smc", "svb", "wh", "wi" ) ~ "diversity",

spp %in% c("ah", "cc", "unk" ) ~ "other",

TRUE ~ type, # Keeps the existing value of "type" for all other cases

)) %>%

mutate(

seedA = supA + expA

) %>%

select(!c(supA, expA)) %>%

mutate(harvest = if\_else(harvest =="gasline", "gas\_line", harvest))

regen2023seedl <- regen2023 %>% # extract the data for the 4" to 12" seedlings to merge into the site data permanent file

select(harvest, wall, point,type, seedA) %>%

filter(!harvest == "recknagel\_north") %>%

filter(type == "commercial") %>%

group\_by(harvest, wall, point, type) %>%

summarize(

seedA = sum(seedA)

) %>%

mutate(

season = "2023"

) %>%

select(-type)

```

site/plot data read and manipulate

```{r}

plot2019 <- read.csv(file\_path\_2019\_plot) %>%

mutate(

location = case\_when(

location == "control" ~ "c",

location == "interior" ~ "i",

location == "perimeter" ~ "p",

TRUE ~ harvest # Keeps other values unchanged

)

) %>%

mutate(

seedA = comm\_sdl,

rhizom = fern,

season = as.character(season),

upper.spp = as.character(upper.spp),

maple = as.integer(maple),

point = as.character(point)

) %>%

mutate(across(c(comm\_sdl), ~ replace\_na(., 0))) %>%

rename(

native\_cov = low.cov,

native\_spp = low.spp,

invasive\_spp = invasive.spp,

invasive\_cov = invasive.cov,

mid\_cov = mid.cov,

mid\_spp = mid.spp,

upper\_cov = upper.cov,

upper\_spp = upper.spp,

vine\_cov = vine.cov,

vine\_spp = vine.spp

) %>%

mutate(seedA = seedA \*(43560 / (pi \* 3.7 \* 3.7))) %>%

select(harvest, season, date, location, point, disturb, slash, fern, herb, native\_cov,

native\_spp, invasive\_cov, invasive\_spp, mid\_cov, mid\_spp, upper\_cov, upper\_spp, vine\_cov,

vine\_spp, baf, oakhick, beech, maple, birch, hdwd, conifer, comm\_sdl, seedA, ) # ba\_acre, cut, year, numb\_yrs\_grow,

plot2020 <- read.csv(file\_path\_2020\_plot) %>% # this has data for boot, wedge and sta\_rd separately presented

mutate(across(c(comm\_sdl), ~ replace\_na(., 0))) %>%

mutate(

season = as.character(season),

rhizom = fern,

mid\_spp = as.character(mid\_spp),

upper\_spp = as.character(upper\_spp),

point = as.character(point),

seedA = case\_when(

comm\_sdl == 0 ~ 0, # If category is 0, the random value is 0

comm\_sdl == 1 ~ sample(1:5, size = n(), replace = TRUE), # Random number from 1 to 5 for each row

comm\_sdl == 2 ~ sample(6:15, size = n(), replace = TRUE), # Random number from 6 to 15 for each row

comm\_sdl == 3 ~ sample(16:30, size = n(), replace = TRUE), # Random number from 16 to 30 for each row

comm\_sdl == 4 ~ sample(31:100, size = n(), replace = TRUE) # Random number from 31 to 100 for each row

)

) %>%

mutate(seedA = seedA \*(43560 / (pi \* 3.7 \* 3.7))) %>%

select(harvest, season, date, location, point, disturb, slash, fern, herb, native\_cov,

native\_spp, invasive\_cov, invasive\_spp, mid\_cov, mid\_spp, upper\_cov, upper\_spp, vine\_cov,

vine\_spp, baf, oakhick, beech, maple, birch, hdwd, conifer, comm\_sdl, seedA, ) # ba\_acre, cut, year, numb\_yrs\_grow,

plot2021 <- read\_excel(file\_path\_2021\_plot, sheet = "site-plot", skip = 6) %>%

mutate(across(c(comm\_sdl), ~ replace\_na(., 0))) %>%

mutate(

season = as.character(season),

rhizom = fern,

mid\_spp = as.character(mid\_spp),

upper\_spp = as.character(upper\_spp),

invasive\_spp = as.character(invasive\_spp),

point = as.character(point),

seedA = case\_when(

comm\_sdl == 0 ~ 0, # If category is 0, the random value is 0

comm\_sdl == 1 ~ sample(1:5, size = n(), replace = TRUE), # Random number from 1 to 5 for each row

comm\_sdl == 2 ~ sample(6:15, size = n(), replace = TRUE), # Random number from 6 to 15 for each row

comm\_sdl == 3 ~ sample(16:30, size = n(), replace = TRUE), # Random number from 16 to 30 for each row

comm\_sdl == 4 ~ sample(31:100, size = n(), replace = TRUE) # Random number from 31 to 100 for each row

)

) %>%

mutate(seedA = seedA \*(43560 / (pi \* 3.7 \* 3.7))) %>%

select(harvest, season, date, location, point, disturb, slash, fern, herb, native\_cov,

native\_spp, invasive\_cov, invasive\_spp, mid\_cov, mid\_spp, upper\_cov, upper\_spp, vine\_cov,

vine\_spp, baf, oakhick, beech, maple, birch, hdwd, conifer, comm\_sdl, seedA, ) # ba\_acre, cut, year, numb\_yrs\_grow,

plot2022\_original <- read\_excel(file\_path\_2022, sheet = "site-plot", skip = 6) %>%

mutate(

point = as.character(point)

)

plot2022 <- read\_excel(file\_path\_2022, sheet = "site-traits", skip = 1) %>% #read in the cafri plot traits

mutate(

point = as.character(point)

) %>%

bind\_rows(plot2022\_original) %>%

mutate(across(c(comm\_sdl), ~ replace\_na(., 0))) %>%

mutate(

season = as.character(season),

rhizom = fern,

point = as.character(point),

seedA = case\_when(

comm\_sdl == 0 ~ 0, # If category is 0, the random value is 0

comm\_sdl == 1 ~ sample(1:5, size = n(), replace = TRUE), # Random number from 1 to 5 for each row

comm\_sdl == 2 ~ sample(6:15, size = n(), replace = TRUE), # Random number from 6 to 15 for each row

comm\_sdl == 3 ~ sample(16:30, size = n(), replace = TRUE), # Random number from 16 to 30 for each row

comm\_sdl == 4 ~ sample(31:100, size = n(), replace = TRUE) # Random number from 31 to 100 for each row

)

) %>%

mutate(seedA = seedA \*(43560 / (pi \* 3.7 \* 3.7))) %>%

select(harvest, season, date, location, point, comm\_sdl, seedA ) # ba\_acre, cut, year, numb\_yrs\_grow,

```

2023 plot and site CAFRI and other

```{r}

cafri\_point\_labels <- read\_excel("CAFRI Treatment & Plot Location 8-2024.xlsx", sheet = "point counts", skip = 4) %>%

rename(

harvest = stand,

trmt = status

) %>%

mutate(

wall = as.character(wall),

point = as.character(point),

location = trmt

) %>%

mutate(

location = case\_when(

location == "control" ~ "c",

location == "protected" ~ "p",

TRUE ~ location # Keeps other values unchanged

)

)

plot2023cafri <- read\_excel(file\_path\_2023\_plot, sheet = "cafri\_site", skip = 3) %>%

select(-comments) %>%

rename(harvest = stand) %>%

mutate(

wall = as.character(wall),

season = as.character(season),

invasive\_spp = as.character(invasive\_spp),

mid\_spp = as.character(mid\_spp),

point = as.character(point)

) %>%

full\_join(cafri\_point\_labels, by = c("harvest", "wall", "point"), suffix = c("\_cafri", "\_labels")) %>%

# select(harvest, wall, season, date, point, location, trmt) %>%

arrange(harvest, point)

# mismatches <- plot2023cafri %>% filter(is.na(harvest\_cafri) | is.na(harvest\_labels))

# #after cafri labels were joined in

#

# # Rows in plot2023cafri but not in cafri\_point\_labels

# extra\_in\_plot <- plot2023cafri %>% anti\_join(cafri\_point\_labels, by = c("harvest", "wall", "point"))

#

# # Rows in cafri\_point\_labels but not in plot2023cafri

# extra\_in\_labels <- cafri\_point\_labels %>% anti\_join(plot2023cafri, by = c("harvest", "wall", "point"))

#

# # Print the results

# print("Rows in plot2023cafri but not in cafri\_point\_labels:")

# print(extra\_in\_plot)

#

# print("Rows in cafri\_point\_labels but not in plot2023cafri:")

# print(extra\_in\_labels)

#

duplicates\_in\_plot <- plot2023cafri %>% group\_by\_all() %>% filter(n() > 1)

print("Duplicates in plot2023cafri:")

print(duplicates\_in\_plot)

#

#

extra\_in\_cafriplot <- plot2023cafri %>% anti\_join(cafri\_point\_labels, by = c("harvest", "wall", "point"))

extra\_in\_labels <- cafri\_point\_labels %>% anti\_join(plot2023cafri, by = c("harvest", "wall", "point"))

duplicates\_in\_cafri <- plot2023cafri %>% group\_by\_all() %>% filter(n() > 1)

duplicates\_in\_labels <- cafri\_point\_labels %>% group\_by\_all() %>% filter(n() > 1)

plot2023other <- read\_excel(file\_path\_2023\_plot, sheet = "other\_site\_baf10", skip = 2) %>%

select(-comments) %>%

rename(harvest = stand) %>%

mutate(

spp = as.character(spp),

wall = as.character(wall),

season = as.character(season),

invasive\_spp = as.character(invasive\_spp),

mid\_spp = as.character(mid\_spp),

point = as.character(point)

) %>%

mutate(wall = if\_else(is.na(wall), "99", wall) )

plot2023\_overstory <- plot2023other %>%

select(harvest, wall, location, season, date, point, spp, dbh, in16ft, status,crown, logs)

output\_file <- file.path(output\_folder, paste0("campridge\_overstory\_by\_point\_2023data(13feb2025)", ".xlsx"))

write\_xlsx(plot2023\_overstory, output\_file)

plot2023\_site <- plot2023other %>%

select(harvest, wall, location, season, date, point, rhizom, clump, grass,

herb, herb\_spp, native\_cov, native\_spp, mid\_cov, mid\_spp, invasive\_cov, invasive\_spp, vine\_cov, vine\_spp) %>%

distinct(point, .keep\_all = TRUE)

plot2023 <- bind\_rows(plot2023cafri, plot2023\_site) %>%

mutate(

point = as.character(point)

) %>%

left\_join(regen2023seedl, by = c("harvest", "season", "wall", "point"))

#regen2023seedl has the seedling 4" - 12" data as "seedA" from the vegetation file.

#

```

notes 2/5/2025 for the plotxxx files

1. DONE standardize variable names for plot2019 thru plot2022. go to original excel files

2. bind\_rows for plotxxxx df, write a master file of site/plot variables to \*.xlsx in the summary folder

3. standardize harvest names to match df=all\_merge

4. select(harvest, point, seedA, location, wall)

5. bind\_rows of plot df with all\_merge df. Change NA to zero for seedling variables

6. make complete

7. bind\_rows with regen2023

To standardize labels the strata will be denoted as:

A = 4" to 12" (these are not sorted as suppressed vs. exposed and spp is "004" = other hardwood)

B = 12" to 4.5/5 ft

C1 = 4.5 ft to 9 ft

C2 = 9ft to 1" dbh

D = 4.5/5 ft to 1"

P = 1" to 4" (saplings)

Merge Site/Plot Data

```{r}

site\_data <- bind\_rows(plot2019, plot2020, plot2021, plot2022, plot2023) %>%

mutate(across(c(oakhick, beech, maple, birch, hdwd, conifer), ~ replace\_na(., 0))) %>%

mutate(across(c(baf), ~ replace\_na(., 10))) %>%

mutate(across(c(wall), ~ replace\_na(., "99"))) %>%

mutate(

basalarea = baf \* (oakhick + beech + maple + birch + hdwd + conifer)

) %>%

mutate(

harvest = case\_when(

harvest == "b" ~ "sta\_rd",

harvest == "boot" ~ "sta\_rd",

harvest == "cgl" ~ "gas\_line",

harvest == "circle1" ~ "circle",

harvest == "circle2" ~ "circle",

harvest == "circle3" ~ "circle",

harvest == "cricle1" ~ "circle",

harvest == "crp" ~ "red\_pine",

harvest == "cwb" ~ "sta\_rd",

harvest == "gasline" ~ "gas\_line",

harvest == "gl" ~ "gas\_line",

harvest == "recksouth" ~ "recknagel",

harvest == "rect1" ~ "recknagel",

harvest == "rect2" ~ "recknagel",

harvest == "rect3" ~ "recknagel",

harvest == "rect4" ~ "recknagel",

harvest == "rectangle1" ~ "recknagel",

harvest == "rectangle2" ~ "recknagel",

harvest == "rectangle3" ~ "recknagel",

harvest == "rectangle4" ~ "recknagel",

harvest == "rp" ~ "red\_pine",

harvest == "redpine" ~ "red\_pine",

harvest == "stationrd" ~ "sta\_rd",

harvest == "w" ~ "sta\_rd",

harvest == "wb" ~ "sta\_rd",

harvest == "wedge" ~ "sta\_rd",

harvest == "old" ~ "new",

TRUE ~ harvest # Keeps other values unchanged

)

) %>%

select(- c(location, trmt)) %>%

left\_join(point\_labels, by = c("harvest", "point")) %>%

left\_join(cafri\_point\_labels, by = c("harvest", "wall", "point")) %>%

mutate(

trmt = coalesce(trmt.x, trmt.y), # Use non-NA value from trmt.x or trmt.y

location = coalesce(location.x, location.y), # Same for location.x and location.y

height = coalesce(height, height.y)

) %>%

select(-trmt.x, -trmt.y, -location.x, -location.y, -height.y) %>% # Drop the old columns

select(harvest, wall, season, date, point, location, trmt, height, disturb, slash, fern, rhizom, clump,

grass, herb, herb\_spp, native\_cov, native\_spp, mid\_cov, mid\_spp, vine\_cov, vine\_spp, invasive\_cov,

invasive\_spp, upper\_cov, upper\_spp, basalarea, baf, oakhick, beech, maple, birch, hdwd, conifer,

comm\_sdl, seedA) %>%

arrange(harvest, season, wall, point) %>%

# left\_join(regen2023seedl, by = c("harvest", "wall", "point")) %>%

mutate(

# seedA = coalesce(seedA.x, seedA.y),

rhiz\_fern = coalesce(fern, rhizom)

) %>%

select( -fern, -rhizom)

# filter(harvest == "gas\_line")

#write the aggregation of all site data to the output file

##change the name of the output file to today's date

output\_file <- file.path(output\_folder, paste0("site\_plot\_data\_2019\_2023(13feb2025)", ".xlsx"))

write\_xlsx(site\_data, output\_file)

```

#Final, bind rows()

# change harvest names()

# when origin == single, change count to 1

# make sure all variable names and type match

# make sure plot count = 248 for each season (join with location\_by\_point.xlsx), check harvest names

# this will also populate "location" for years GE 2021

# convert to per acre values

# calculate clumps/acre and ramets

To standardize labels:

A = 4" to 12" radius = 3.7 ft; count in 2019, 2023, and 2024 otherwise categorical 0 = 0; 1 = 1to5; 2 = 6 to 15; 3 = 16 to 30; 4 = 31+

B = 12" to 4.5/5 ft radius = 6 ft

C1 = 4.5 ft to 9 ft radius = 6 ft

C2 = 9ft to 1" dbh radius = 6 ft

D = 4.5/5 ft to 1" radius = 6 ft

P = 1" to 4" (saplings) radius thru 2022 = 6 ft; then 16 ft

merging veg data across years, select 4" to 12" seedlings from the site/plot data

df=merge2019\_2022\_a

```{r}

seedlingspre2023 <- site\_data %>% #from counts of seedlings onto site tally sheets

filter (!season == "2023"

) %>%

select(harvest, wall, point, season, seedA) %>%

mutate(

spp ="004"

)

seedlings2019\_2023 <- regen2023seedl %>% #this is the count of seedlings 4" to 12" from the 2023 veg data

bind\_rows(seedlingspre2023) %>% #seedlings before 2023

mutate(

spp = "004",

origin = "single"

)

#convert regen data to "per acre" for 2019 - 2022 before merging with 2023.

merge2019\_2022\_a <- bind\_rows(regen2019, regen2020, regen2021, regen2022) %>%

select(harvest, season, location, point, origin, spp, date, count, numb.clumps, sup.seed01, sup.seed02, sup.seed03,

sup.sapl, exp.seed01, exp.seed02, exp.seed03, exp.sapl) %>%

#next, standardize the seedling labels for variance across years

rename(

supB = sup.seed01,

supC1 = sup.seed02,

supC2 = sup.seed03,

supP = sup.sapl,

expB = exp.seed01,

expC1 = exp.seed02,

expC2 = exp.seed03,

expP = exp.sapl

) %>%

#next, standardize harvest names

mutate(

harvest = case\_when(

harvest == "b" ~ "sta\_rd",

harvest == "boot" ~ "sta\_rd",

harvest == "cgl" ~ "gas\_line",

harvest == "circle1" ~ "circle",

harvest == "circle2" ~ "circle",

harvest == "circle3" ~ "circle",

harvest == "cricle1" ~ "circle",

harvest == "crp" ~ "red\_pine",

harvest == "cwb" ~ "sta\_rd",

harvest == "gasline" ~ "gas\_line",

harvest == "gl" ~ "gas\_line",

harvest == "recksouth" ~ "recknagel",

harvest == "rect1" ~ "recknagel",

harvest == "rect2" ~ "recknagel",

harvest == "rect3" ~ "recknagel",

harvest == "rect4" ~ "recknagel",

harvest == "rectangle1" ~ "recknagel",

harvest == "rectangle2" ~ "recknagel",

harvest == "rectangle3" ~ "recknagel",

harvest == "rectangle4" ~ "recknagel",

harvest == "rp" ~ "red\_pine",

harvest == "redpine" ~ "red\_pine",

harvest == "stationrd" ~ "sta\_rd",

harvest == "w" ~ "sta\_rd",

harvest == "wb" ~ "sta\_rd",

harvest == "wedge" ~ "sta\_rd",

harvest == "old" ~ "new",

TRUE ~ harvest # Keeps other values unchanged

)

)

```

Merge veg data, assign attributes, average stems.ramet, and convert to per acre values at each point

```{r}

#Next, assign attribute labels to each point

merge2019\_2022\_b <- merge2019\_2022\_a %>%

select(- c(location)) %>%

left\_join(point\_labels, by = c("harvest", "point")) %>%

left\_join(cafri\_point\_labels, by = c("harvest", "point")) %>%

mutate(

trmt = coalesce(trmt.x, trmt.y), # Use non-NA value from trmt.x or trmt.y

location = coalesce(location.x, location.y), # Same for location.x and location.y

height = coalesce(height.x, height.y),

wall = if\_else(is.na(wall), "99", wall), # if variable "wall" is.NA, then make it "99"

) %>%

select(-trmt.x, -trmt.y, -location.x, -location.y, -height.x, -height.y) %>% # Drop the old columns

arrange(harvest, season, wall, point, origin, spp) %>%

#Next, calculate the average # stems per ramet

mutate(

stems.ramet = if\_else(origin == "clump", #TODO need to double check this calculation #DONE, calculation confirmed 2/13/2025

rowMeans(across(starts\_with("sup") | starts\_with("exp"),

~ if\_else(. > 0, ., NA\_real\_)), na.rm = TRUE), #within a row, assess if >0, then calculates mean of clump size

NA\_real\_)

) %>%

#Next, convert plot counts to "per acre" values

mutate( #converts plot count to count per acre

count = if\_else(is.na(count), 1, count), # if count missing (it's a single) then make count = 1

stems.ramet = if\_else(is.na(stems.ramet), 0, stems.ramet), # if variable is NA, change to 0, otherwise leave as is

radius01 = 3.7,

radius02 = 6,

radius03 = 16,

radius = if\_else(season == "2023", radius03, radius02), # Dynamically assign the correct radius (no 2023 data included as of 2/13/2025)

supB = count \* (supB \* (43560 / (pi \* radius02^2))),

supC1 = count \* (supC1 \* (43560 / (pi \* radius02^2))),

supC2 = count \* (supC2 \* (43560 / (pi \* radius02^2))),

supP = count \* (supP \* (43560 / (pi \* radius^2))), # Uses the new "radius" variable here

expB = count \* (expB \* (43560 / (pi \* radius02^2))),

expC1 = count \* (expC1 \* (43560 / (pi \* radius02^2))),

expC2 = count \* (expC2 \* (43560 / (pi \* radius02^2))),

expP = count \* (expP \* (43560 / (pi \* radius^2))) # Uses the new "radius" variable here

) %>%

ungroup() %>%

mutate( #if there is concern there are blank spaces in the grouping variable that cause grouping problems

season = str\_trim(season),

harvest = str\_trim(harvest),

point = str\_trim(point),

spp = str\_trim(spp),

origin = str\_trim(origin)

)

```

#Next, aggregate numeric values by harvest, season, point, spp, and origin. Calucate number of clumps and stems per ramet

# most data are on a per acre basis within the plot

#still need to calcuate number of clumps per acre at the point level

```{r}

merge2019\_2022\_c <- merge2019\_2022\_b %>%

arrange(harvest, season, wall, point, spp, origin) %>%

group\_by(season, harvest, wall, point, spp, origin) %>%

summarize(

wall = first(wall),

trmt = first(trmt),

location = first(location),

height = first(height),

n = n(), #should be the number of occurrences of a species x origin combo.

#For clumps, this is the number of clumps per point, with variable radii

clumps.acre = sum(numb.clumps \* (43560 / (pi \* radius02^2))),

supB = sum(supB),

supC1 = sum(supC1),

supC2 = sum(supC2),

supP = sum(supP),

expB = sum(expB),

expC1 = sum(expC1),

expC2 = sum(expC2),

expP = sum(expP),

stems.ramet = if\_else(sum(stems.ramet != 0, na.rm = TRUE) > 0, mean(stems.ramet[stems.ramet != 0], na.rm = TRUE), NA\_real\_),

#the above code checks the group to make sure the sum is >0, and if so then calculates the mean

richness = n\_distinct(spp),

.groups = "drop"

) %>%

select(harvest, season, wall, trmt, point, spp, origin, clumps.acre, stems.ramet, everything()) %>%

arrange(harvest, season, wall, point, spp, origin) %>%

mutate(

stemcount = supB + supC1 + supC2 + supP + expB + expC1 + expC2 + expP,

upperexposed = expC1 + expC2 + expP,

present = if\_else(stemcount > 0, 1, stemcount), # used to calculate diversity if a spp is present at a point

success = if\_else(upperexposed > 0, 1, upperexposed), # used to calculate diversity of spp of successful stems

stems.ramet = if\_else(is.na(stems.ramet), 0, stems.ramet)

) %>%

# change spp to alpha

mutate(

spp = as.character(fct\_recode(

factor(spp),

"rp" = "125", "wp" = "129", "scp" = "130", "he" = "261",

"svb" = "356", "sm" = "310", "stm" = "315", "rm" = "316",

"sm" = "318", "ah" = "341", "ab" = "351", "svb" = "355",

"yb" = "371", "sb" = "372", "pb" = "375", "ah" = "391",

"bih" = "402", "sh" = "407", "pc" = "461", "unk" = "492",

"haw" = "500", "ab" = "521", "ab" = "53", "ab" = "531",

"wa" = "541", "wh" = "585", "yp" = "621", "ab" = "631",

"app" = "660", "la" = "70", "eh" = "701", "unk" = "706",

"bta" = "743", "qa" = "746", "pc" = "761", "bc" = "762",

"cc" = "763", "wo" = "802", "co" = "832", "ro" = "833",

"smc" = "866", "ns" = "91", "bl" = "901", "wi" = "920",

"sa" = "931", "ba" = "951", "toh" = "998", "unk" = "999",

"dws" = "Devil's Walking Stick",

#data recording errors or previously missed

"sb" = "273", "gb" = "379", "au" = "535", "ec" = "742",

"sp" = "90", "ohw" = "001", "ohw" = "1", "ab" = "331",

"unk" = "ush"

))

)

```

#create a stand alone reference to the numeric variables that will be in the "complete" datafile

```{r}

numeric\_vars <- c("clumps.acre", "stems.ramet", "height", "n", "supB", "supC1", "supC2",

"supP", "expB", "expC1", "expC2", "expP", "richness", "stemcount",

"upperexposed", "present", "success")

```

attempt #1 - make the data complete (failed)

```{r eval=FALSE}

veg\_2019\_2022\_complete <- merge2019\_2022\_c %>%

ungroup() %>%

select(harvest, season, wall, point, spp, numeric\_vars, -location, -origin, -trmt, -type) %>%

arrange(harvest, season, wall, point, spp) %>%

group\_by(harvest, season, wall, point) %>%

# complete(spp, fill = list(!!!setNames(rep(0, length(numeric\_vars)), numeric\_vars))) %>%

complete(spp, fill = list(!!!setNames(rep(NA, length(numeric\_vars)), numeric\_vars))) %>%

ungroup() %>%

arrange(harvest, season, wall, spp, origin, point)

```

attempt #2 - make the data complete (failed)

```{r eval=FALSE}

# Step 1: Create a full list of unique species (spp) per wall

species\_per\_wall <- merge2019\_2022\_c %>%

group\_by(wall) %>%

summarize(tableall\_species = list(unique(spp)), .groups = "drop")

# Step 2: Use `complete()` to expand every point within a wall to include all species

veg\_2019\_2022\_complete <- merge2019\_2022\_c %>%

left\_join(species\_per\_wall, by = "wall") %>% # Add all\_species for each wall

unnest(all\_species) %>% # Expand species for each wall

rename(spp\_all = all\_species) %>% # Avoid overwriting original spp

group\_by(harvest, season, wall, point) %>% # Group by the desired levels

complete(

spp = spp\_all, # Ensure all species are present

fill = list(

!!!setNames(rep(0, length(numeric\_vars)), numeric\_vars), # Fill numeric columns with 0

origin = "unknown" # Fill origin with "unknown"

)

) %>%

ungroup() %>%

arrange(harvest, season, wall, spp, origin, point)

```

#attempt #3 - make the data complete (this worked!)

```{r}

# Step 1: Create a full list of unique species (spp) per wall

species\_per\_wall <- merge2019\_2022\_c %>%

group\_by(wall) %>%

summarize(all\_species = list(unique(spp)), .groups = "drop")

# Step 2: Create all combinations of `harvest`, `season`, `wall`, `point`, and `all\_species`

all\_combinations <- merge2019\_2022\_c %>%

select(harvest, season, wall, point, -trmt, -location) %>%

distinct() %>%

left\_join(species\_per\_wall, by = "wall") %>% # Attach the full species list per wall

unnest(all\_species) %>% # Expand the species list

rename(spp = all\_species) # Rename to match the original dataset

# Step 3: Complete the dataset by joining the full combinations with the original data

veg\_2019\_2022\_complete <- all\_combinations %>%

left\_join(merge2019\_2022\_c, by = c("harvest", "season", "wall", "point", "spp")) %>%

select(harvest, season, wall, point, spp, origin, all\_of(numeric\_vars)) %>%

mutate(

# Fill missing numeric variables with 0

across(all\_of(numeric\_vars), ~ replace\_na(.x, 0)),

# Fill `origin` with "unknown" if missing

origin = replace\_na(origin, "unknown")

) %>%

arrange(harvest, season, wall, spp, point, origin)

```

```{r}

veg01 <- veg\_2019\_2022\_complete %>%

left\_join(seedlings2019\_2023, by = c("harvest", "season", "wall", "point", "spp", "origin")) %>% #season = 2023 excluded

select(-height) %>%

left\_join(cafri\_point\_labels, by = c("harvest", "wall", "point")) %>%

left\_join(point\_labels, by = c("harvest", "point")) %>%

arrange(harvest, season, wall, spp, point, origin) %>%

mutate(

seedA = coalesce(seedA.x, seedA.y),

height = coalesce(height.x, height.y),

trmt = coalesce(trmt.x, trmt.y),

location = coalesce(location.x, location.y),

supD = supC1 + supC2,

expD = expC1 + expC2

) %>%

select(

-seedA.x, -seedA.y, -height.x, -height.y, -trmt.x, -trmt.y, -location.x, -location.y

) %>%

select(harvest, season, wall, trmt, point, location, height, spp, origin, clumps.acre, stems.ramet, everything()) %>%

select(-richness, -stemcount, -upperexposed, -n) %>%

# Assign "type" category lables to the species

mutate(

type = "blank",

height = as.character(height)

) %>%

mutate(type = case\_when(

spp %in% c("ba", "bc", "bih", "bl", "co", "he", "ohw", "rm", "ro", "sa",

"sb", "sh", "sm", "wa", "wo", "wp", "yb", "yp") ~ "commercial",

spp %in% c("ab", "eh", "pc", "stm", "toh", "au" ) ~ "interfere",

spp %in% c("app", "bta", "haw", "brs", "dws", "ec", "la", "ns", "pb", "gb", "qa", "rp", "scp", "smc", "svb", "wh", "wi" ) ~ "diversity",

spp %in% c("ah", "cc", "unk", "sp" ) ~ "other",

TRUE ~ type, # Keeps the existing value of "type" for all other cases

))

# left\_join df=site\_data for var=seedA by season, harvest, point. Make spp=ohw and origin = single

# make this df complete

# align variables of this df with df=regen2023, then bind rows

```

```{r}

regen2023\_updated <- regen2023 %>%

mutate(

season = "2023"

) %>%

rename(

trmt = status, #new = old

stems.ramet = avg\_ramet\_size,

clumps.acre = clump\_ac

) %>%

select(-clump\_ac\_small, -clump\_ac\_med, -clump\_ac\_large, -clump\_ac\_sapl, -overstory, -stemcount, -exposed) %>%

left\_join(seedlings2019\_2023, by = c("harvest", "season", "wall", "point", "spp", "origin")) %>%

mutate(

seedA = coalesce(seedA.x, seedA.y)

) %>%

select(

-seedA.x, -seedA.y

)

```

```{r}

yay <- bind\_rows(veg01, regen2023\_updated) %>%

mutate(

firstgrow = 2035, # default value for the first growing season

firstgrow = case\_when(

harvest %in% c("gas\_line", "red\_pine") ~ 2017,

harvest %in% c("sta\_rd") ~ 2018,

harvest %in% c("campridge", "recknagel", "circle", "decker") ~ 2021,

harvest %in% c("6\_6", "6-9") ~ 2022,

TRUE ~ firstgrow # Keeps the existing value of "firstgrow" for all other cases

),

year = as.numeric(season),

growingseasons = (year - firstgrow) + 1

) %>%

ungroup() %>%

group\_by(harvest, season, wall, trmt) %>%

summarize(

season = first(season),

firstgrow = first(firstgrow),

year = first(year),

growingseasons = first(growingseasons),

richness = n\_distinct(spp[present > 0], na.rm = TRUE),

group\_size = n(),

.groups = "drop"

)```