

Clean data

Tim White

June 2024

Packages and data

```
library(readxl)
library(tidyverse)

orig_all_data <- read_excel("data/data_raw.xlsx", sheet = "All_data_Categories")
orig_plots_data <- read_excel("data/data_raw.xlsx", sheet = "Plots_data")
orig_seedlings_data <- read_excel("data/data_raw.xlsx", sheet = "Seedlings_cut_sprouting_together")
```

Stem-level data

```
data_stems <- orig_all_data %>%
  # Remove unnecessary columns
  # Note: Forest'sAge is removed because it is the same as VegetationType
  # Note: SoilType is removed per Florencia's request
  select(plot_id = `Plot #`,
         dbh = `DBH (cm)`,
         ba_per_stem = `BasalArea per tree m2 ha-1``,
         harvested = Harvested,
         vegetation_type = VegetationType,
         milpa = `Milpa(has it been milpa)``,
         artisan = Artesanos) %>%
  # Add column for size class
  mutate(size_class = case_when(
    dbh <= 4 ~ "sapling",
    dbh <= 9 ~ "tree_05to09",
    dbh <= 14 ~ "tree_10to14",
    dbh <= 19 ~ "tree_15to19",
    dbh >= 20 ~ "tree_20plus"
  )) %>%
  # Convert variables to factors
  mutate(plot_id = as.factor(plot_id),
         harvested = as.factor(harvested),
         vegetation_type = as.factor(vegetation_type),
         milpa = as.factor(milpa),
         size_class = as.factor(size_class),
         artisan = as.factor(artisan)) %>%
```

```

# Clean up factor levels
mutate(harvested = fct_recode(harvested, "yes" = "Yes", "no" = "No"),
       vegetation_type = fct_recode(vegetation_type,
                                      "juuche" = "Ju'uche'",
                                      "keelenche" = "Keelenche'",
                                      "nukuuchche" = "Nuku'uch che'")) %>%
# Change the 167 "Maybe" cases for milpa to "No"
mutate(milpa = fctCollapse(milpa, yes = c("Yes"), no = c("Maybe", "No"))) %>%
# Reorder columns
select(plot_id, dbh, ba_per_stem, size_class, everything())

# Write to csv
write_csv(data_stems, "data/data_stems.csv")

```

Plot-level data

```

data_plots <- orig_plots_data %>%
  # Remove unnecessary columns
  # Note: Forest'sAge is removed because it is the same as VegetationType
  # Note: SoilType is removed per Florencia's request
  select(plot_id = `Plot #ID`,
         ba_totaltrees = `Basal Area (m2/ha-1)`,
         harvested = Harvested,
         vegetation_type = VegetationType,
         milpa = `Milpa(has it been milpa)` ,
         artisan = Artesanos,
         N, W) %>%
  # Convert variables to factors
  mutate(plot_id = as.factor(plot_id),
         harvested = as.factor(harvested),
         vegetation_type = as.factor(vegetation_type),
         milpa = as.factor(milpa),
         artisan = as.factor(artisan)) %>%
  # Convert coordinates to decimals
  extract(N, into = c("Nh", "Nm", "Ns"), regex = "(\\d{2})'(\\d{2})'\\'((\\d{2}(?:\\.\\d+)?"))
  mutate(Nh = as.numeric(Nh), Nm = as.numeric(Nm), Ns = as.numeric(Ns)) %>%
  mutate(latitude = Nh + Nm/60 + Ns/3600) %>%
  extract(W, into = c("Wh", "Wm", "Ws"), regex = "(\\d{2})'\\'\\s*(\\d{2})'\\'\\'((\\d{2}(?:\\.\\d+)?"))
  mutate(Wh = as.numeric(Wh), Wm = as.numeric(Wm), Ws = as.numeric(Ws)) %>%
  mutate(longitude = -(Wh + Wm/60 + Ws/3600)) %>%
  select(-Nh, -Nm, -Ns, -Wh, -Wm, -Ws) %>%
  # Clean up factor levels
  mutate(harvested = fct_recode(harvested, "yes" = "Yes", "no" = "No"),
         vegetation_type = fct_recode(vegetation_type,
                                      "juuche" = "Ju'uche'",
                                      "keelenche" = "Keelenche'",
                                      "nukuuchche" = "Nuku'uch che'")) %>%
# Change the 167 "Maybe" cases for milpa to "No"
mutate(milpa = fctCollapse(milpa, yes = c("Yes"), no = c("Maybe", "No")))

# Create variables to store basal area (m2/ha) and stem density (stems per ha) for each size class

```

```

ba_by_size_class <- data_stems %>%
  group_by(plot_id, size_class) %>%
  summarize(ba = sum(ba_per_stem), .groups = "drop") %>%
  pivot_wider(names_from = size_class,
              values_from = ba, values_fill = 0) %>%
  mutate(ba_seedlings = 0) %>%
  select(plot_id, ba_seedlings,
         ba_saplings = sapling,
         ba_trees05to09 = tree_05to09,
         ba_trees10to14 = tree_10to14,
         ba_trees15to19 = tree_15to19,
         ba_trees20plus = tree_20plus)

stemden_by_size_class <- data_stems %>%
  count(plot_id, size_class) %>%
  # For all size classes except seedlings (which is already per ha),
  # we multiply by 10000/(100*pi) to convert from (# of stems per
  # plot) to (# of stems per ha)
  mutate(n = (10000/(100*pi))*n) %>%
  pivot_wider(names_from = size_class,
              values_from = n, values_fill = 0) %>%
  mutate(stemden_seedlings = orig_seedlings_data$`Seedlings per Ha`,
         stemden_totaltrees = tree_05to09 + tree_10to14 +
          tree_15to19 + tree_20plus) %>%
  select(plot_id, stemden_seedlings, stemden_totaltrees,
         stemden_saplings = sapling,
         stemden_trees05to09 = tree_05to09,
         stemden_trees10to14 = tree_10to14,
         stemden_trees15to19 = tree_15to19,
         stemden_trees20plus = tree_20plus)

data_plots <- data_plots %>%
  left_join(ba_by_size_class, by = "plot_id") %>%
  left_join(stemden_by_size_class, by = "plot_id") %>%
  # Total BA for each plot should not include sapling BA
  mutate(ba_totaltrees = ba_totaltrees - ba_saplings) %>%
  select(plot_id, harvested, vegetation_type, milpa,
         artisan, latitude, longitude,
         starts_with("ba"), starts_with("stemden"))

# Check to make sure Milpa is consistent between data_stems and data_plots
data_stems %>%
  group_by(plot_id) %>%
  summarize(stems_milpa = unique(milpa)) %>%
  left_join(data_plots %>% select(plot_id, plots_milpa = milpa), by = "plot_id") %>%
  summarize(all(stems_milpa == plots_milpa)) %>% pull()

## [1] TRUE

# Check to make sure that for each plot, total BA = sum of BA of all trees
data_plots %>% summarize(all(abs(ba_totaltrees - ba_trees05to09 - ba_trees10to14 -
  ba_trees15to19 - ba_trees20plus) < 1e-6)) %>% pull()

```

```
## [1] TRUE

# Check to make sure that for each plot, # total trees = sum of all trees
data_plots %>% summarize(all(abs(stemden_totaltrees - stemden_trees05to09 - stemden_trees10to14 -
                                     stemden_trees15to19 - stemden_trees20plus) < 1e-6)) %>% pull()

## [1] TRUE

# Write to csv
write_csv(data_plots, "data/data_plots.csv")
```