

# Assessing Influence of Land Cover Type on Soil Nutrient Status in Southeastern U.S.

Daniel Whitehead, Elizabeth Weyl, and P. Louis Lu

2025-03-13

## Load packages

```
library(here)
```

```
## here() starts at /Users/danielwhitehead/ENV710/projectset
```

```
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
```

```
library(dunn.test)
library(gt)
library(paletteer)
library(moments)
library(dplyr)
library(vcdExtra)
```

```
## Loading required package: vcd
## Loading required package: grid
## Loading required package: gnm
##
## Attaching package: 'vcdExtra'
##
## The following object is masked from 'package:dplyr':
##
##     summarise
```

```
library(ggplot2)
library(readr)
library(dplyr)
library(readxl)
library(leaflet)
library(readr)
library(htmlwidgets)
library(webshot2)
library(ggspatial)
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(viridis)
```

```
## Loading required package: viridisLite
##
## Attaching package: 'viridis'
##
## The following object is masked from 'package:maps':
##
##     unemp
```

```
library(broom)
library(tidyr)
library(e1071)
```

```
##
## Attaching package: 'e1071'
##
## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```
library(prism)
```

```
## Be sure to set the download folder using 'prism_set_dl_dir()'.
```

```
library(terra)
```

```
## terra 1.8.29
##
## Attaching package: 'terra'
##
## The following object is masked from 'package:e1071':
##
##     interpolate
##
## The following objects are masked from 'package:vcd':
##
##     mosaic, sieve
##
## The following object is masked from 'package:grid':
##
##     depth
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(sp)
library(knitr)
```

```
##
## Attaching package: 'knitr'
##
## The following object is masked from 'package:terra':
##
##     spin
```

## Wrangling

```
A_horizon <- read_excel("Data/A_horizon.xls", skip=12)
A_horizon <- A_horizon[-1, ]
View(A_horizon)

# Reference for Units
A_horizon_reference <- read_excel("Data/A_horizon.xls", skip=11)
View(A_horizon_reference)
A_horizon_units <- head(A_horizon_reference, 2)
View(A_horizon_units)
```

## Initial Data

```
A_states <- A_horizon %>%
  filter(StateID %in% c("NC", "SC", "VA", "GA", "FL"))
```

```
# View the filtered data
head(A_states)
```

```
## # A tibble: 6 x 82
##   A_LabID SiteID StateID Latitude Longitude CollDate LandCover1
##   <chr>    <dbl> <chr>   <chr>      <chr>    <dtm>      <chr>
## 1 C-328641    32 FL      30.4575    -85.9548~ 2009-02-16 00:00:00 Forested ~
## 2 C-328546    48 FL      25.435300000~ -80.6948~ 2009-02-05 00:00:00 Herbaceou~
## 3 C-326183   416 FL      30.130500000~ -83.3389~ 2009-02-14 00:00:00 Planted/C~
## 4 C-328693   480 FL      28.550799999~ -82.3023~ 2009-02-11 00:00:00 Planted/C~
## 5 C-328705   736 FL      27.251100000~ -80.6932~ 2009-02-06 00:00:00 Planted/C~
## 6 C-328703   928 FL      29.268599999~ -81.7343~ 2009-02-12 00:00:00 Forested ~
## # i 75 more variables: LandCover2 <chr>, A_Depth <chr>, A_Quartz <chr>,
## #   A_Tot_K_fs <chr>, A_Tot_Plug <chr>, A_Tot_Flds <chr>, A_Tot_14A <chr>,
## #   A_Tot_10A <chr>, A_Kaolinit <chr>, A_Tot_Clay <chr>, A_Gibbsite <chr>,
## #   A_Calcite <chr>, A_Dolomite <chr>, A_Aragon <chr>, A_Tot_Carb <chr>,
## #   A_Analcime <chr>, A_Heuland <chr>, A_Tot_Zeol <chr>, A_Gypsum <chr>,
## #   A_Talc <chr>, A_Hornbl <chr>, A_Serpent <chr>, A_Hematite <chr>,
## #   A_Goethite <chr>, A_Pyroxene <chr>, A_Pyrite <chr>, A_Other <chr>, ...
```

Figure 1: Land Cover Frequency by State

```
landcover_counts <- A_states %>%
  group_by(StateID, LandCover1) %>%
  summarize(count = n(), .groups = "drop")

# Create the bar plot
landcover_count <- ggplot(landcover_counts, aes(x = StateID, y = count, fill = LandCover1)) +
  geom_bar(stat = "identity", position = "dodge") + # 'dodge' makes the bars for each state side by side
  labs(
    title = "Land Cover Frequency by State",
    x = "State",
    y = "Number of Sites",
    fill = "Land Cover Type"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotates the x-axis labels for better readability
landcover_count

ggsave(landcover_count,
  filename = "landcover_count.jpg",
  height = 12,
  width = 15,
  units = "cm")
```

Figure 2: Total Nutrient Abundance Across States

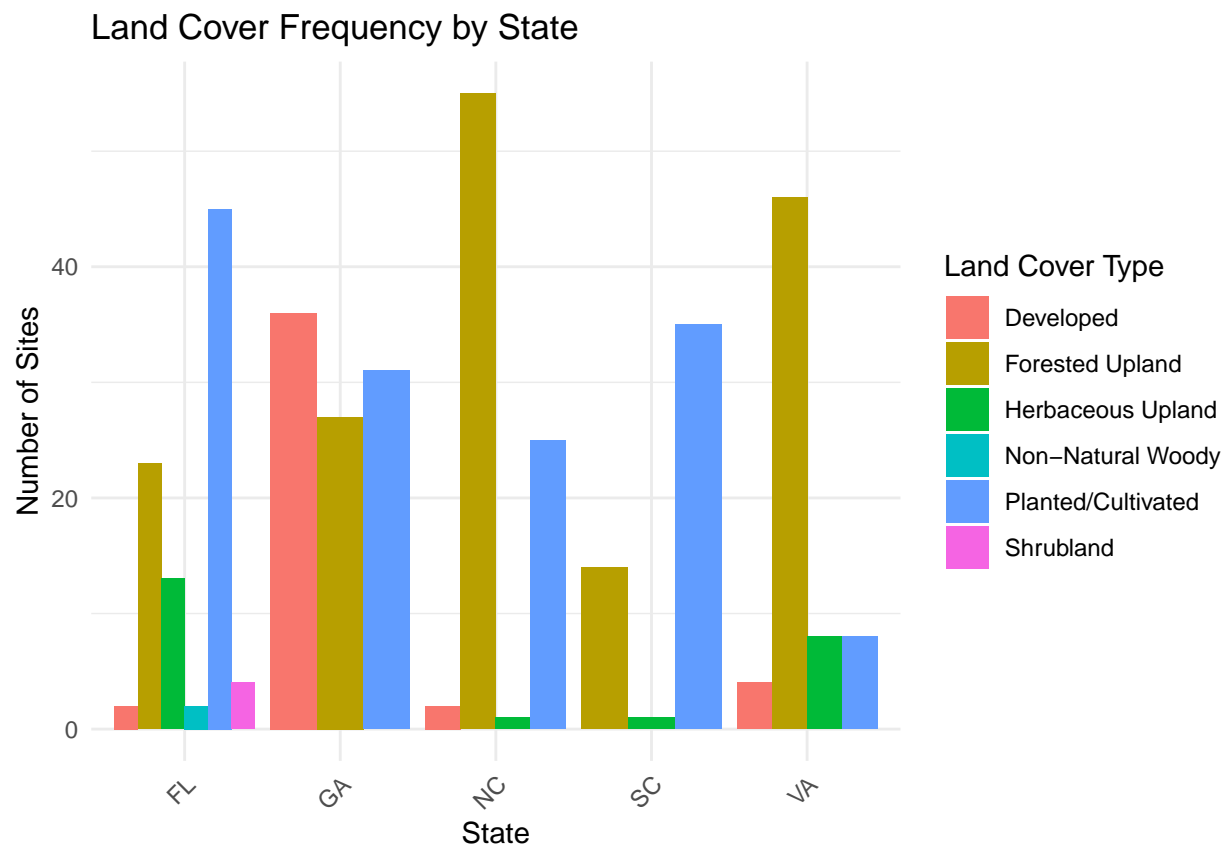


Figure 1: Land status sample type in Florida, Georgia, North Carolina, South Carolina, and Virginia. This data is from the USGS Geochemical and mineralogical data for soils of the conterminous United States database

```

# Convert relevant columns to numeric (direct conversion)
A_states$A_C_Tot <- as.numeric(A_states$A_C_Tot)

## Warning: NAs introduced by coercion

A_states$A_Ca <- as.numeric(A_states$A_Ca)

## Warning: NAs introduced by coercion

A_states$A_Mg <- as.numeric(A_states$A_Mg)

## Warning: NAs introduced by coercion

#A_states$A_Tot_K_fs <- as.numeric(A_states$A_Tot_K_fs) # Use A_Tot_K_fs instead of A_Kaolinit
A_states$A_K <- as.numeric(A_states$A_K) # No, we should use A_K instead of A_Tot_K_fs

## Warning: NAs introduced by coercion

A_states$A_P <- as.numeric(A_states$A_P)

## Warning: NAs introduced by coercion

A_states$A_S <- as.numeric(A_states$A_S)

## Warning: NAs introduced by coercion

A_states <- A_states %>%
  mutate(A_P = A_P/10000)

# Now reshape the data into long format for plotting
A_states_long <- A_states %>%
  pivot_longer(cols = c(A_C_Tot, A_Ca, A_Mg, A_K, A_P, A_S),
               names_to = "Nutrient", values_to = "Abundance")

# Sum the abundances by nutrient
A_states_sum <- A_states_long %>%
  group_by(StateID, Nutrient) %>%
  summarize(Total_Abundance = sum(Abundance, na.rm = TRUE))

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

```

## Original bar plots

```

#narrowing the dataset to just macro nutrients
A_macros <- A_states %>%
  select(SiteID, StateID, CollDate, Longitude, Latitude, LandCover1, LandCover2, A_Depth, A_K, A_Ca, A_S,
         rename(K = A_K,
                Ca = A_Ca,
                S = A_S,
                C = A_C_Tot,
                Mg = A_Mg,
                P = A_P
               )

A_macros_long <- A_macros %>%
  pivot_longer(cols = c(C, Ca, Mg, K, P, S),
               names_to = "Nutrient", values_to = "Abundance")

A_macros_sum <- A_macros_long %>%
  group_by(StateID, LandCover1, Nutrient) %>%
  summarize(Total_Abundance = sum(Abundance, na.rm = TRUE))

```

## 'summarise()' has grouped output by 'StateID', 'LandCover1'. You can override  
## using the '.groups' argument.

```

#without P
A_macros_noP <- A_states %>%
  select(SiteID, StateID, CollDate, LandCover1, LandCover2, A_Depth, A_K, A_Ca, A_S, A_C_Tot, A_Mg) %>%
  rename(K = A_K,
         Ca = A_Ca,
         S = A_S,
         C = A_C_Tot,
         Mg = A_Mg)

A_macros_long_noP <- A_macros_noP %>%
  pivot_longer(cols = c(C, Ca, Mg, K, S),
               names_to = "Nutrient", values_to = "Abundance")

A_macros_sum_noP <- A_macros_long_noP %>%
  group_by(StateID, LandCover1, Nutrient) %>%
  summarize(Total_Abundance = sum(Abundance, na.rm = TRUE))

```

## 'summarise()' has grouped output by 'StateID', 'LandCover1'. You can override  
## using the '.groups' argument.

## Macronutrients boxplots by land cover type

```

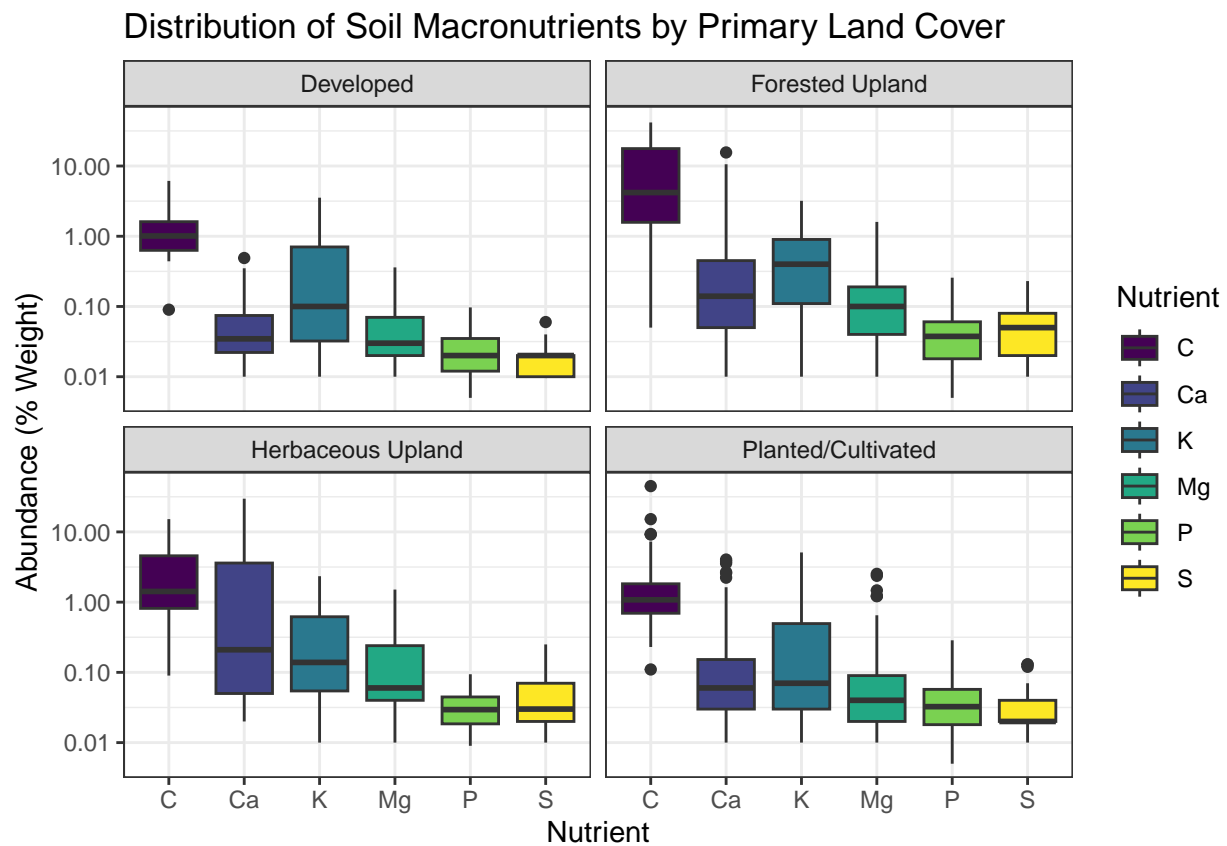
A_macros_long <- A_macros_long %>%
  filter(LandCover1 != "Non-Natural Woody" &
         LandCover1 != "Shrubland")

macronutrient_box <- ggplot(A_macros_long, aes(x = Nutrient, y = Abundance, fill = Nutrient))+

```

```
geom_boxplot() +
#geom_bar(stat = "identity", .groups = "drop")+
facet_wrap(vars(LandCover1), nrow = 2)+
theme_bw()+
labs(x = "Nutrient",
     y = "Abundance (% Weight)",
     title = "Distribution of Soil Macronutrients by Primary Land Cover") +
scale_fill_viridis(discrete = TRUE) +
scale_y_log10()
macronutrient_box
```

```
## Warning: Removed 284 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```



```
annotations <- data.frame(
  Nutrient = rep(c("C", "Ca", "K", "Mg", "P", "S"), each = 4), # 3 annotations per facet
  x = c(1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4,
        1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4), # x positions for each annotation in the respective facet
  y = c(80, 80, 80, 80, 80, 80, 80, 80,
        8, 8, 8, 8, 5, 5, 5, 5,
        0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), # y positions for each annotation
  label = c("a", "b", "a", "a",
            "a", "b", "b", "c",
            "a", "b", "a", "a"),
```



```

      "a", "b", "b", "a",
      "a", "b", "b", "b",
      "a", "b", "b", "c")) # Text for each annotation

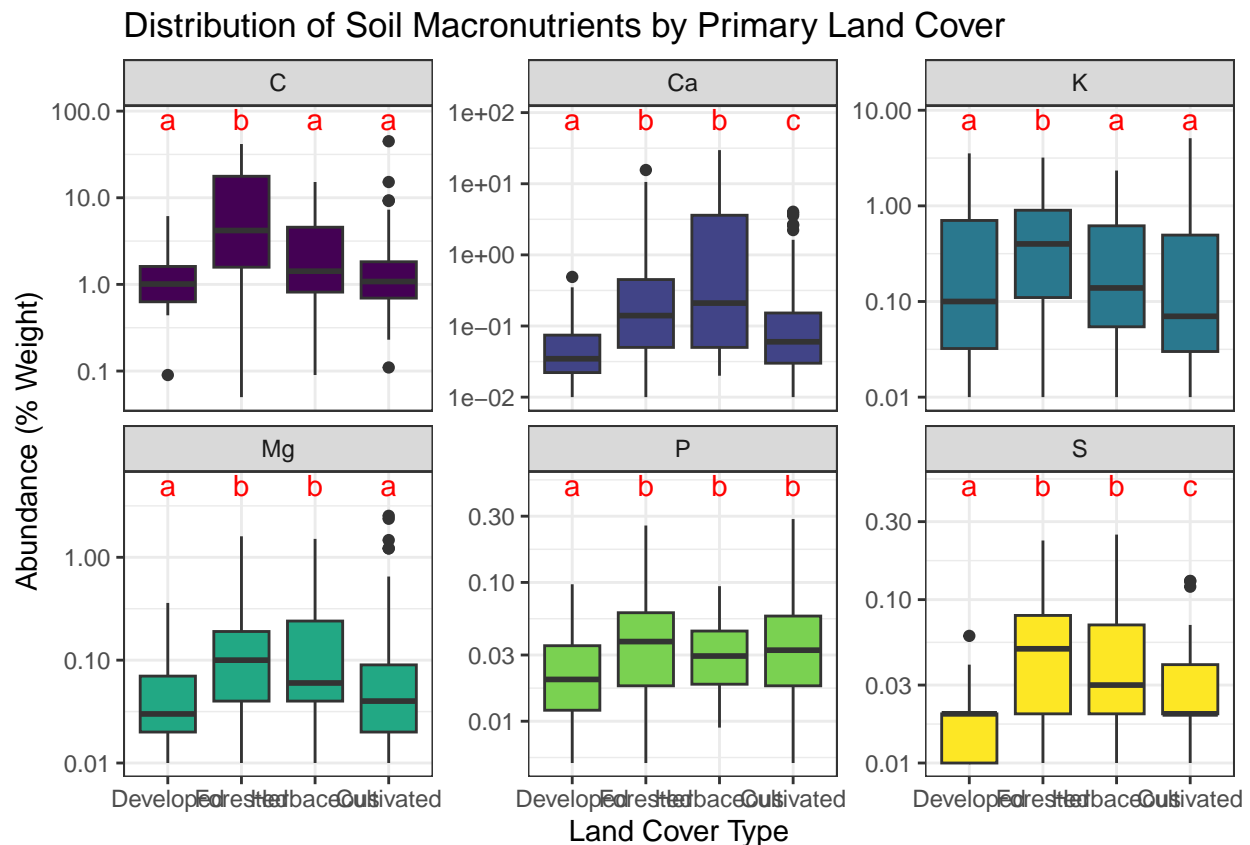
# Boxplots by Nutrients
macronutrient_box1 <- ggplot(A_macros_long, aes(x = LandCover1, y = Abundance, fill = Nutrient))+
  geom_boxplot() +
  #geom_bar(stat = "identity", .groups = "drop")+
  facet_wrap(vars(Nutrient), nrow = 2, scales = "free_y")+
  theme_bw()+
  labs(x = "Land Cover Type",
       y = "Abundance (% Weight)",
       title = "Distribution of Soil Macronutrients by Primary Land Cover") +
  scale_fill_viridis(discrete = TRUE) +
  scale_y_log10() +
  theme(legend.position = "none") +
  scale_x_discrete(labels = c("Developed" = "Developed",
                             "Forested Upland" = "Forested",
                             "Herbaceous Upland" = "Herbaceous",
                             "Planted/Cultivated" = "Cultivated")) +
  geom_text(data = annotations, aes(x = x, y = y, label = label), inherit.aes = FALSE, size = 4, color = "red")
macronutrient_box1

```

```

## Warning: Removed 284 rows containing non-finite outside the scale range
## ('stat_boxplot()').

```



```
ggsave("macronutrient_boxplot_annotated.png", plot = macronutrient_box1, width = 10, height = 6, units = "in")
```

```
## Warning: Removed 284 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```

```
ggsave("macronutrient_boxplot.png", plot = macronutrient_box, width = 10, height = 6, units = "in")
```

```
## Warning: Removed 284 rows containing non-finite outside the scale range
## ('stat_boxplot()').
```

## Figure 4: Site map

```
# Make long & lat numeric
A_states$Longitude <- as.numeric(A_states$Longitude)
A_states$Latitude <- as.numeric(A_states$Latitude)
# Convert to sf object
points_sf <- st_as_sf(A_states, coords = c("Longitude", "Latitude"), crs = 4326)

# Get background map data
usa_map <- map_data("usa")
state_map <- map_data("state")
world_map <- map_data("world")

# Create a static map using ggplot2
map <- ggplot() +
  # Add world base map
  geom_polygon(data = world_map, aes(x = long, y = lat, group = group),
    fill = "white", color = "gray70", size = 0.2) +
  geom_polygon(data = state_map, aes(x = long, y = lat, group = group),
    fill = NA, color = "gray70", size = 0.3) +
  # Add points with category color and value size
  geom_point(data = A_states, aes(x = Longitude, y = Latitude, color = LandCover1),
    size = 1.2, shape = 16) +
  # Styling
  scale_color_manual(values = c("darkorange", "green", "indianred2", "blue", "pink", "purple"),
    name = "Land Cover Types") +
  #scale_size_continuous(name = "Value", range = c(3, 6)) +
  # Focus map on our data
  coord_sf(xlim = c(min(A_states$Longitude) - 2, max(A_states$Longitude) + 2),
    ylim = c(min(A_states$Latitude) - 1, max(A_states$Latitude) + 1)) +
  # Add map elements
  annotation_north_arrow(location = "tl", style = north_arrow_fancy_orienteering) +
  # Theme and labels
  theme_minimal() +
  labs(title = "Soil Sampling Sites by Land Cover Type",
    caption = "Map showing soil sampling sites (n = 382) in the southeastern U.S. by land cover type",
    x = "Longitude", y = "Latitude") +
  theme(
    panel.background = element_rect(fill = "white", color = "white"), # White background
    plot.background = element_rect(fill = "white", color = "white"), # White plot background
```

```

#panel.grid.major = element_blank(), # Optional: remove gridlines
#panel.grid.minor = element_blank() # Optional: remove minor gridlines
)

```

```

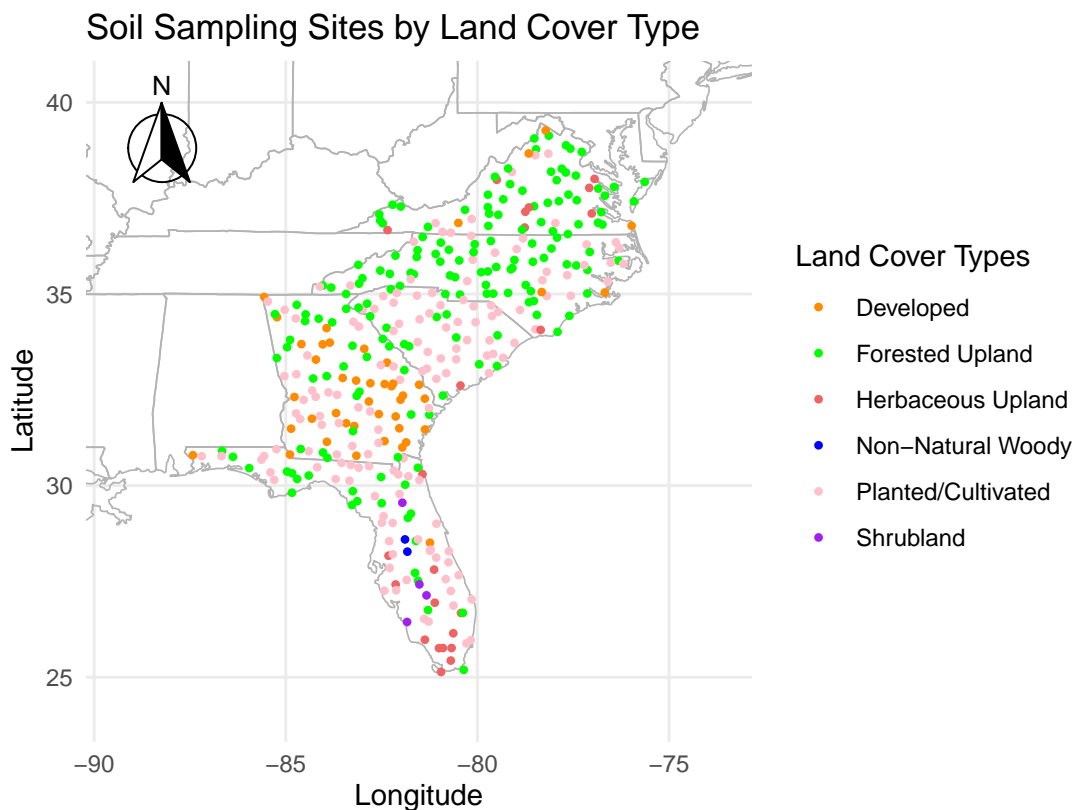
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

print(map)

```



ving soil sampling sites (n = 382) in the southeastern U.S. by land cover types

```

ggsave("site_map_uniform.png", map, dpi = 300)

```

```

## Saving 6.5 x 4.5 in image

```

## Confidence Interval

```

# filter the dataset by four land cover types
forest_macro <- A_macros %>%
  filter(LandCover1 == "Forested Upland")

```

```

developed_macro <- A_macros %>%
  filter(LandCover1 == "Developed")

herbaceous_macro <- A_macros %>%
  filter(LandCover1 == "Herbaceous Upland")

cultivated_macro <- A_macros %>%
  filter(LandCover1 == "Planted/Cultivated")

A_macros_4 <- A_macros %>%
  filter(LandCover1 != "Non-Natural Woody" &
    LandCover1 != "Shrubland")

```

Calculate 95% CI for each macronutrient by land cover

```

# Carbon
forest_C_test <- t.test(forest_macro$C, conf.level = 0.95)

forest_C_test

```

```

##
## One Sample t-test
##
## data: forest_macro$C
## t = 11.649, df = 164, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 8.872564 12.494345
## sample estimates:
## mean of x
## 10.68345

```

```

developed_C_test <- t.test(developed_macro$C, conf.level = 0.95)

developed_C_test

```

```

##
## One Sample t-test
##
## data: developed_macro$C
## t = 7.4373, df = 43, p-value = 3.012e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 1.082990 1.888829
## sample estimates:
## mean of x
## 1.485909

```

```
herbaceous_C_test <- t.test(herbaceous_macro$C, conf.level = 0.95)
```

```
herbaceous_C_test
```

```
##
## One Sample t-test
##
## data: herbaceous_macro$C
## t = 3.9799, df = 22, p-value = 0.0006334
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.876100 5.958683
## sample estimates:
## mean of x
##  3.917391
```

```
cultivated_C_test <- t.test(cultivated_macro$C, conf.level = 0.95)
```

```
cultivated_C_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$C
## t = 5.9124, df = 142, p-value = 2.397e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.360443 2.727109
## sample estimates:
## mean of x
##  2.043776
```

```
C_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_C = mean(C, na.rm = TRUE),
            sd = sd(C, na.rm = TRUE),
            CI95_lower = t.test(C, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(C, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()
```

```
# View
C_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_C      sd CI95_lower CI95_upper
##   <chr>          <dbl> <dbl>    <dbl>    <dbl>
## 1 Developed         1.49  1.33      1.08      1.89
## 2 Forested Upland    10.7  11.8      8.87     12.5
## 3 Herbaceous Upland   3.92  4.72      1.88      5.96
## 4 Planted/Cultivated  2.04  4.13      1.36      2.73
```

```
# Ca
forest_Ca_test <- t.test(forest_macro$Ca, conf.level = 0.95)

forest_Ca_test
```

```
##
## One Sample t-test
##
## data: forest_macro$Ca
## t = 3.7098, df = 144, p-value = 0.0002955
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.2263178 0.7425098
## sample estimates:
## mean of x
## 0.4844138
```

```
developed_Ca_test <- t.test(developed_macro$Ca, conf.level = 0.95)

developed_Ca_test
```

```
##
## One Sample t-test
##
## data: developed_macro$Ca
## t = 4.5067, df = 37, p-value = 6.399e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.03852861 0.10147139
## sample estimates:
## mean of x
## 0.07
```

```
herbaceous_Ca_test <- t.test(herbaceous_macro$Ca, conf.level = 0.95)

herbaceous_Ca_test
```

```
##
## One Sample t-test
##
## data: herbaceous_macro$Ca
## t = 2.6961, df = 22, p-value = 0.01319
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 1.369108 10.495240
## sample estimates:
## mean of x
## 5.932174
```

```
cultivated_Ca_test <- t.test(cultivated_macro$Ca, conf.level = 0.95)

cultivated_Ca_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$Ca
## t = 4.3366, df = 127, p-value = 2.92e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.1432711 0.3837601
## sample estimates:
## mean of x
## 0.2635156
```

```
Ca_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_S = mean(Ca, na.rm = TRUE),
            sd = sd(Ca, na.rm = TRUE),
            CI95_lower = t.test(Ca, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(Ca, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()

# View
Ca_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_S      sd CI95_lower CI95_upper
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 Developed      0.07  0.0957     0.0385     0.101
## 2 Forested Upland 0.484  1.57      0.226     0.743
## 3 Herbaceous Upland 5.93 10.6      1.37      10.5
## 4 Planted/Cultivated 0.264 0.687     0.143     0.384
```

```
# K
forest_K_test <- t.test(forest_macro$K, conf.level = 0.95)

forest_K_test
```

```
##
## One Sample t-test
##
## data: forest_macro$K
## t = 11.934, df = 156, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.5119553 0.7150511
## sample estimates:
## mean of x
## 0.6135032
```

```
developed_K_test <- t.test(developed_macro$K, conf.level = 0.95)

developed_K_test
```

```
##
## One Sample t-test
##
## data: developed_macro$K
## t = 4.0954, df = 41, p-value = 0.0001933
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.2745580 0.8087753
## sample estimates:
## mean of x
## 0.5416667
```

```
herbaceous_K_test <- t.test(herbaceous_macro$K, conf.level = 0.95)
```

```
herbaceous_K_test
```

```
##
## One Sample t-test
##
## data: herbaceous_macro$K
## t = 3.2782, df = 21, p-value = 0.003587
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.1700186 0.7599814
## sample estimates:
## mean of x
## 0.465
```

```
cultivated_K_test <- t.test(cultivated_macro$K, conf.level = 0.95)
```

```
cultivated_K_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$K
## t = 6.1402, df = 129, p-value = 9.455e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.2727795 0.5321436
## sample estimates:
## mean of x
## 0.4024615
```

```
K_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_K = mean(K, na.rm = TRUE),
            sd = sd(K, na.rm = TRUE),
            CI95_lower = t.test(K, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(K, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()
```



```
# View
K_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_K    sd CI95_lower CI95_upper
##   <chr>          <dbl> <dbl>      <dbl>      <dbl>
## 1 Developed      0.542 0.857      0.275      0.809
## 2 Forested Upland 0.614 0.644      0.512      0.715
## 3 Herbaceous Upland 0.465 0.665      0.170      0.760
## 4 Planted/Cultivated 0.402 0.747      0.273      0.532
```

```
# Mg
forest_Mg_test <- t.test(forest_macro$Mg, conf.level = 0.95)

forest_Mg_test
```

```
##
## One Sample t-test
##
## data: forest_macro$Mg
## t = 9.6118, df = 150, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.1341061 0.2035098
## sample estimates:
## mean of x
## 0.1688079
```

```
developed_Mg_test <- t.test(developed_macro$Mg, conf.level = 0.95)

developed_Mg_test
```

```
##
## One Sample t-test
##
## data: developed_macro$Mg
## t = 4.7334, df = 41, p-value = 2.633e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.04436567 0.11039623
## sample estimates:
## mean of x
## 0.07738095
```

```
herbaceous_Mg_test <- t.test(herbaceous_macro$Mg, conf.level = 0.95)

herbaceous_Mg_test
```

```
##
## One Sample t-test
##
```

```
## data: herbaceous_macro$Mg
## t = 2.8055, df = 20, p-value = 0.01092
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.05703248 0.38772942
## sample estimates:
## mean of x
## 0.222381
```

```
cultivated_Mg_test <- t.test(cultivated_macro$Mg, conf.level = 0.95)

cultivated_Mg_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$Mg
## t = 4.6379, df = 125, p-value = 8.735e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.08758373 0.21797183
## sample estimates:
## mean of x
## 0.1527778
```

```
Mg_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_Mg = mean(Mg, na.rm = TRUE),
            sd = sd(Mg, na.rm = TRUE),
            CI95_lower = t.test(Mg, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(Mg, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()

# View
Mg_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_Mg    sd CI95_lower CI95_upper
##   <chr>          <dbl> <dbl>      <dbl>      <dbl>
## 1 Developed      0.0774 0.106      0.0444      0.110
## 2 Forested Upland 0.169 0.216      0.134      0.204
## 3 Herbaceous Upland 0.222 0.363      0.0570      0.388
## 4 Planted/Cultivated 0.153 0.370      0.0876      0.218
```

```
# P
forest_P_test <- t.test(forest_macro$P, conf.level = 0.95)

forest_P_test
```

```
##
## One Sample t-test
##
```

```
## data: forest_macro$P
## t = 14.889, df = 151, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.04067757 0.05312506
## sample estimates:
## mean of x
## 0.04690132
```

```
developed_P_test <- t.test(developed_macro$P, conf.level = 0.95)

developed_P_test
```

```
##
## One Sample t-test
##
## data: developed_macro$P
## t = 7.7425, df = 40, p-value = 1.772e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.02038459 0.03478615
## sample estimates:
## mean of x
## 0.02758537
```

```
herbaceous_P_test <- t.test(herbaceous_macro$P, conf.level = 0.95)

herbaceous_P_test
```

```
##
## One Sample t-test
##
## data: herbaceous_macro$P
## t = 6.6617, df = 19, p-value = 2.267e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.02417489 0.04632511
## sample estimates:
## mean of x
## 0.03525
```

```
cultivated_P_test <- t.test(cultivated_macro$P, conf.level = 0.95)

cultivated_P_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$P
## t = 12.003, df = 131, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

```
## 0.03815895 0.05321984
## sample estimates:
## mean of x
## 0.04568939
```

```
P_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_P = mean(P, na.rm = TRUE),
            sd = sd(P, na.rm = TRUE),
            CI95_lower = t.test(P, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(P, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()

# View
P_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_P      sd CI95_lower CI95_upper
##   <chr>          <dbl>   <dbl>     <dbl>     <dbl>
## 1 Developed      0.0276 0.0228     0.0204     0.0348
## 2 Forested Upland 0.0469 0.0388     0.0407     0.0531
## 3 Herbaceous Upland 0.0353 0.0237     0.0242     0.0463
## 4 Planted/Cultivated 0.0457 0.0437     0.0382     0.0532
```

```
# S
forest_S_test <- t.test(forest_macro$S, conf.level = 0.95)

forest_S_test
```

```
##
## One Sample t-test
##
## data: forest_macro$S
## t = 15.285, df = 125, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.04774043 0.06194211
## sample estimates:
## mean of x
## 0.05484127
```

```
developed_S_test <- t.test(developed_macro$S, conf.level = 0.95)

developed_S_test
```

```
##
## One Sample t-test
##
## data: developed_macro$S
## t = 6.9832, df = 20, p-value = 8.901e-07
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

```
## 0.01469370 0.02721106
## sample estimates:
## mean of x
## 0.02095238
```

```
herbaceous_S_test <- t.test(herbaceous_macro$S, conf.level = 0.95)
```

```
herbaceous_S_test
```

```
##
## One Sample t-test
##
## data: herbaceous_macro$S
## t = 3.5743, df = 13, p-value = 0.003395
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.02486522 0.10084906
## sample estimates:
## mean of x
## 0.06285714
```

```
cultivated_S_test <- t.test(cultivated_macro$S, conf.level = 0.95)
```

```
cultivated_S_test
```

```
##
## One Sample t-test
##
## data: cultivated_macro$S
## t = 9.9599, df = 65, p-value = 1.045e-14
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.02543810 0.03819826
## sample estimates:
## mean of x
## 0.03181818
```

```
S_summary <- A_macros_4 %>%
  group_by(LandCover1) %>%
  summarize(mean_S = mean(S, na.rm = TRUE),
            sd = sd(S, na.rm = TRUE),
            CI95_lower = t.test(S, conf.level = 0.95)$conf.int[1],
            CI95_upper = t.test(S, conf.level = 0.95)$conf.int[2]) %>%
  ungroup()
```

```
# View
S_summary
```

```
## # A tibble: 4 x 5
##   LandCover1      mean_S      sd CI95_lower CI95_upper
##   <chr>          <dbl> <dbl>    <dbl>    <dbl>
## 1 Developed      0.0210 0.0137    0.0147    0.0272
```

```
## 2 Forested Upland      0.0548 0.0403      0.0477      0.0619
## 3 Herbaceous Upland   0.0629 0.0658      0.0249      0.101
## 4 Planted/Cultivated 0.0318 0.0260      0.0254      0.0382
```

## Assessing Normality

### Shapiro Wilk Test

```
# Shapiro-Wilk test for Carbon (C)
shapiro_C <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "C"])
shapiro_C_tidy <- tidy(shapiro_C)

# Shapiro-Wilk test for Calcium (Ca)
shapiro_Ca <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "Ca"])
shapiro_Ca_tidy <- tidy(shapiro_Ca)

# Shapiro-Wilk test for Magnesium (Mg)
shapiro_Mg <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "Mg"])
shapiro_Mg_tidy <- tidy(shapiro_Mg)

# Shapiro-Wilk test for Potassium (K)
shapiro_K <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "K"])
shapiro_K_tidy <- tidy(shapiro_K)

# Shapiro-Wilk test for Phosphorus (P)
shapiro_P <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "P"])
shapiro_P_tidy <- tidy(shapiro_P)

# Shapiro-Wilk test for Sulfur (S)
shapiro_S <- shapiro.test(A_macros_long$Abundance[A_macros_long$Nutrient == "S"])
shapiro_S_tidy <- tidy(shapiro_S)

normality_results <- bind_rows(
  shapiro_C_tidy %>% mutate(Nutrient = "C"),
  shapiro_Ca_tidy %>% mutate(Nutrient = "Ca"),
  shapiro_Mg_tidy %>% mutate(Nutrient = "Mg"),
  shapiro_K_tidy %>% mutate(Nutrient = "K"),
  shapiro_P_tidy %>% mutate(Nutrient = "P"),
  shapiro_S_tidy %>% mutate(Nutrient = "S")
)

print(normality_results)
```

```
## # A tibble: 6 x 4
##   statistic p.value method      Nutrient
##   <dbl>     <dbl> <chr>      <chr>
## 1    0.625 8.53e-28 Shapiro-Wilk normality test C
## 2    0.200 2.49e-35 Shapiro-Wilk normality test Ca
## 3    0.495 5.94e-30 Shapiro-Wilk normality test Mg
## 4    0.717 6.01e-24 Shapiro-Wilk normality test K
## 5    0.762 5.52e-22 Shapiro-Wilk normality test P
```

```
## 6      0.809 5.60e-16 Shapiro-Wilk normality test S
```

## Skewness and kurtosis

```
abundance_C <- A_macros_long$Abundance[A_macros_long$Nutrient == "C"] %>% na.omit()
if(length(unique(abundance_C)) > 1) {
  skew_C <- skewness(abundance_C)
  kurt_C <- kurtosis(abundance_C)
} else {
  skew_C <- kurt_C <- NA
}

# Skewness and Kurtosis for Calcium (Ca)
abundance_Ca <- A_macros_long$Abundance[A_macros_long$Nutrient == "Ca"] %>% na.omit()
if(length(unique(abundance_Ca)) > 1) {
  skew_Ca <- skewness(abundance_Ca)
  kurt_Ca <- kurtosis(abundance_Ca)
} else {
  skew_Ca <- kurt_Ca <- NA
}

# Skewness and Kurtosis for Magnesium (Mg)
abundance_Mg <- A_macros_long$Abundance[A_macros_long$Nutrient == "Mg"] %>% na.omit()
if(length(unique(abundance_Mg)) > 1) {
  skew_Mg <- skewness(abundance_Mg)
  kurt_Mg <- kurtosis(abundance_Mg)
} else {
  skew_Mg <- kurt_Mg <- NA
}

# Skewness and Kurtosis for Potassium (K)
abundance_K <- A_macros_long$Abundance[A_macros_long$Nutrient == "K"] %>% na.omit()
if(length(unique(abundance_K)) > 1) {
  skew_K <- skewness(abundance_K)
  kurt_K <- kurtosis(abundance_K)
} else {
  skew_K <- kurt_K <- NA
}

# Skewness and Kurtosis for Phosphorus (P)
abundance_P <- A_macros_long$Abundance[A_macros_long$Nutrient == "P"] %>% na.omit()
if(length(unique(abundance_P)) > 1) {
  skew_P <- skewness(abundance_P)
  kurt_P <- kurtosis(abundance_P)
} else {
  skew_P <- kurt_P <- NA
}

# Skewness and Kurtosis for Sulfur (S)
abundance_S <- A_macros_long$Abundance[A_macros_long$Nutrient == "S"] %>% na.omit()
if(length(unique(abundance_S)) > 1) {
  skew_S <- skewness(abundance_S)
```

```

kurt_S <- kurtosis(abundance_S)
} else {
  skew_S <- kurt_S <- NA
}

skew_kurt_results <- data.frame(
  Nutrient = c("C", "Ca", "Mg", "K", "P", "S"),
  Skewness = c(skew_C, skew_Ca, skew_Mg, skew_K, skew_P, skew_S),
  Kurtosis = c(kurt_C, kurt_Ca, kurt_Mg, kurt_K, kurt_P, kurt_S)
)

```

## Normality Histograms

```

create_histogram <- function(nutrient_name) {
  # Filter data for the selected nutrient
  nutrient_data <- A_macros_long %>%
    filter(Nutrient == nutrient_name) %>%
    na.omit()

  # creating the histogram plot
  hist_plot <- ggplot(nutrient_data, aes(x = Abundance, fill = LandCover1)) +
    geom_histogram(binwidth = 0.1, color = "black", position = "dodge") + # Adjust binwidth as needed
    labs(title = paste("Distribution of", nutrient_name, "Abundance by Primary Land Cover"),
         x = paste(nutrient_name, "Abundance by % weight"),
         y = "Frequency",
         caption = paste("Figure: Distribution of", nutrient_name, "Abundance by Primary Land Cover")) +
    scale_fill_manual(values = c("LandCover1_Value1" = "skyblue", "LandCover1_Value2" = "salmon")) + #
    theme_minimal() +
    theme(legend.position = "right") +
    facet_grid(. ~ LandCover1) # Faceting by LandCover1 for each plot

  return(hist_plot)
}

plot_C <- create_histogram("C")
plot_Ca <- create_histogram("Ca")
plot_Mg <- create_histogram("Mg")
plot_K <- create_histogram("K")
plot_P <- create_histogram("P")
plot_S <- create_histogram("S")

# Print each plot
print(plot_C)

```

```

## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.

```



## Distribution of C Abundance by Primary Land Cover

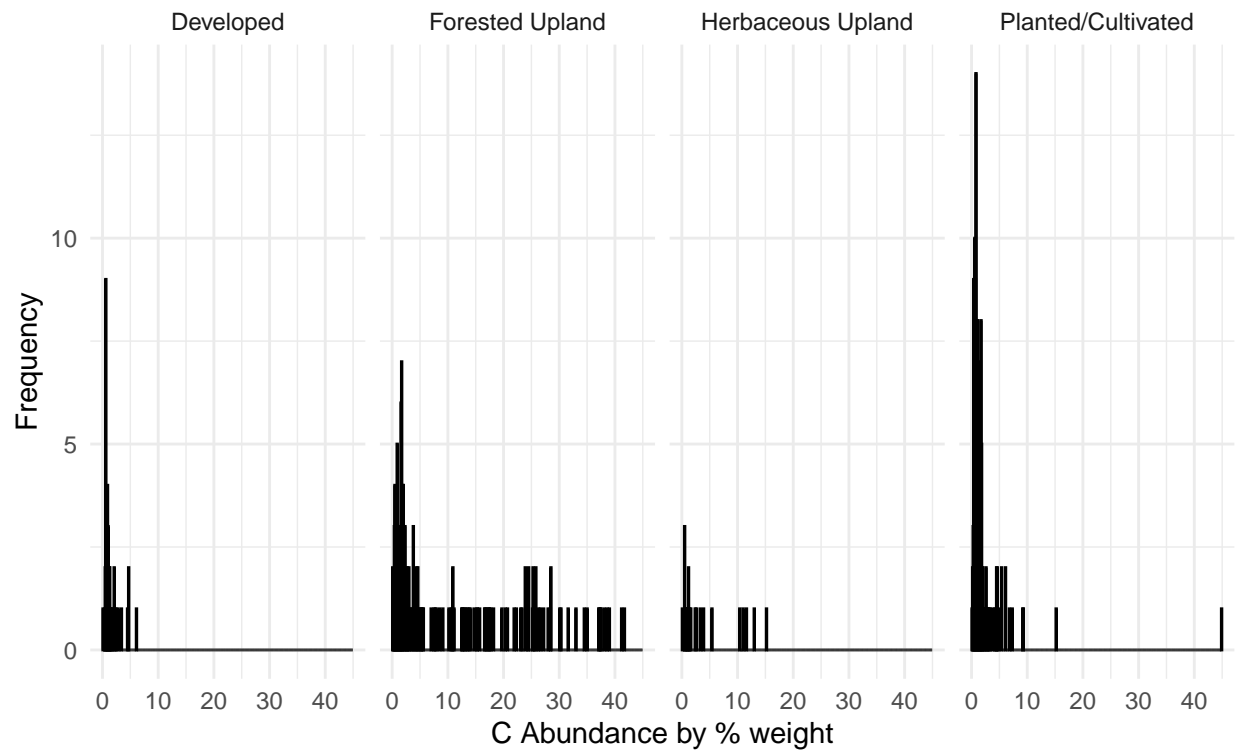


Figure: Distribution of C Abundance by Primary Land Cover

```
print(plot_Ca)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Distribution of Ca Abundance by Primary Land Cover

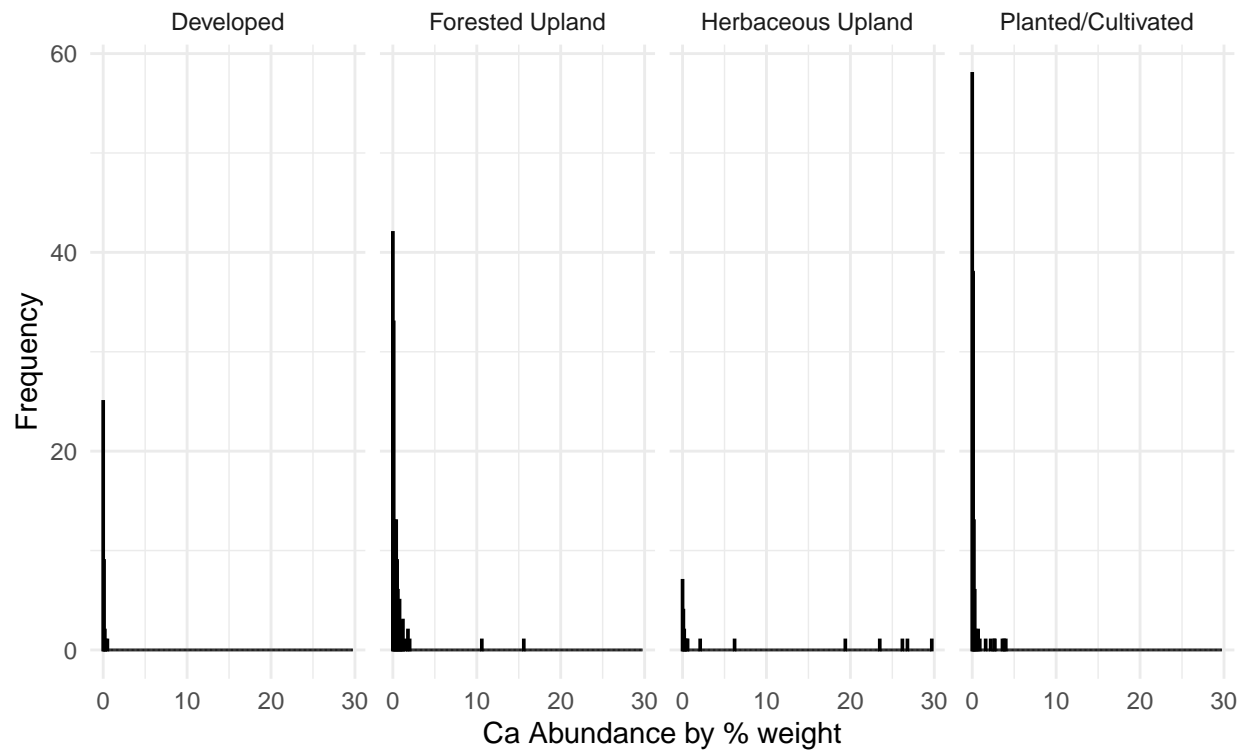


Figure: Distribution of Ca Abundance by Primary Land Cover

```
print(plot_Mg)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Distribution of Mg Abundance by Primary Land Cover

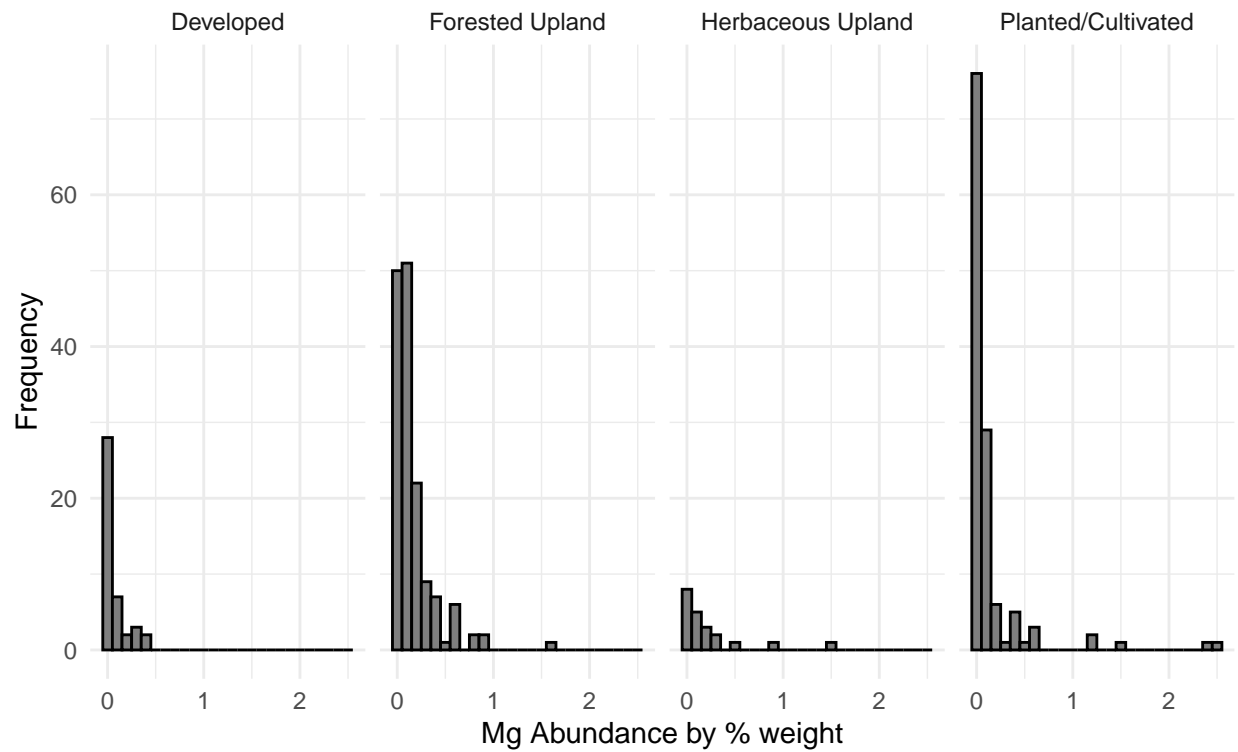


Figure: Distribution of Mg Abundance by Primary Land Cover

```
print(plot_K)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Distribution of K Abundance by Primary Land Cover

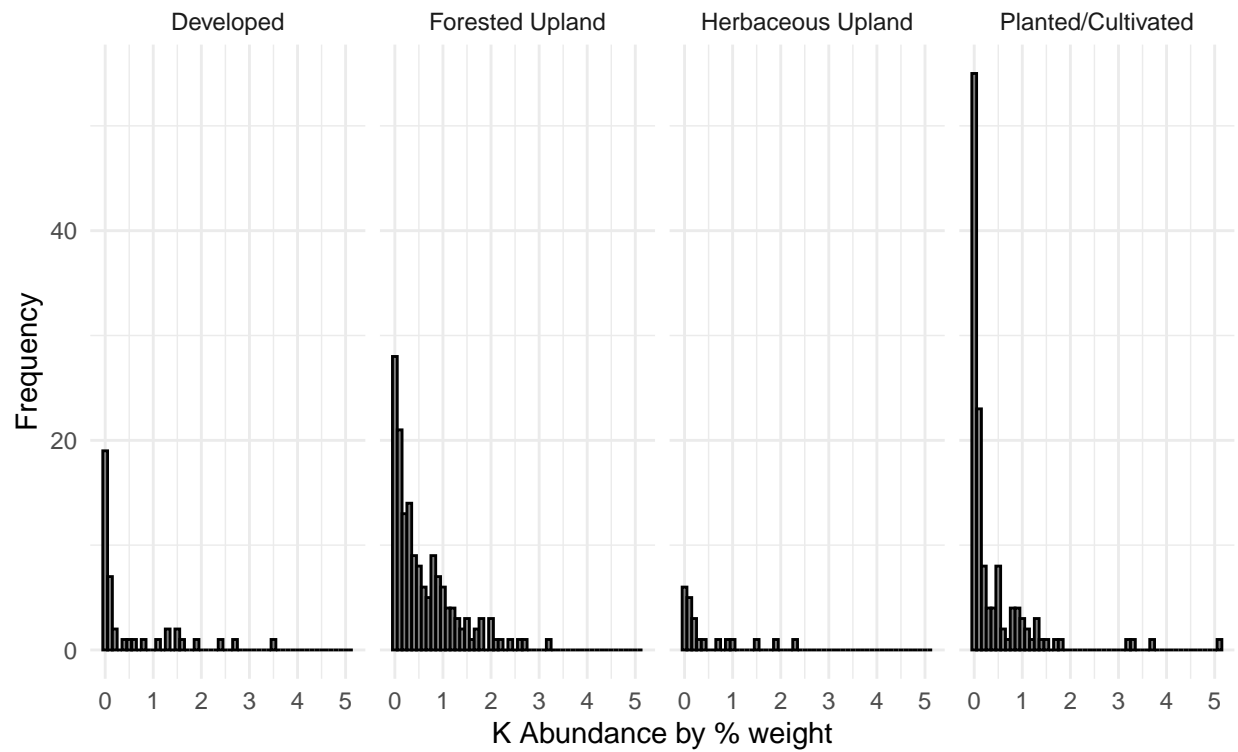


Figure: Distribution of K Abundance by Primary Land Cover

```
print(plot_P)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Distribution of P Abundance by Primary Land Cover

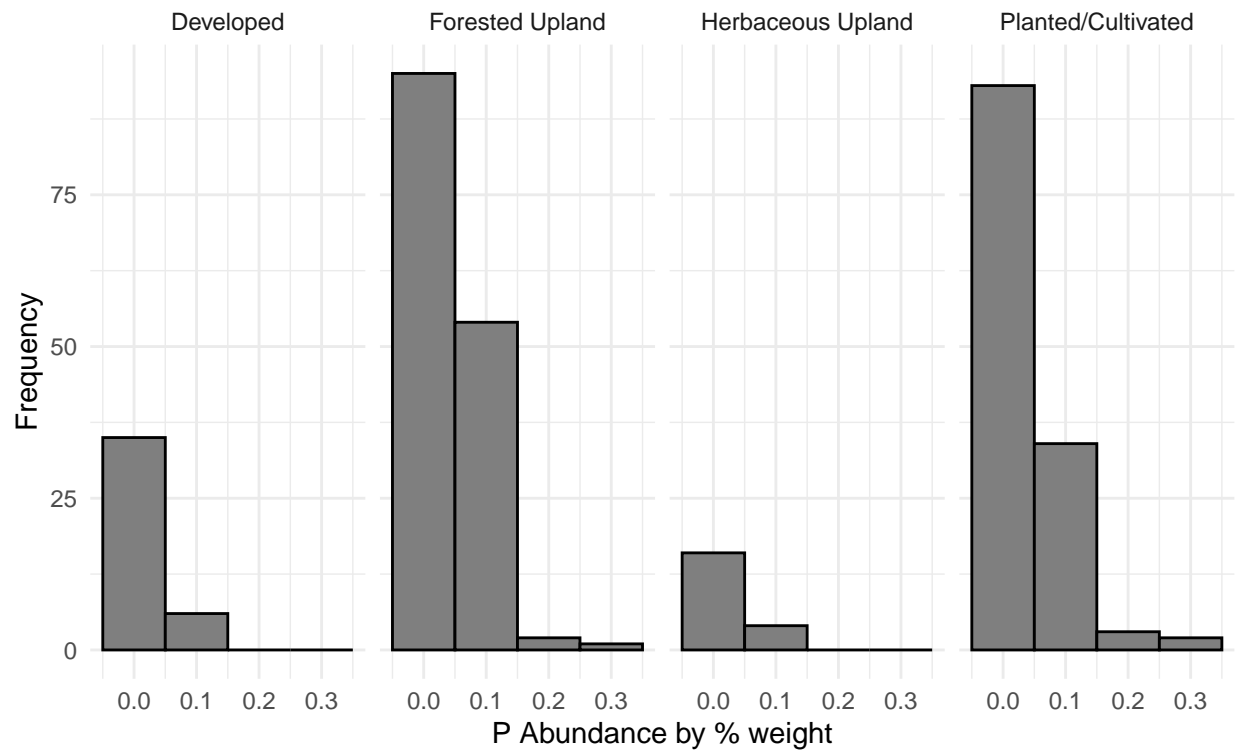


Figure: Distribution of P Abundance by Primary Land Cover

```
print(plot_S)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Distribution of S Abundance by Primary Land Cover

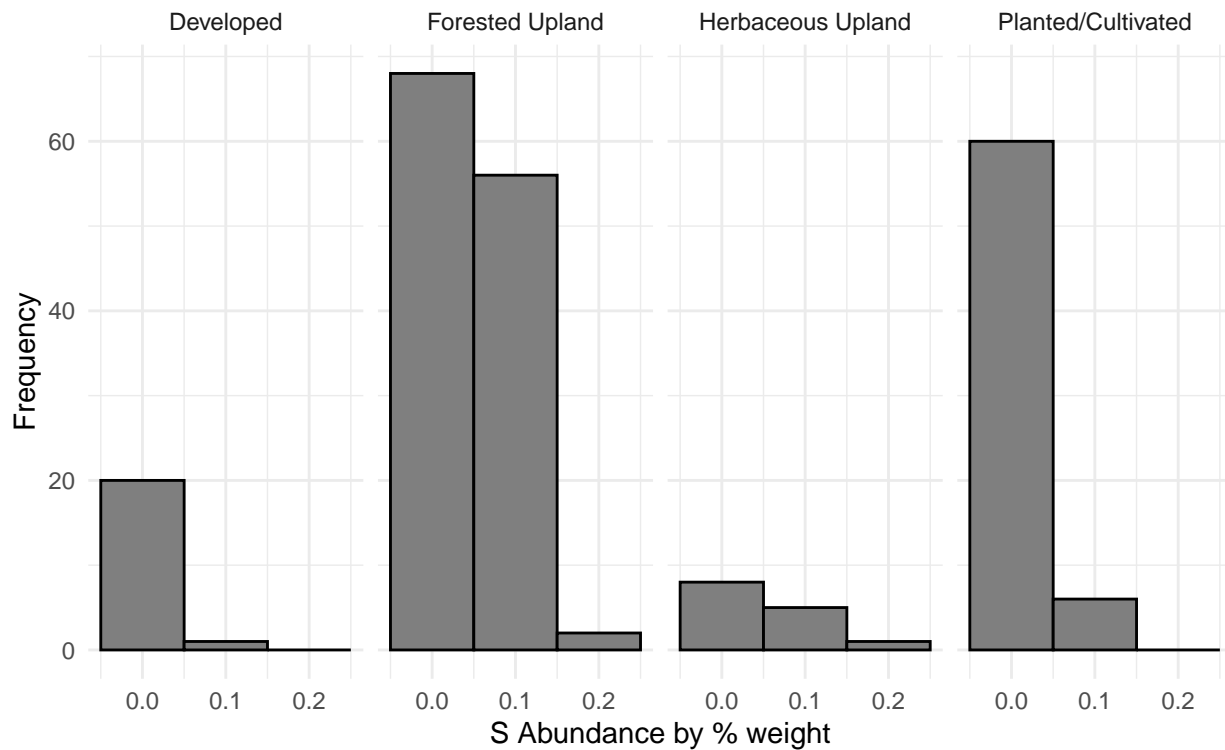


Figure: Distribution of S Abundance by Primary Land Cover

## Log Transformed Normality

```
create_log_transformed_histogram <- function(nutrient_name) {
  nutrient_data <- A_macros_long %>%
    filter(Nutrient == nutrient_name) %>%
    na.omit() %>%
    mutate(log_Abundance = log(Abundance + 1))

  hist_plot <- ggplot(nutrient_data, aes(x = log_Abundance, fill = LandCover1)) +
    geom_histogram(binwidth = 0.1, color = "black", position = "dodge") +
    labs(title = paste("Log-Transformed Distribution of", nutrient_name, "Abundance by LandCover1"),
         x = paste("Log-transformed", nutrient_name, "Abundance by % weight"),
         y = "Frequency",
         caption = paste("Figure: Log-transformed Distribution of", nutrient_name, "Abundance by LandCover1"),
         scale_fill_manual(values = c("LandCover1_Value1" = "skyblue", "LandCover1_Value2" = "salmon"))) + #
    theme_minimal() +
    theme(legend.position = "right") +
    facet_grid(. ~ LandCover1)

  return(hist_plot)
}

plot_log_C <- create_log_transformed_histogram("C")
```

```

plot_log_Ca <- create_log_transformed_histogram("Ca")
plot_log_Mg <- create_log_transformed_histogram("Mg")
plot_log_K <- create_log_transformed_histogram("K")
plot_log_P <- create_log_transformed_histogram("P")
plot_log_S <- create_log_transformed_histogram("S")

print(plot_log_C)

```

```

## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.

```

### Log-Transformed Distribution of C Abundance by LandCover1

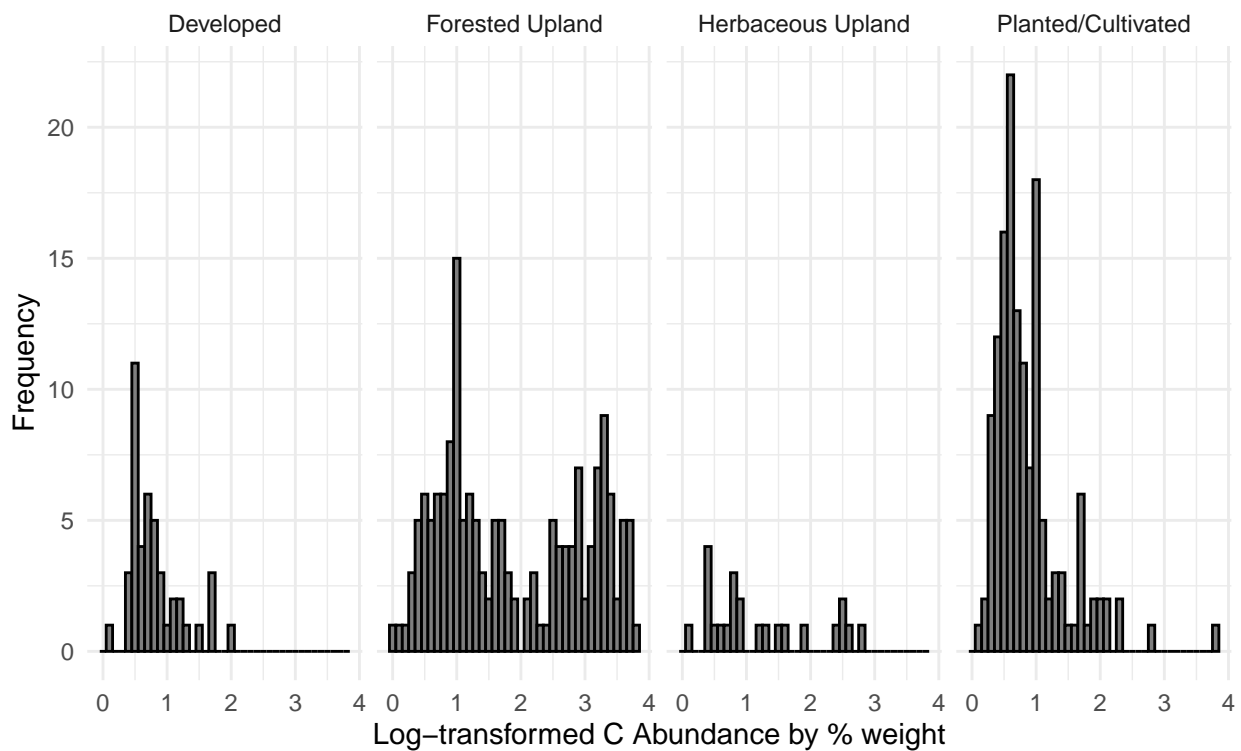


Figure: Log-transformed Distribution of C Abundance by LandCover1

```

print(plot_log_Ca)

```

```

## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.

```

## Log-Transformed Distribution of Ca Abundance by LandCover1

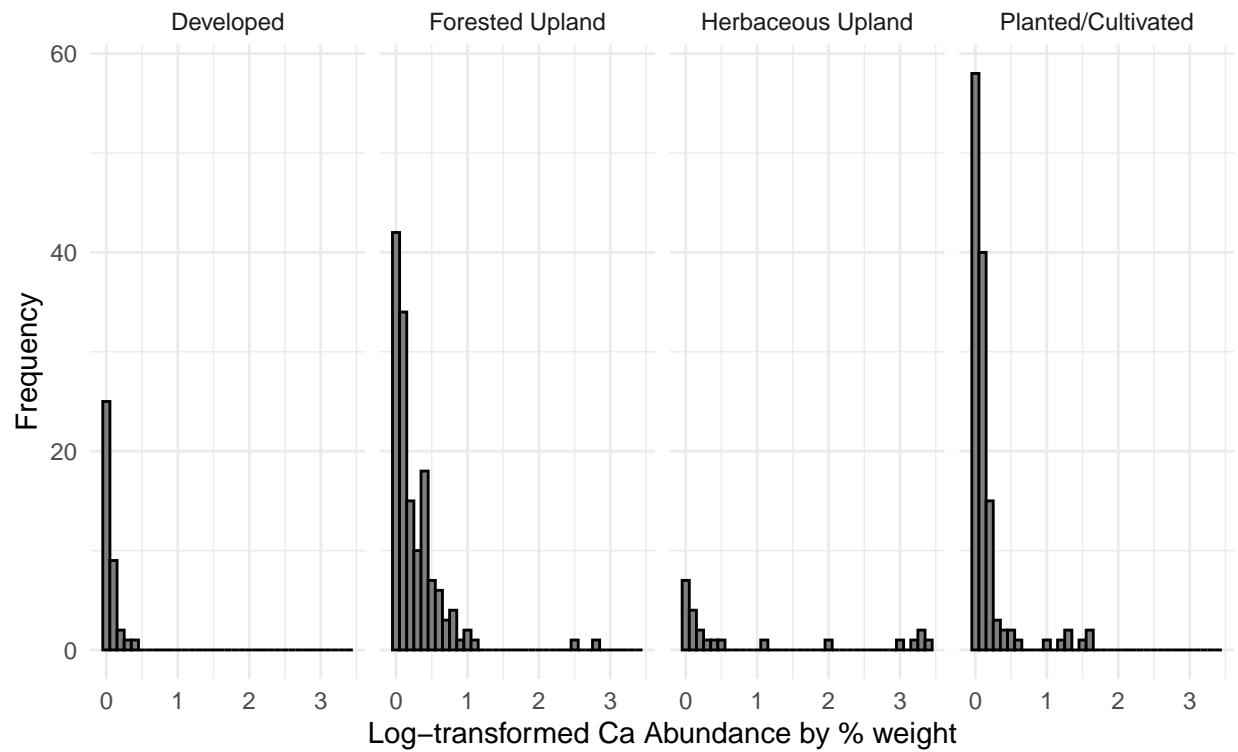


Figure: Log-transformed Distribution of Ca Abundance by LandCover1

```
print(plot_log_Mg)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```



## Log-Transformed Distribution of Mg Abundance by LandCover1

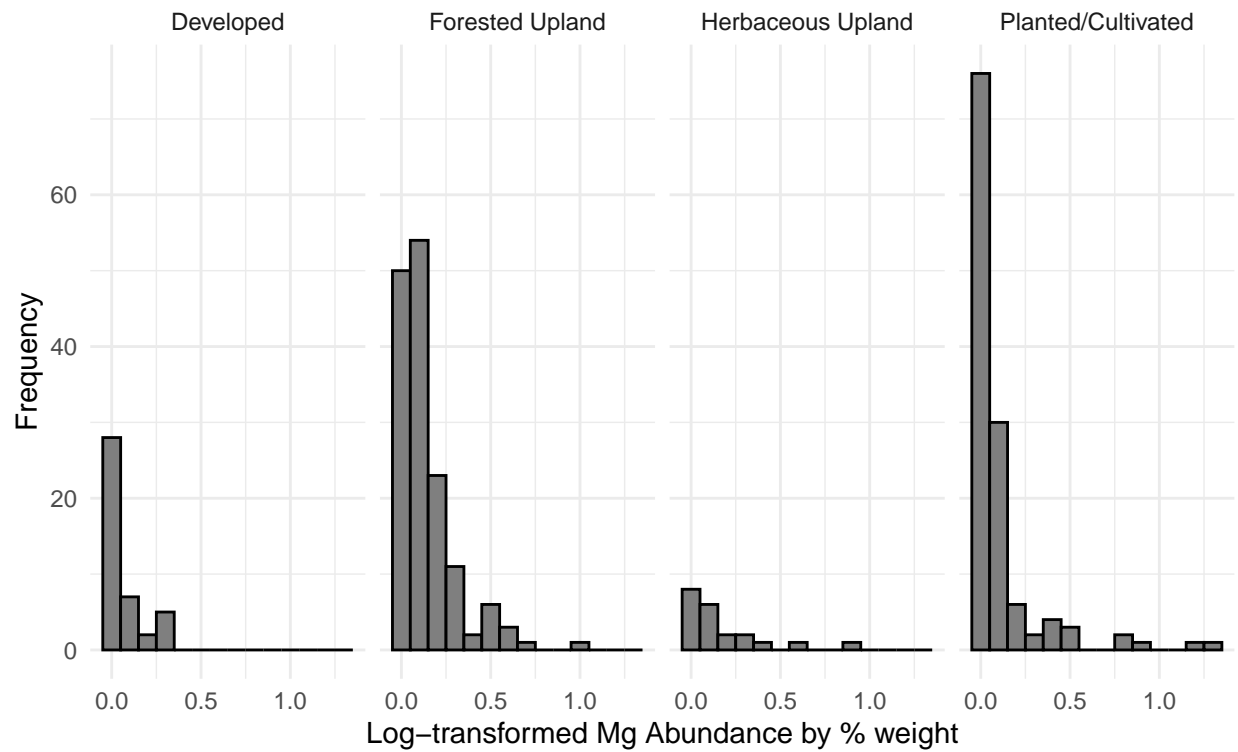


Figure: Log-transformed Distribution of Mg Abundance by LandCover1

```
print(plot_log_K)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Log-Transformed Distribution of K Abundance by LandCover1

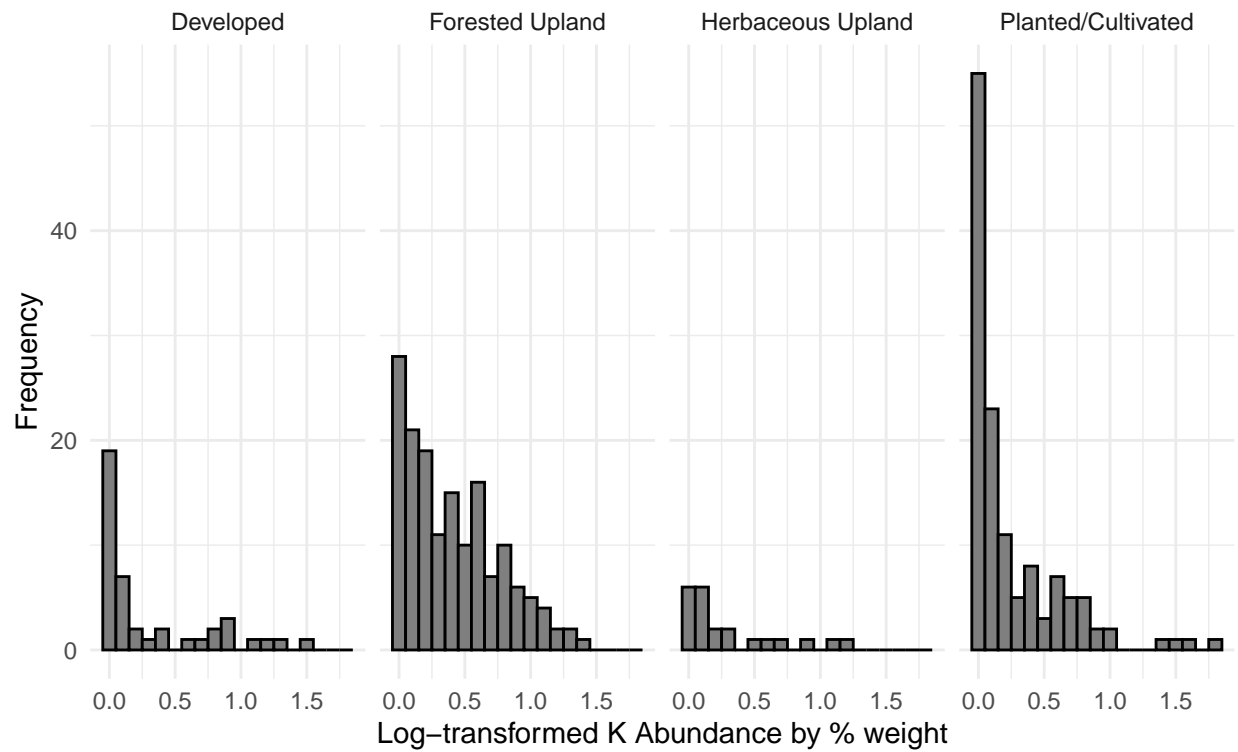


Figure: Log-transformed Distribution of K Abundance by LandCover1

```
print(plot_log_P)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Log-Transformed Distribution of P Abundance by LandCover1

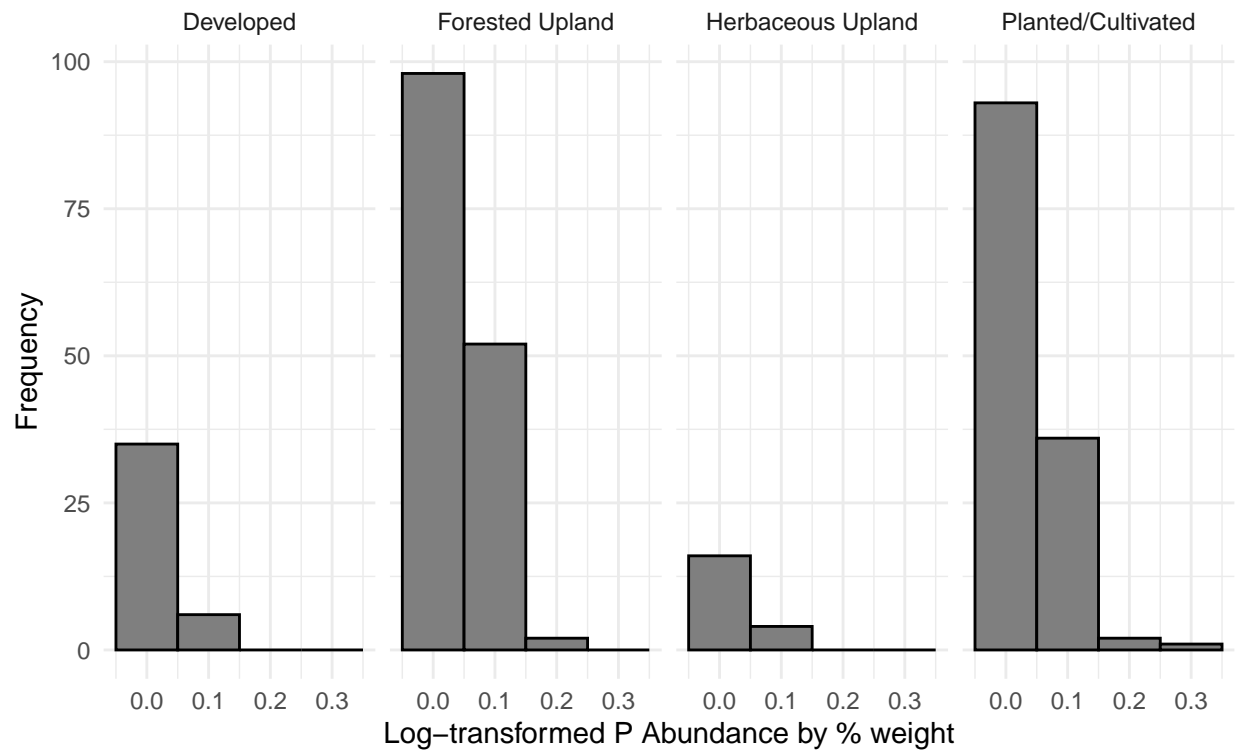


Figure: Log-transformed Distribution of P Abundance by LandCover1

```
print(plot_log_S)
```

```
## Warning: No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
## No shared levels found between 'names(values)' of the manual scale and the
## data's fill values.
```

## Log-Transformed Distribution of S Abundance by LandCover1

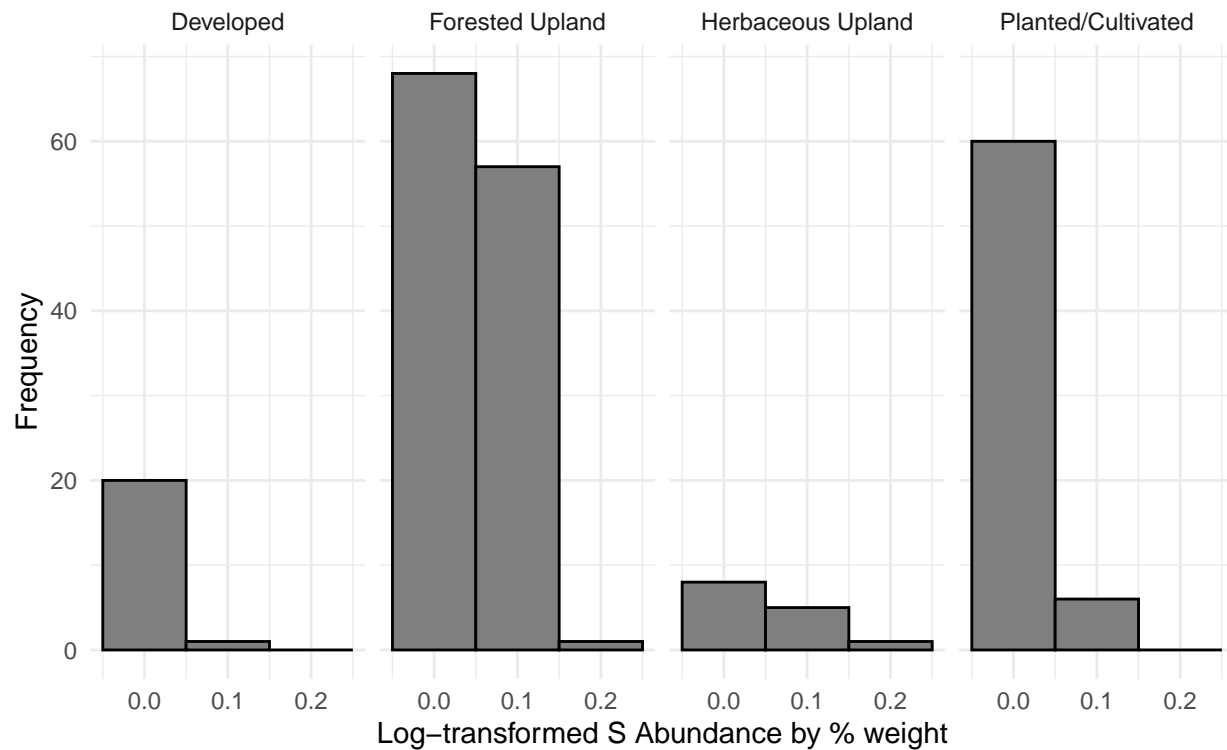


Figure: Log-transformed Distribution of S Abundance by LandCover1

```
shapiro_test_log <- function(nutrient_name) {
  nutrient_data <- A_macros_long %>%
    filter(Nutrient == nutrient_name) %>%
    na.omit() %>%
    mutate(log_Abundance = log(Abundance + 1))

  shapiro_result <- shapiro.test(nutrient_data$log_Abundance)

  return(data.frame(Nutrient = nutrient_name,
                    W_statistic = shapiro_result$statistic,
                    p_value = shapiro_result$p.value))
}

shapiro_results_log <- bind_rows(
  shapiro_test_log("C"),
  shapiro_test_log("Ca"),
  shapiro_test_log("Mg"),
  shapiro_test_log("K"),
  shapiro_test_log("P"),
  shapiro_test_log("S")
)

print(shapiro_results_log)
```

```
##      Nutrient W_statistic    p_value
```

```
## W...1      C  0.8536849 3.045843e-18
## W...2     Ca  0.4702728 2.757250e-30
## W...3     Mg  0.6322265 2.253176e-26
## W...4      K  0.8315857 7.019074e-19
## W...5      P  0.7904033 9.257088e-21
## W...6      S  0.8237032 2.500237e-15
```

```
shapiro_results_log1 <- shapiro_results_log %>%
  select(Nutrient, p_value) # Modify this depending on your actual column names

kable(shapiro_results_log1, col.names = c("Nutrient", "P-Value"), caption = "Shapiro-Wilk Test Results ")
```

Table 1: Shapiro-Wilk Test Results for Nutrients

	Nutrient	P-Value
W...1	C	0
W...2	Ca	0
W...3	Mg	0
W...4	K	0
W...5	P	0
W...6	S	0

```
shapiro_test_double_log <- function(nutrient_name) {
  nutrient_data <- A_macros_long %>%
    filter(Nutrient == nutrient_name) %>%
    na.omit() %>%
    mutate(log_Abundance = log(Abundance + 1), # First log transformation
           log_log_Abundance = log(log_Abundance + 1)) # Second log transformation

  # Perform the Shapiro-Wilk test on the second log-transformed data
  shapiro_result <- shapiro.test(nutrient_data$log_log_Abundance)

  # Return the result (p-value and W statistic)
  return(data.frame(Nutrient = nutrient_name,
                    W_statistic = shapiro_result$statistic,
                    p_value = shapiro_result$p.value))
}

shapiro_results_double_log <- bind_rows(
  shapiro_test_double_log("C"),
  shapiro_test_double_log("Ca"),
  shapiro_test_double_log("Mg"),
  shapiro_test_double_log("K"),
  shapiro_test_double_log("P"),
  shapiro_test_double_log("S")
)

print(shapiro_results_double_log)
```

```
##      Nutrient W_statistic      p_value
```

```
## W...1      C    0.9293899 2.557148e-12
## W...2     Ca    0.6346716 4.355717e-26
## W...3     Mg    0.7096506 7.388525e-24
## W...4      K    0.8681129 1.026683e-16
## W...5      P    0.8136264 1.201978e-19
## W...6      S    0.8358042 9.135612e-15
```

## Kruskal Wallis and Dunn Tests

```
nutrient_C <- A_macros_long %>%
  filter(Nutrient == "C") %>%
  na.omit()

# Run Kruskal-Wallis test for C
kruskal_C <- kruskal.test(Abundance ~ LandCover1, data = nutrient_C)
print("Kruskal-Wallis Test Result for C:")
```

### Kruskal-Wallis and Dunn's Test for Nutrient C

```
## [1] "Kruskal-Wallis Test Result for C:"
```

```
print(kruskal_C)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Abundance by LandCover1
## Kruskal-Wallis chi-squared = 84.206, df = 3, p-value < 2.2e-16
```

```
dunn_C <- dunn.test(nutrient_C$Abundance, nutrient_C$LandCover1, kw = TRUE, label = TRUE)
```

```
##  Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 84.2062, df = 3, p-value = 0
##
##
##                               Comparison of x by group
##                               (No adjustment)
## Col Mean-|
## Row Mean |   Develope   Forested   Herbaceo
## -----+-----
## Forested |  -6.055791
##           |    0.0000*
##           |
## Herbaceo  |  -1.574664   2.796011
##           |    0.0577    0.0026*
##           |
## Planted/ |  -0.375575   8.426413   1.515282
```

```
##          |      0.3536      0.0000*      0.0649
##
## alpha = 0.05
## Reject Ho if p <= alpha/2

print(dunn_C)

## $chi2
## [1] 84.20621
##
## $Z
## [1] -6.0557916 -1.5746650  2.7960114 -0.3755753  8.4264135  1.5152829
##
## $P
## [1] 6.986442e-10 5.766690e-02 2.586878e-03 3.536163e-01 1.782105e-17
## [6] 6.485039e-02
##
## $P.adjusted
## [1] 6.986442e-10 5.766690e-02 2.586878e-03 3.536163e-01 1.782105e-17
## [6] 6.485039e-02
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```

## Kruskal-Wallis and Dunn's Test for Nutrient Ca

```
nutrient_Ca <- A_macros_long %>%
  filter(Nutrient == "Ca") %>%
  na.omit()

# Run Kruskal-Wallis test for Ca
kruskal_Ca <- kruskal.test(Abundance ~ LandCover1, data = nutrient_Ca)
print("Kruskal-Wallis Test Result for Ca:")
```

```
## [1] "Kruskal-Wallis Test Result for Ca:"
```

```
print(kruskal_Ca)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: Abundance by LandCover1
## Kruskal-Wallis chi-squared = 36.783, df = 3, p-value = 5.115e-08
```

```
dunn_Ca <- dunn.test(nutrient_Ca$Abundance, nutrient_Ca$LandCover1, kw = TRUE, label = TRUE)
```

```
## Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 36.7826, df = 3, p-value = 0
##
##
## Comparison of x by group
## (No adjustment)
## Col Mean-|
## Row Mean |   Develope   Forested   Herbaceo
## -----+-----
## Forested | -5.098869
##          |  0.0000*
##          |
## Herbaceo | -4.182923 -0.783437
##          |  0.0000*   0.2167
##          |
## Planted/ | -2.484682  3.877070  2.852645
##          |  0.0065*   0.0001*   0.0022*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
print(dunn_Ca)
```

```
## $chi2
## [1] 36.7826
##
## $Z
## [1] -5.0988693 -4.1829233 -0.7834378 -2.4846827  3.8770701  2.8526455
##
## $P
## [1] 1.708442e-07 1.438922e-05 2.166850e-01 6.483347e-03 5.286096e-05
## [6] 2.167848e-03
##
## $P.adjusted
## [1] 1.708442e-07 1.438922e-05 2.166850e-01 6.483347e-03 5.286096e-05
## [6] 2.167848e-03
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```



## Kruskal-Wallis and Dunn's Test for Nutrient Mg

```
nutrient_Mg <- A_macros_long %>%
  filter(Nutrient == "Mg") %>%
  na.omit()

# Run Kruskal-Wallis test for Mg
kruskal_Mg <- kruskal.test(Abundance ~ LandCover1, data = nutrient_Mg)
print("Kruskal-Wallis Test Result for Mg:")

## [1] "Kruskal-Wallis Test Result for Mg:"

print(kruskal_Mg)

##
## Kruskal-Wallis rank sum test
##
## data: Abundance by LandCover1
## Kruskal-Wallis chi-squared = 32.385, df = 3, p-value = 4.341e-07

dunn_Mg <- dunn.test(nutrient_Mg$Abundance, nutrient_Mg$LandCover1, kw = TRUE, label = TRUE)

## Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 32.3853, df = 3, p-value = 0
##
##
## Comparison of x by group
## (No adjustment)
## Col Mean-|
## Row Mean | Develope Forested Herbaceo
## -----+-----
## Forested | -4.381838
##          | 0.0000*
##          |
## Herbaceo | -2.612559 0.284099
##          | 0.0045* 0.3882
##          |
## Planted/ | -1.139868 4.651942 2.100704
##          | 0.1272 0.0000* 0.0178*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2

print(dunn_Mg)

## $chi2
## [1] 32.38533
##
```

```
## $Z
## [1] -4.3818389 -2.6125595  0.2840998 -1.1398689  4.6519427  2.1007041
##
## $P
## [1] 5.884090e-06 4.493353e-03 3.881669e-01 1.271705e-01 1.644113e-06
## [6] 1.783347e-02
##
## $P.adjusted
## [1] 5.884090e-06 4.493353e-03 3.881669e-01 1.271705e-01 1.644113e-06
## [6] 1.783347e-02
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```

## Kruskal-Wallis and Dunn's Test for Nutrient K

```
nutrient_K <- A_macros_long %>%
  filter(Nutrient == "K") %>%
  na.omit()

# Run Kruskal-Wallis test for K
kruskal_K <- kruskal.test(Abundance ~ LandCover1, data = nutrient_K)
print("Kruskal-Wallis Test Result for K:")
```

```
## [1] "Kruskal-Wallis Test Result for K:"
```

```
print(kruskal_K)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: Abundance by LandCover1
## Kruskal-Wallis chi-squared = 24.268, df = 3, p-value = 2.196e-05
```

```
dunn_K <- dunn.test(nutrient_K$Abundance, nutrient_K$LandCover1, kw = TRUE, label = TRUE)
```

```
## Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 24.2678, df = 3, p-value = 0
##
##
## Comparison of x by group
## (No adjustment)
## Col Mean-|
```

```
## Row Mean |   Develope   Forested   Herbaceo
## -----+-----
## Forested |  -2.562854
##          |    0.0052*
##          |
## Herbaceo |  -0.308132   1.599511
##          |    0.3790    0.0549
##          |
## Planted/ |   0.683177   4.777082   0.877737
##          |    0.2472    0.0000*    0.1900
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
print(dunn_K)
```

```
## $chi2
## [1] 24.26776
##
## $Z
## [1] -2.5628547 -0.3081326  1.5995115  0.6831773  4.7770826  0.8777372
##
## $P
## [1] 5.190774e-03 3.789907e-01 5.485350e-02 2.472474e-01 8.892835e-07
## [6] 1.900432e-01
##
## $P.adjusted
## [1] 5.190774e-03 3.789907e-01 5.485350e-02 2.472474e-01 8.892835e-07
## [6] 1.900432e-01
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```

## Kruskal-Wallis and Dunn's Test for Nutrient P

```
nutrient_P <- A_macros_long %>%
  filter(Nutrient == "P") %>%
  na.omit()

# Run Kruskal-Wallis test for P
kruskal_P <- kruskal.test(Abundance ~ LandCover1, data = nutrient_P)
print("Kruskal-Wallis Test Result for P:")

## [1] "Kruskal-Wallis Test Result for P:"
```

```
print(kruskal_P)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: Abundance by LandCover1
## Kruskal-Wallis chi-squared = 13.429, df = 3, p-value = 0.003795
```

```
dunn_P <- dunn.test(nutrient_P$Abundance, nutrient_P$LandCover1, kw = TRUE, label = TRUE)
```

```
## Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 13.4287, df = 3, p-value = 0
##
##
## Comparison of x by group
## (No adjustment)
## Col Mean-|
## Row Mean | Develope Forested Herbaceo
## -----+-----
## Forested | -3.602941
## | 0.0002*
## |
## Herbaceo | -1.364699 1.100766
## | 0.0862 0.1355
## |
## Planted/ | -2.969818 0.866353 -0.661634
## | 0.0015* 0.1931 0.2541
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
print(dunn_P)
```

```
## $chi2
## [1] 13.42874
##
## $Z
## [1] -3.6029420 -1.3646998 1.1007668 -2.9698181 0.8663537 -0.6616344
##
## $P
## [1] 0.0001573179 0.0861737112 0.1354990826 0.0014898809 0.1931481038
## [6] 0.2541027713
##
## $P.adjusted
## [1] 0.0001573179 0.0861737112 0.1354990826 0.0014898809 0.1931481038
## [6] 0.2541027713
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
```

```
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```

## Kruskal-Wallis and Dunn's Test for Nutrient S

```
nutrient_S <- A_macros_long %>%
  filter(Nutrient == "S") %>%
  na.omit()

# Run Kruskal-Wallis test for S
kruskal_S <- kruskal.test(Abundance ~ LandCover1, data = nutrient_S)
print("Kruskal-Wallis Test Result for S:")
```

```
## [1] "Kruskal-Wallis Test Result for S:"
```

```
print(kruskal_S)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: Abundance by LandCover1
## Kruskal-Wallis chi-squared = 28.175, df = 3, p-value = 3.337e-06
```

```
dunn_S <- dunn.test(nutrient_S$Abundance, nutrient_S$LandCover1, kw = TRUE, label = TRUE)
```

```
## Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 28.1754, df = 3, p-value = 0
##
##
## Comparison of x by group
## (No adjustment)
## Col Mean-|
## Row Mean | Develope Forested Herbaceo
## -----+-----
## Forested | -4.317729
## | 0.0000*
## |
## Herbaceo | -2.924394 0.030834
## | 0.0017* 0.4877
## |
## Planted/ | -1.750698 3.811039 1.938489
## | 0.0400 0.0001* 0.0263
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
print(dunn_S)
```

```
## $chi2
## [1] 28.17545
##
## $Z
## [1] -4.317730 -2.924394  0.030835 -1.750699  3.811039  1.938490
##
## $P
## [1] 7.882122e-06 1.725637e-03 4.877006e-01 3.999891e-02 6.919196e-05
## [6] 2.628175e-02
##
## $P.adjusted
## [1] 7.882122e-06 1.725637e-03 4.877006e-01 3.999891e-02 6.919196e-05
## [6] 2.628175e-02
##
## $comparisons
## [1] "Developed - Forested Upland"
## [2] "Developed - Herbaceous Upland"
## [3] "Forested Upland - Herbaceous Upland"
## [4] "Developed - Planted/Cultivated"
## [5] "Forested Upland - Planted/Cultivated"
## [6] "Herbaceous Upland - Planted/Cultivated"
```

## Linear Regression

### Mean temperature and Precipitation as potential independent variables

```
# Set download directory
prism_set_dl_dir('./Data')
# Download temperature data for a specific region
get_prism_normals(type = "tmean",
                  resolution = "4km", # 4km spatial resolution (choose "4km" or "800m")
                  #mon = 1:12,
                  annual = TRUE,
                  keepZip = FALSE)
```

```
## |
##
## PRISM_tmean_30yr_normal_4kmM5_annual_bil.zip already exists. Skipping downloading.
```

```
## |=====

get_prism_normals(type = "ppt",
                  resolution = "4km", # 4km spatial resolution (choose "4km" or "800m")
                  #mon = 1:12,
                  annual = TRUE,
                  keepZip = FALSE)
```

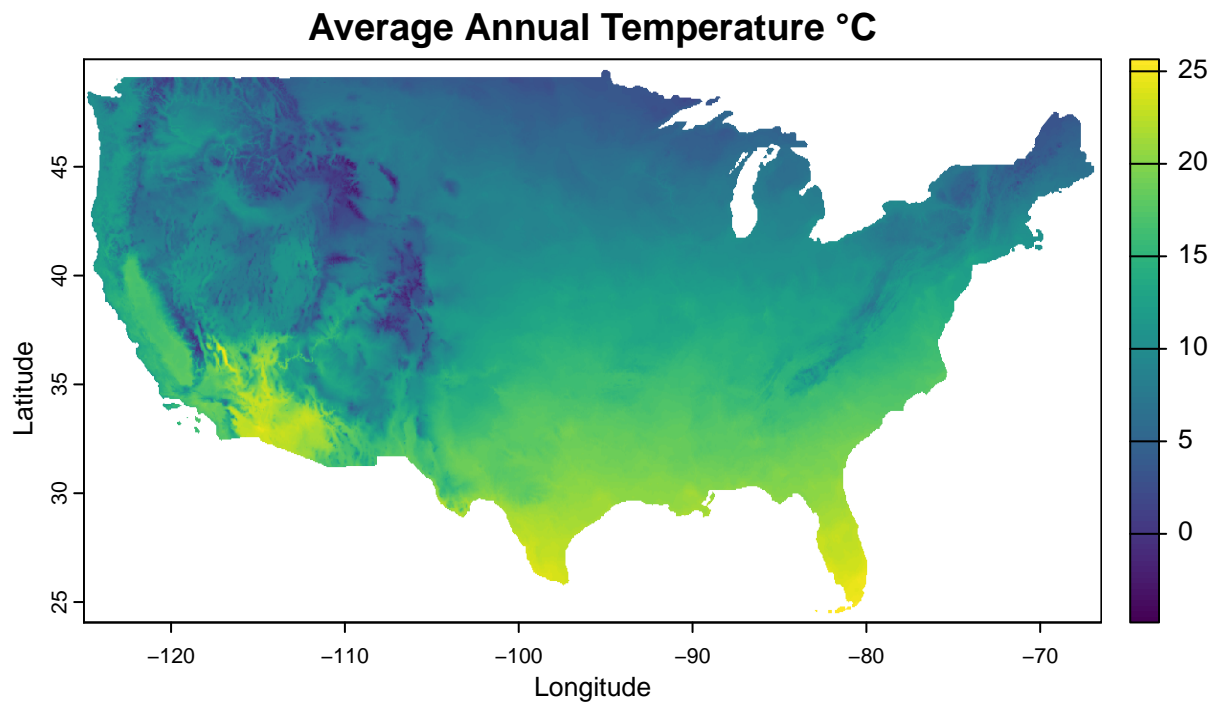
```
## |

##
## PRISM_ppt_30yr_normal_4kmM4_annual_bil.zip already exists. Skipping downloading.

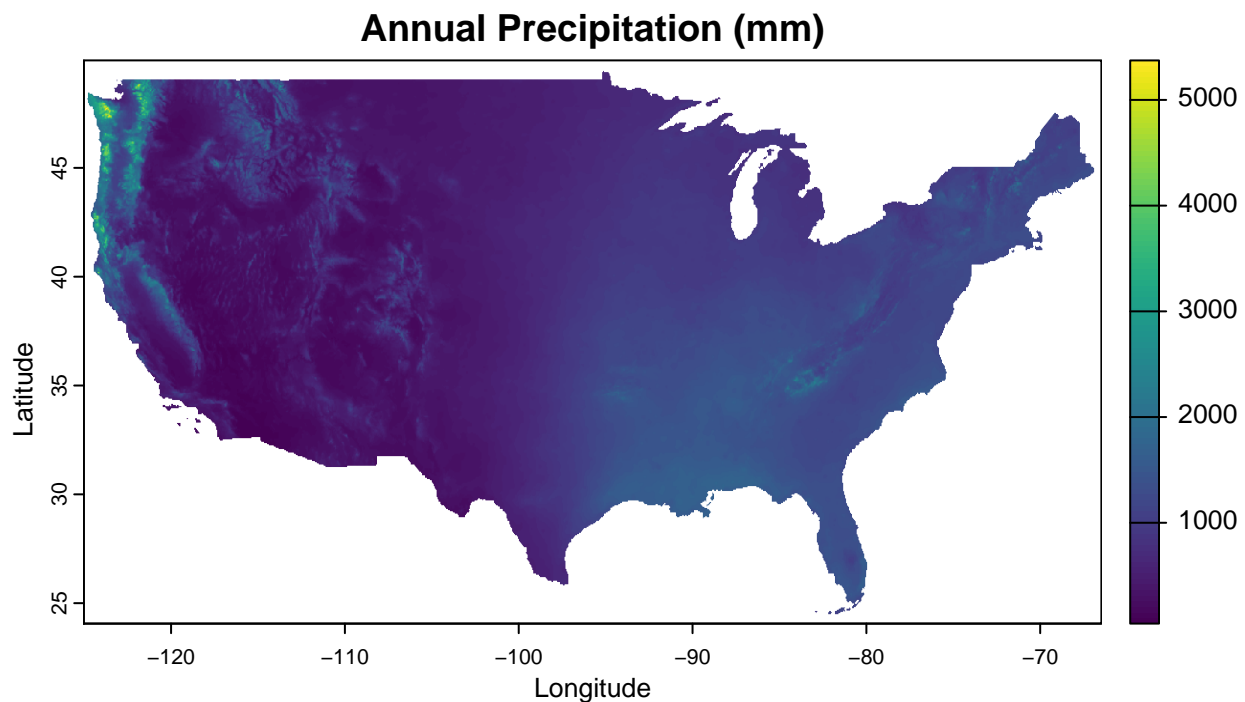
## |=====
```

```
prism_t <- rast("./Data/PRISM_tmean_30yr_normal_4kmM5_annual_bil/PRISM_tmean_30yr_normal_4kmM5_annual_bil.tif")
prism_p <- rast("./Data/PRISM_ppt_30yr_normal_4kmM4_annual_bil/PRISM_ppt_30yr_normal_4kmM4_annual_bil.tif")
```

```
temp_plot <- plot(prism_t,
  main = "Average Annual Temperature °C",
  xlab = "Longitude",
  ylab = "Latitude")
```



```
precipitation_plot <- plot(prism_p,
  main = "Annual Precipitation (mm)",
  xlab = "Longitude",
  ylab = "Latitude")
```



```
A_macros$Longitude <- as.numeric(A_macros$Longitude)
A_macros$Latitude <- as.numeric(A_macros$Latitude)
```

```
# Define the coordinates
coords <- A_macros %>%
  select(Longitude, Latitude)
# rename(x = Longitude, y = Latitude)
```

```
coords <- data.frame(coords)
```

```
points <- vect(coords, geom=c("Longitude", "Latitude"), crs = "EPSG:4326")
print(points)
```

```
## class      : SpatVector
## geometry    : points
## dimensions   : 382, 0 (geometries, attributes)
## extent      : -87.4227, -75.6325, 25.1376, 39.2693 (xmin, xmax, ymin, ymax)
## coord. ref.  : lon/lat WGS 84 (EPSG:4326)
```

```
points_nad83 <- project(points, crs(prism_t))
print(points_nad83)
```

```
## class      : SpatVector
```



```
## geometry      : points
## dimensions    : 382, 0 (geometries, attributes)
## extent       : -87.4227, -75.63251, 25.1376, 39.2693 (xmin, xmax, ymin, ymax)
## coord. ref.   : lon/lat NAD83
```

```
extracted_t <- extract(prism_t, points_nad83)
extracted_p <- extract(prism_p, points_nad83)
```

```
A_macros_tp <- A_macros %>%
  mutate(temperature = extracted_t,
         precipitation = extracted_p) %>%
  rename(clay = A_Tot_Clay,
         depth = A_Depth)
```

## Multiple Linear Regression

### Carbon

```
# convert Depth to average depth
A_macros_tp <- A_macros_tp %>%
  mutate(avg_depth = sapply(strsplit(as.character(depth), "-"),
                             function(x) mean(as.numeric(x))))) %>%
# tidy clay percentage data
  mutate(clay = na_if(clay, "N.D."),
         clay = as.numeric(clay)) %>%
  filter(LandCover1 != "Non-Natural Woody" &
         LandCover1 != "Shrubland")
```

```
## Warning: There was 1 warning in 'mutate()'.
## i In argument: 'clay = as.numeric(clay)'.
## Caused by warning:
## ! NAs introduced by coercion
```

```
A_macros_tp_C <- A_macros_tp %>%
  select(C,
         LandCover1,
         clay,
         avg_depth)
```

```
A_macros_tp_C$temperature <- A_macros_tp$temperature$PRISM_tmean_30yr_normal_4kmM5_annual_bil
A_macros_tp_C$precipitation <- A_macros_tp$precipitation$PRISM_ppt_30yr_normal_4kmM4_annual_bil
```

### View correlation

```
# Pivot all independent variables but not the dependent variable.
A_macros_tp_C_long <- A_macros_tp_C %>%
  select(C,
         clay,
         avg_depth,
```

```

    temperature,
    precipitation) %>%
pivot_longer(names_to = "variable", cols = -C, values_to = "value")

# Create and examine an initial scatterplot of data.
ggplot(A_macros_tp_C_long, aes(x = value, y = C, color = variable)) +
  geom_point() +
  facet_wrap(~variable, scales = "free_x") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs(title = "Scatterplots of Each Independent Var v.s. Dependent Var",
x = "Independent Variable (Name in Subtitles)",
y = "Total Recreation Acreage")

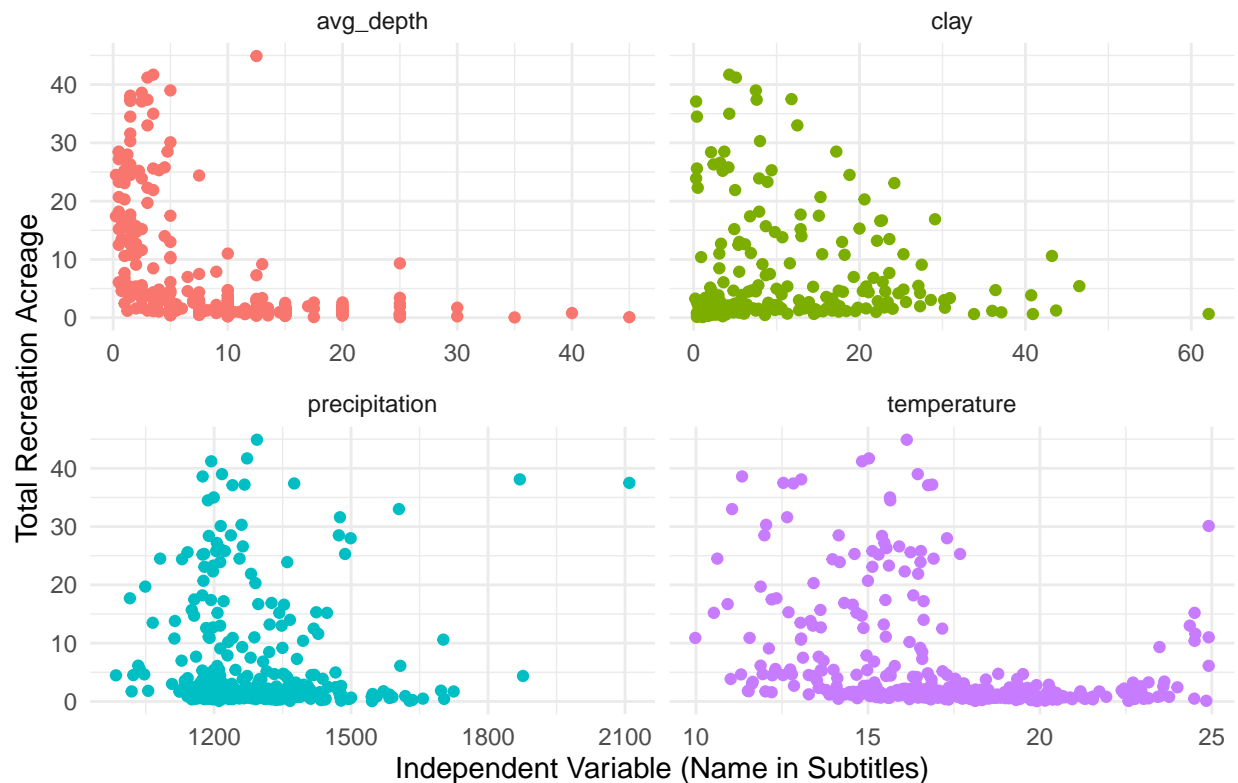
```

```

## Warning: Removed 93 rows containing missing values or values outside the scale range
## ('geom_point()').

```

## Scatterplots of Each Independent Var v.s. Dependent Var



```

# Correlation matrix
A_macros_tp_C1 <- A_macros_tp_C %>%
  select(C, clay, avg_depth, temperature, precipitation) %>%
  na.omit()
cor_matrix_C <- cor(A_macros_tp_C1)

View(cor_matrix_C)

```

```
# Set Landcover column to be a factor with a reference level of Forested.
A_macros_tp_C$LandCover1 <- factor(A_macros_tp_C$LandCover1,
                                   levels = c("Forested Upland", "Developed",
                                              "Planted/Cultivated", "Herbaceous Upland"))
```

```
C.mlm1 <- lm(C ~ clay + avg_depth +
             temperature + precipitation +
             LandCover1,
             data = A_macros_tp_C)
```

```
summary(C.mlm1)
```

## Model Building

```
##
## Call:
## lm(formula = C ~ clay + avg_depth + temperature + precipitation +
##      LandCover1, data = A_macros_tp_C)
##
## Residuals:
```

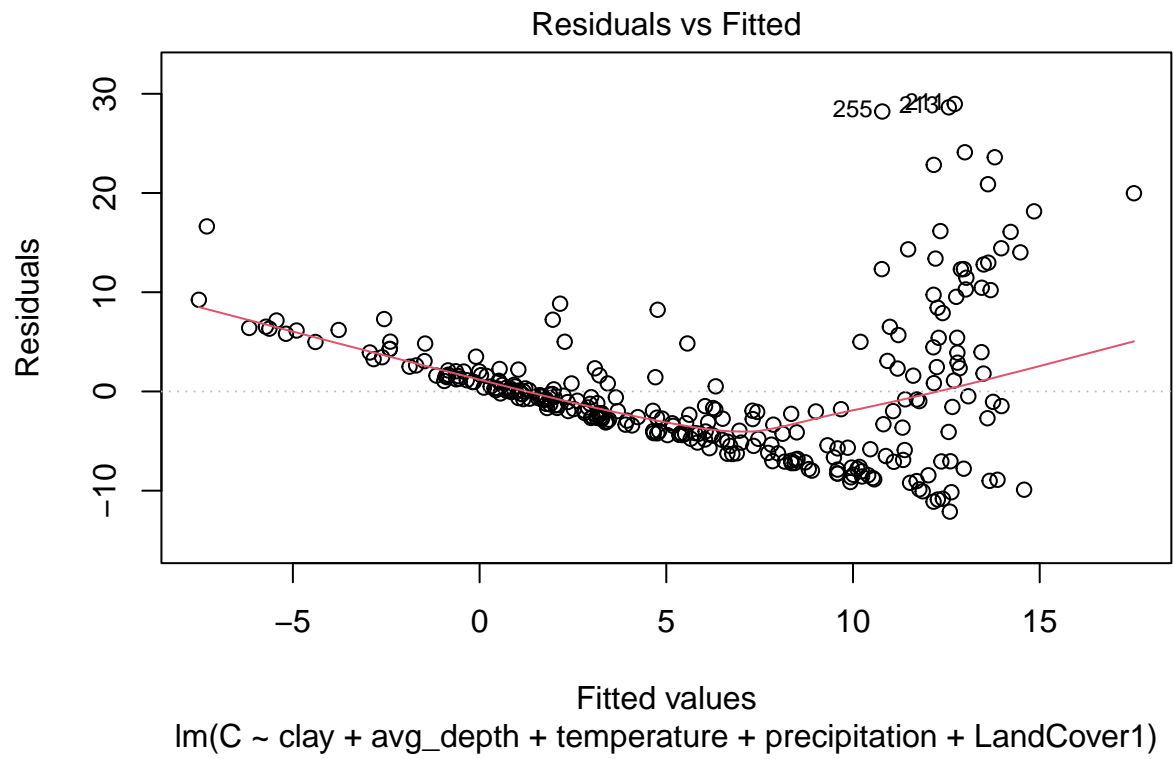
	Min	1Q	Median	3Q	Max
	-12.1126	-4.2628	-0.8887	2.1011	28.9745

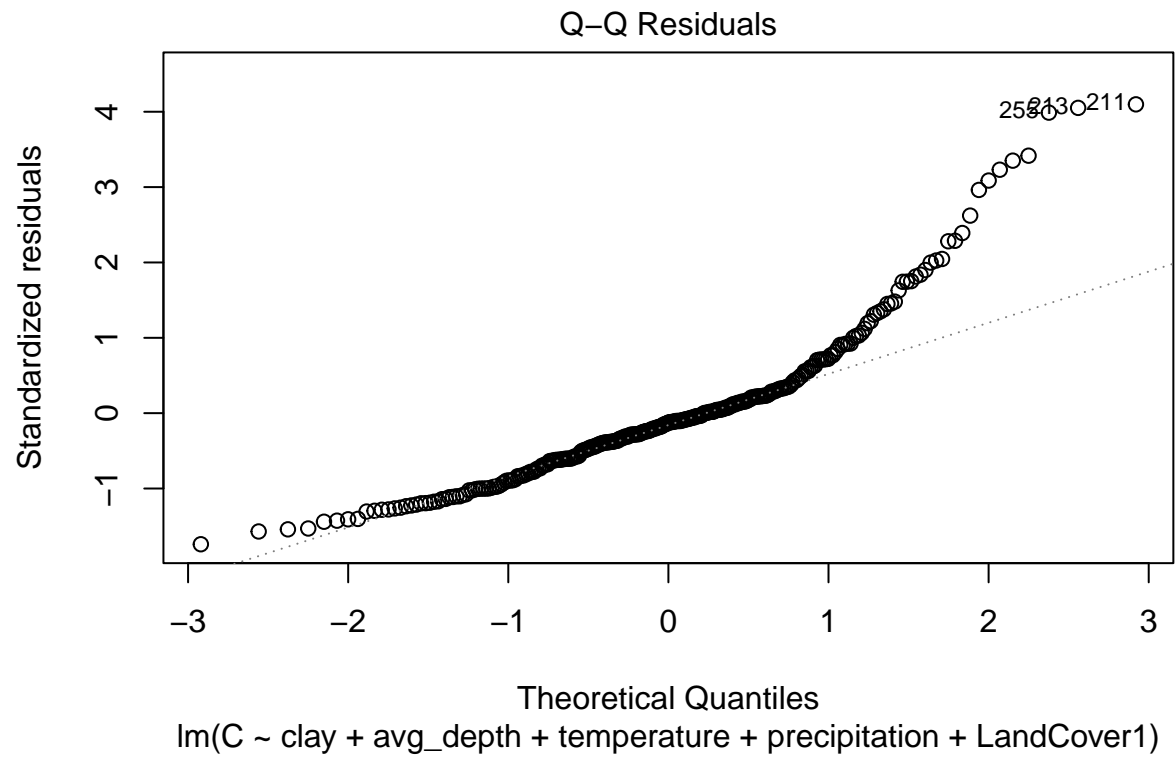
```
##
## Coefficients:
```

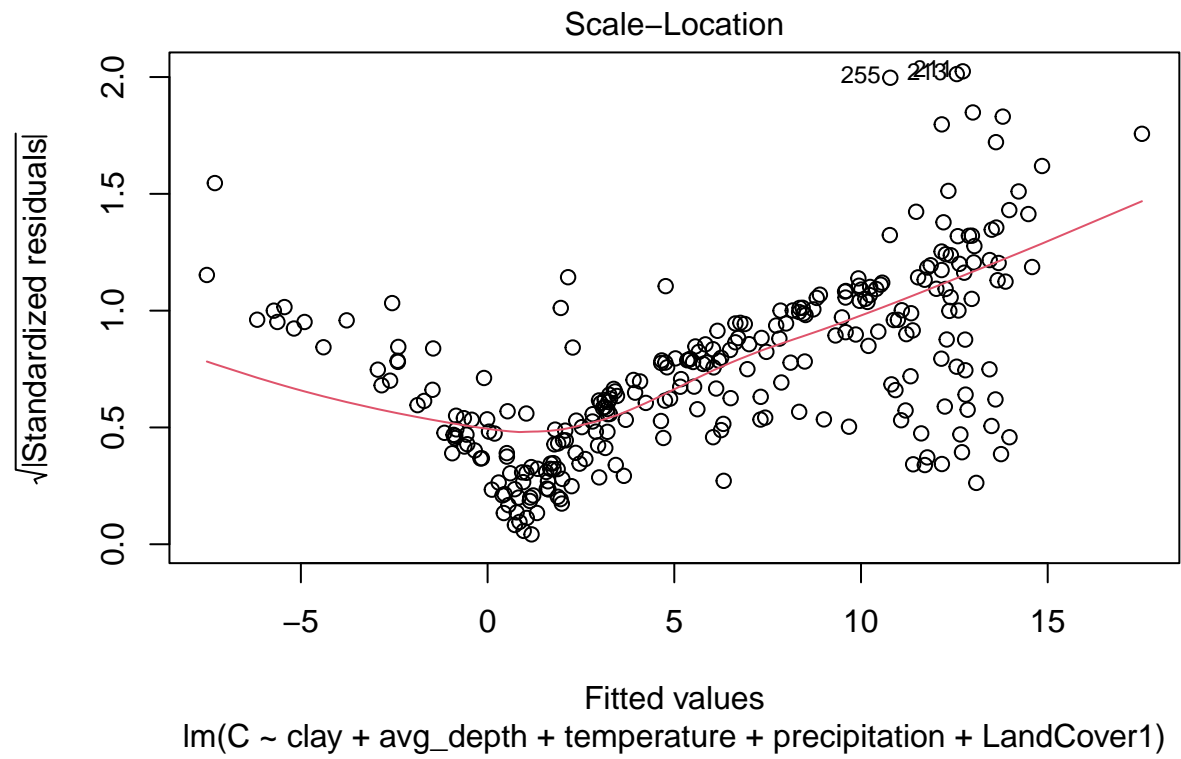
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.430848	4.419880	3.265	0.00123 **
clay	-0.136066	0.048460	-2.808	0.00534 **
avg_depth	-0.492178	0.085003	-5.790	1.90e-08 ***
temperature	-0.358395	0.197333	-1.816	0.07042 .
precipitation	0.004708	0.003196	1.473	0.14190
LandCover1Developed	-6.738402	1.427592	-4.720	3.74e-06 ***
LandCover1Planted/Cultivated	-5.376025	1.036547	-5.186	4.13e-07 ***
LandCover1Herbaceous Upland	-4.068050	2.073189	-1.962	0.05073 .

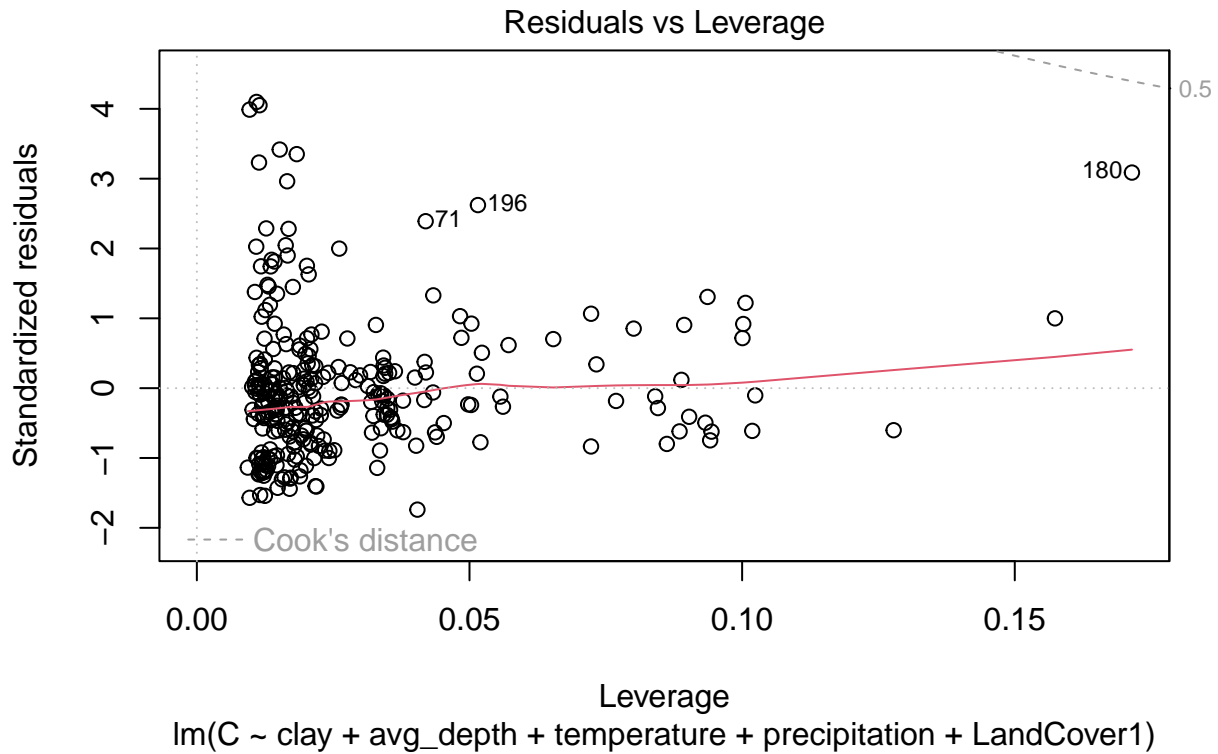
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.109 on 278 degrees of freedom
## (90 observations deleted due to missingness)
## Multiple R-squared:  0.3678, Adjusted R-squared:  0.3519
## F-statistic: 23.11 on 7 and 278 DF, p-value: < 2.2e-16
```

```
plot(C.mlm1)
```









```
A_macros_tp_C_log <- A_macros_tp_C %>%
  mutate(C_log = log(C),
         clay_log = log(clay),
         temp_log = log(temperature),
         precip_log = log(precipitation),
         depth_log = log(avg_depth))
```

log transformed data

```
C.mlm2 <- lm(C_log ~ clay_log + depth_log +
             temp_log + precip_log +
             LandCover1,
             data = A_macros_tp_C_log)

summary(C.mlm2)
```

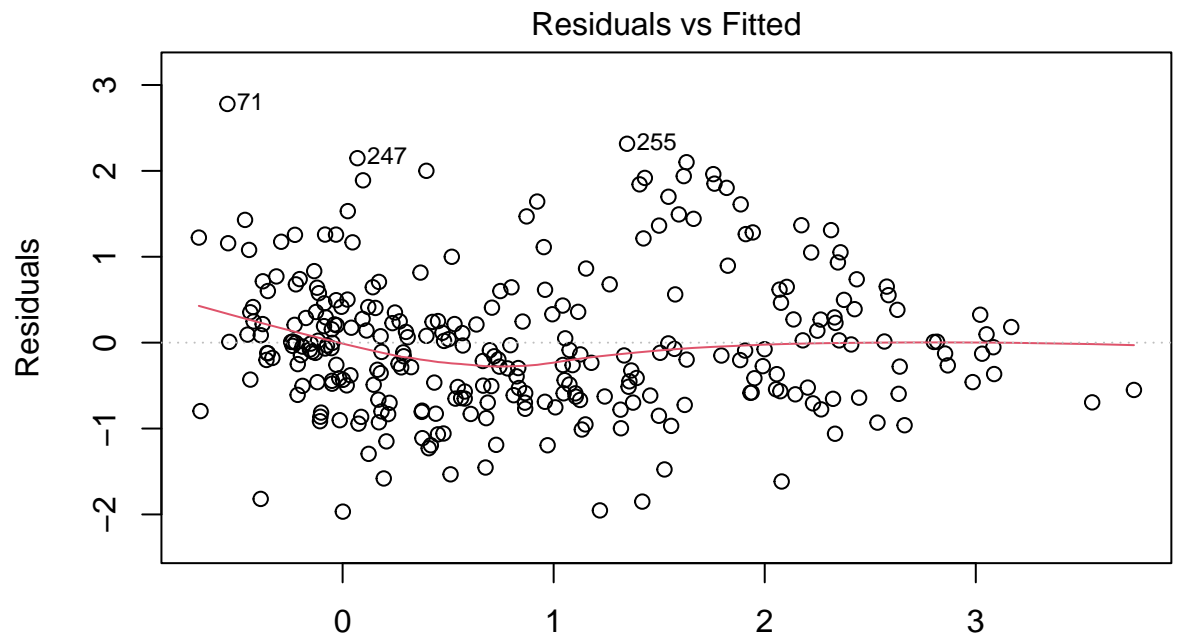
Refit with log-transformed data

##

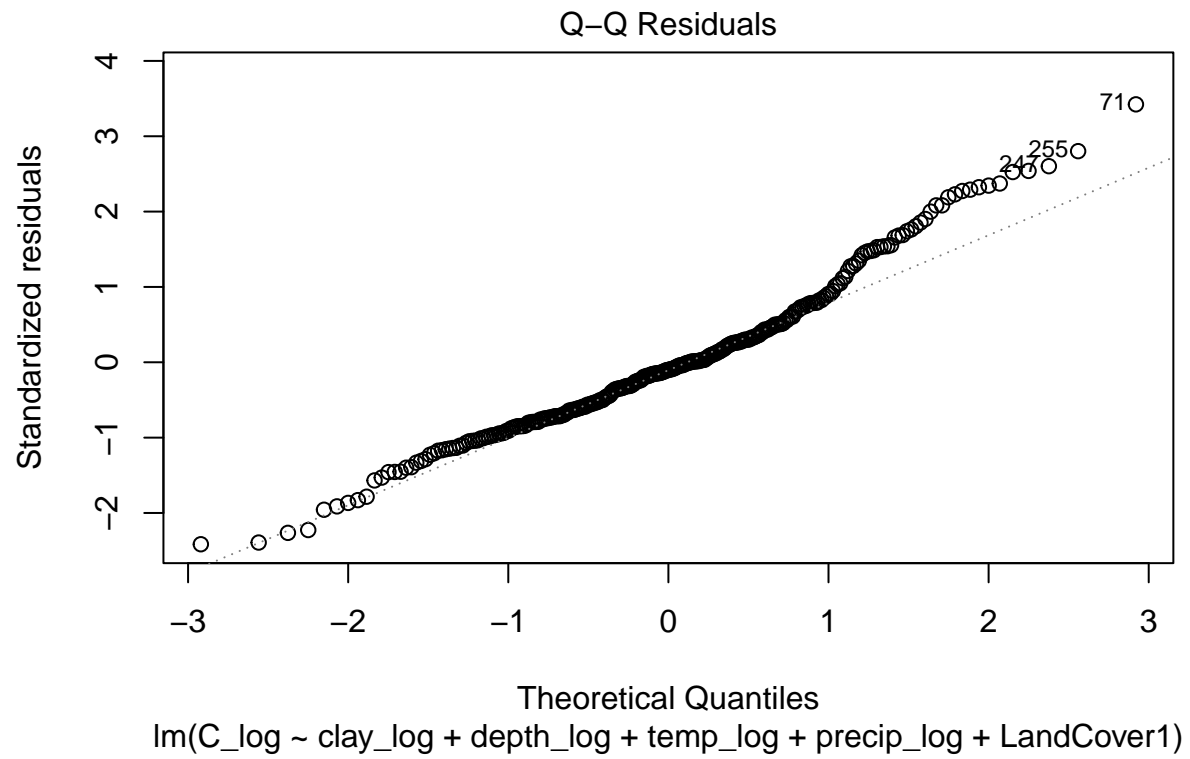
```
## Call:
## lm(formula = C_log ~ clay_log + depth_log + temp_log + precip_log +
##      LandCover1, data = A_macros_tp_C_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.96731 -0.57695 -0.08346  0.40664  2.77907
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.19567    3.51165   1.195   0.233
## clay_log         0.02505    0.05157   0.486   0.627
## depth_log       -0.72743    0.06114 -11.897 < 2e-16 ***
## temp_log        -0.43669    0.41168  -1.061   0.290
## precip_log      -0.07105    0.51065  -0.139   0.889
## LandCover1Developed -0.79293    0.16885  -4.696 4.17e-06 ***
## LandCover1Planted/Cultivated -0.57586    0.12121  -4.751 3.25e-06 ***
## LandCover1Herbaceous Upland -0.23926    0.24083  -0.993   0.321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8308 on 278 degrees of freedom
## (90 observations deleted due to missingness)
## Multiple R-squared:  0.5973, Adjusted R-squared:  0.5871
## F-statistic: 58.9 on 7 and 278 DF, p-value: < 2.2e-16
```

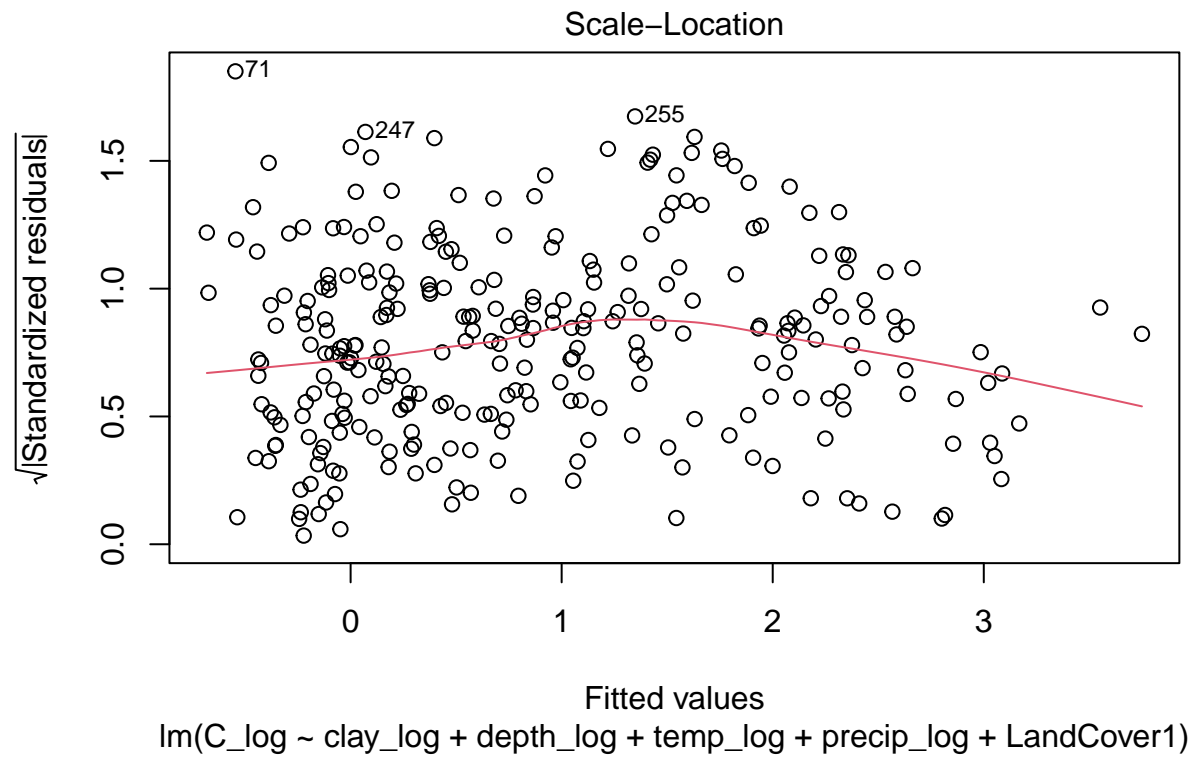
```
plot(C.mlm2)
```

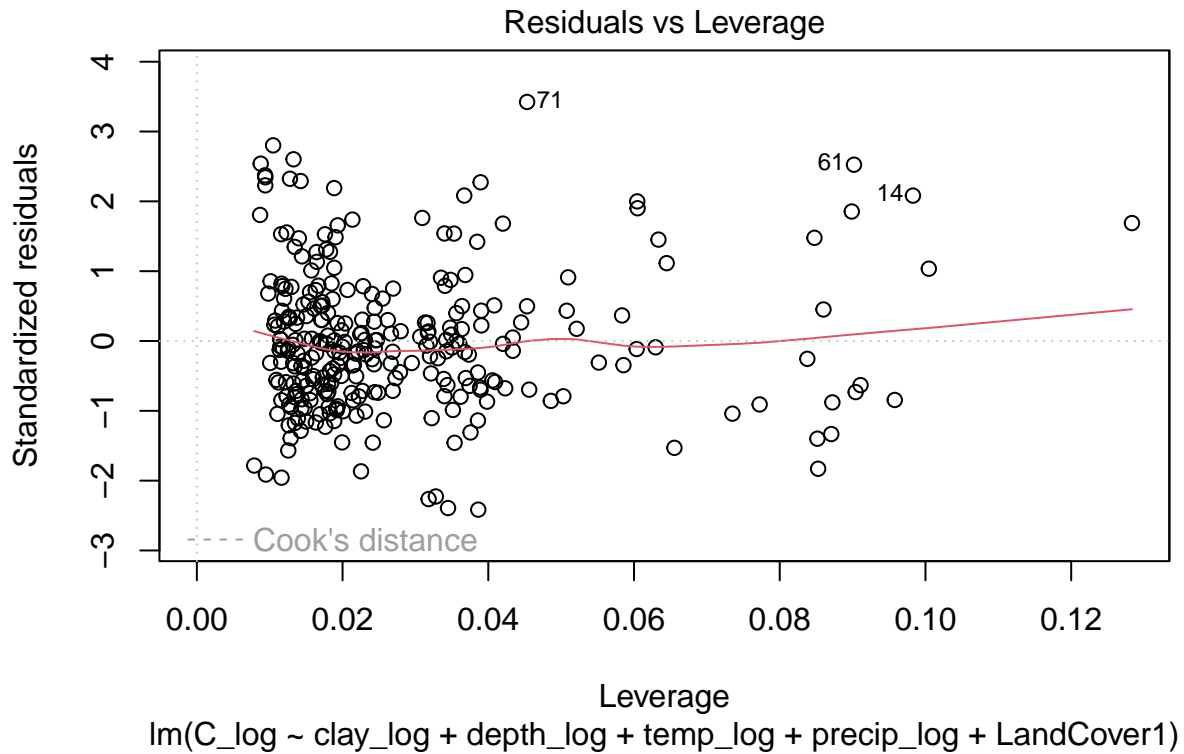




$\text{lm}(\text{C\_log} \sim \text{clay\_log} + \text{depth\_log} + \text{temp\_log} + \text{precip\_log} + \text{LandCover1})$







## Linear Model visualization

```
# Simulate temp and precip data
temp_vector <- rep(seq(from = 10, to = 30), 4)
precip_vector <- rep(seq(from = 800, to = 2000, length.out = 21), 4)
clay_vector <- rep(seq(from = 0.1, to = 40, length.out = 21), 4)
depth_vector <- rep(seq(from = 5, to = 35, length.out = 21), 4)

temp_vector_log = log(temp_vector)
precip_vector_log = log(precip_vector)
clay_vector_log = log(clay_vector)
depth_vector_log = log(depth_vector)

# Simulate landcover data.
land_vector <- c(rep("Forested Upland", 21),
  rep("Developed", 21),
  rep("Planted/Cultivated", 21),
  rep("Herbaceous Upland", 21))

# Join with continuous data.
data_pred <- data.frame(temp_vector_log, precip_vector_log, clay_vector_log, depth_vector_log, land_vector)
colnames(data_pred) <- c("temp_log", "precip_log", "clay_log", "depth_log", "LandCover1")

# Use the original model to predict outcomes.
```

```

prediction <- predict(C.mlm2,
newdata = data_pred,
type = "response",
se.fit = TRUE)

# Pull out predictions.
data_fig <- data.frame(data_pred,
prediction$fit,
prediction$se.fit)

# Rename columns.
colnames(data_fig) <- c("temp_log", "precip_log", "clay_log", "depth_log", "LandCover1", "probability",

# Graph the probabilities of getting a whole tail.
model_C_fig <- ggplot(data_fig, aes(x = depth_log,
y = probability)) +
  geom_line(aes(color = LandCover1)) +
  geom_ribbon(aes(ymin = probability - se,
ymax = probability + se,
fill = LandCover1), alpha = 0.3) +
  labs(title = "Effects of Land Cover Types on Soil Carbon Content",
x = "Log Average Depth", y = "Predicted Log C Content",
color = "LandCover1", fill = "LandCover1") +
  theme_bw()

ggsave("Carbon_model.png", plot = model_C_fig, width = 8, height = 6)

```