Assignment 5: Data Visualization

Sujay Dhanagare

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (use the tidy `NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv` version in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
NTL_LTER_PeterPaul <- read.csv(file = here("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_Pete
NIWOT_ridge_litter <- read.csv(file = here("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.cs

#2

class(NTL_LTER_PeterPaul$sampledate)
```

```
## [1] "factor"
```

```
class(NIWOT_ridge_litter$collectDate)
```

```
## [1] "factor"
```

```
NTL_LTER_PeterPaul$sampledate <- ymd(NTL_LTER_PeterPaul$sampledate)
NIWOT_ridge_litter$collectDate <- ymd(NIWOT_ridge_litter$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```
#3

library(ggplot2)

# Define a custom theme
my_theme <- theme_classic() +
  theme(
    line = element_line(
      color = '#000080',  # Navy blue for the Ashoka Chakra
```

```
    size = 2,
    linetype = 'solid'
  ),
  rect = element_rect(
    fill = 'white',  # White background as in the flag's middle band
    colour = 'black'
  ),
  text = element_text(
    face = 'plain',
    colour = '#000080',  # Navy blue text
    size = 16
  ),

  # Customize Plot Title
  plot.title = element_text(
    face = "bold",
    size = 20,
    color = "#FF9933",  # Saffron color for the plot title
    hjust = 0.5
  ),

  # Axis Titles are blank
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),

  # Customize Axis Ticks
  axis.ticks = element_line(
    color = "#138808"  # Green color for the ticks
  ),

  # Customize Major Grid Lines
  panel.grid.major = element_line(
    color = "#E5E5E5",
    size = 0.5
  ),

  # Remove Minor Grid Lines
  panel.grid.minor = element_blank(),

  # Customize Plot Background
  plot.background = element_rect(
    fill = "#FFFFFF",  # White background
    colour = NA
  ),

  # Customize Panel Background
  panel.background = element_rect(
    fill = "#FFFFFF",
    colour = NA
  ),

  # Customize Legend Key
  legend.key = element_rect(
```

```
      fill = "#FFFFFF",
      colour = "#FF9933"  # Saffron border for legend keys
    ),

    # Set Legend Position
    legend.position = "right",

    # Ensure theme completeness
    complete = TRUE
  )
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Set this theme as default
theme_set(my_theme)
```

Create graphs

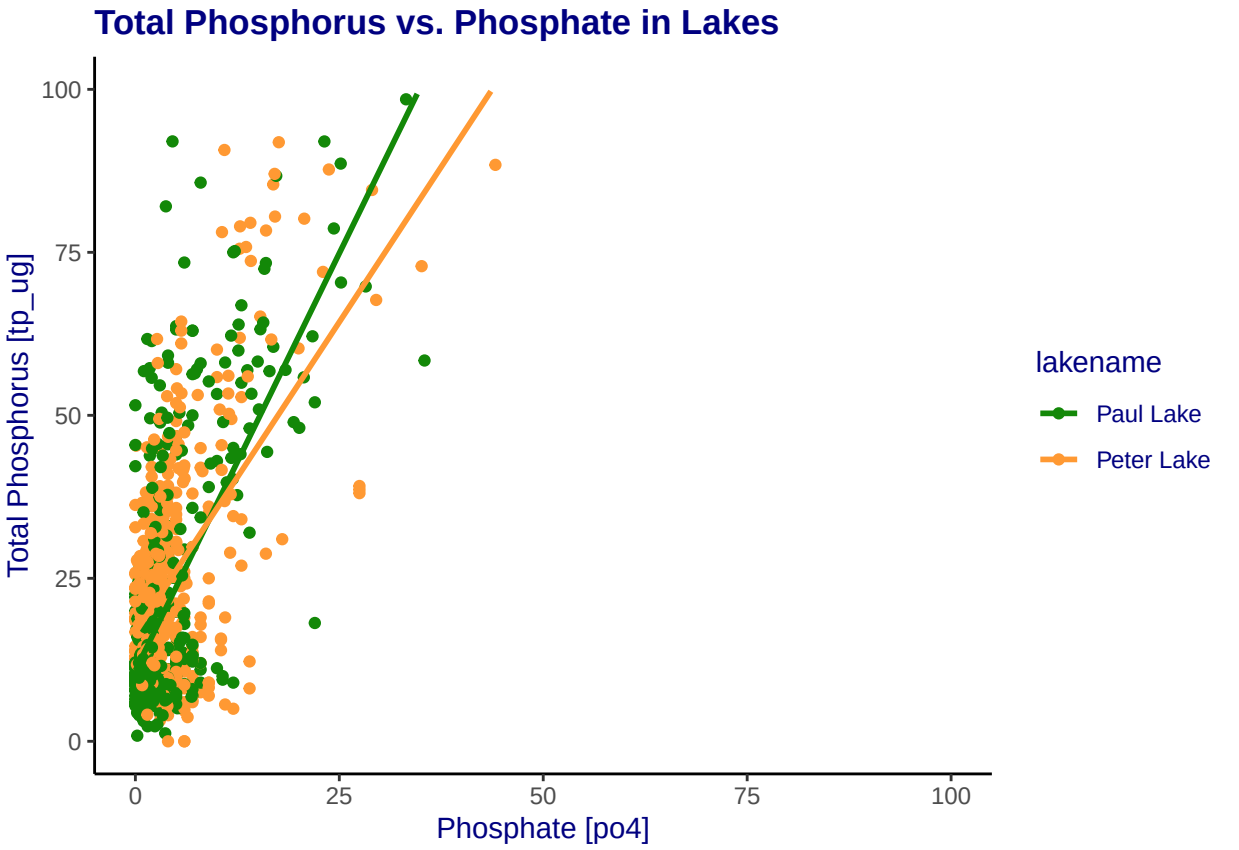
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

plot1 <- ggplot(NTL_LTER_PeterPaul, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +  # Add points
  geom_smooth(method = "lm", se = FALSE) +  # Add linear model lines without confidence intervals
  scale_color_manual(values = c("Peter Lake" = "#FF9933", "Paul Lake" = "#138808")) +
  labs(title = "Total Phosphorus vs. Phosphate in Lakes",
       x = "Phosphate [po4]",
       y = "Total Phosphorus [tp_ug]") +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#000080", face = "bold"),
    text = element_text(color = "#000080")
  ) +
  xlim(0.00, 100.00) +  # Limit x-axis to hide extreme values
  ylim(0, 100)       # Limit y-axis to hide extreme values

print(plot1)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 21964 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 21964 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_smooth()`).
```

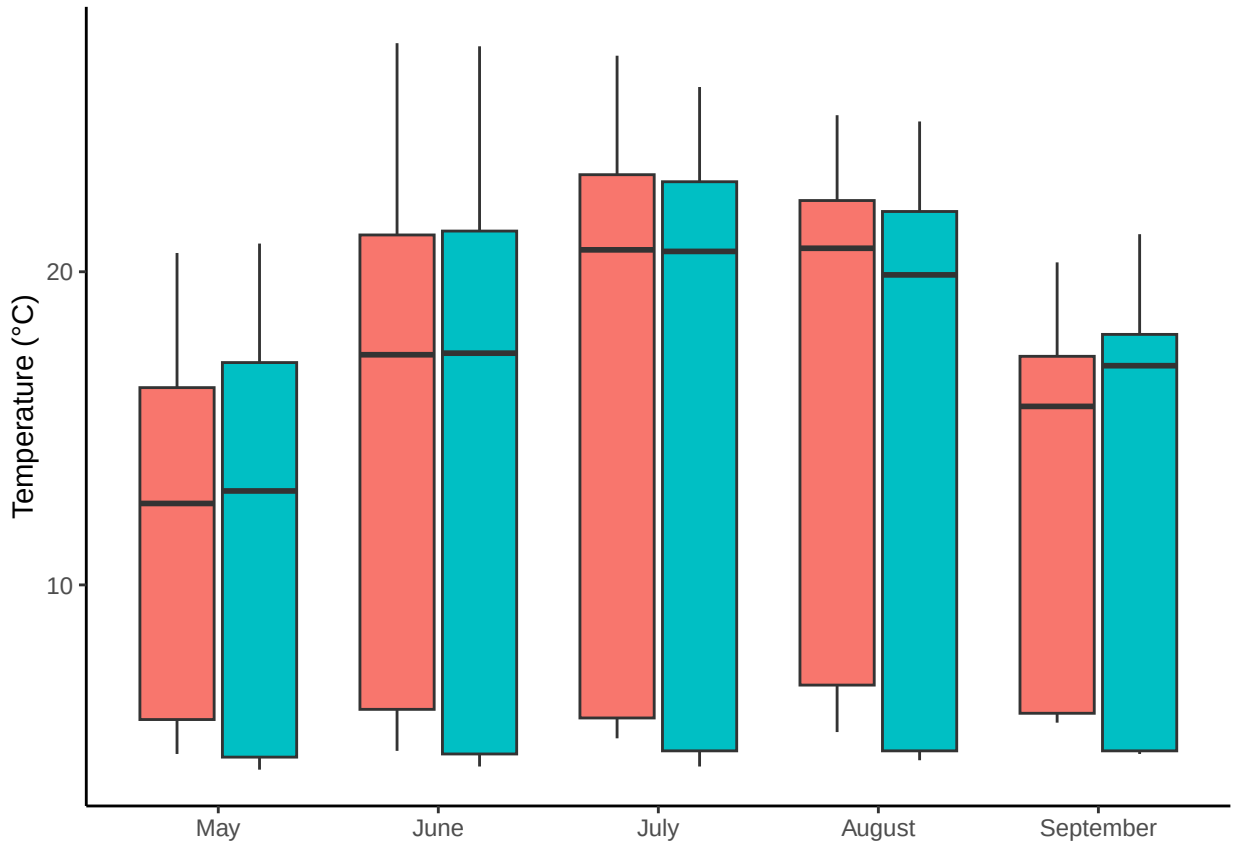


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

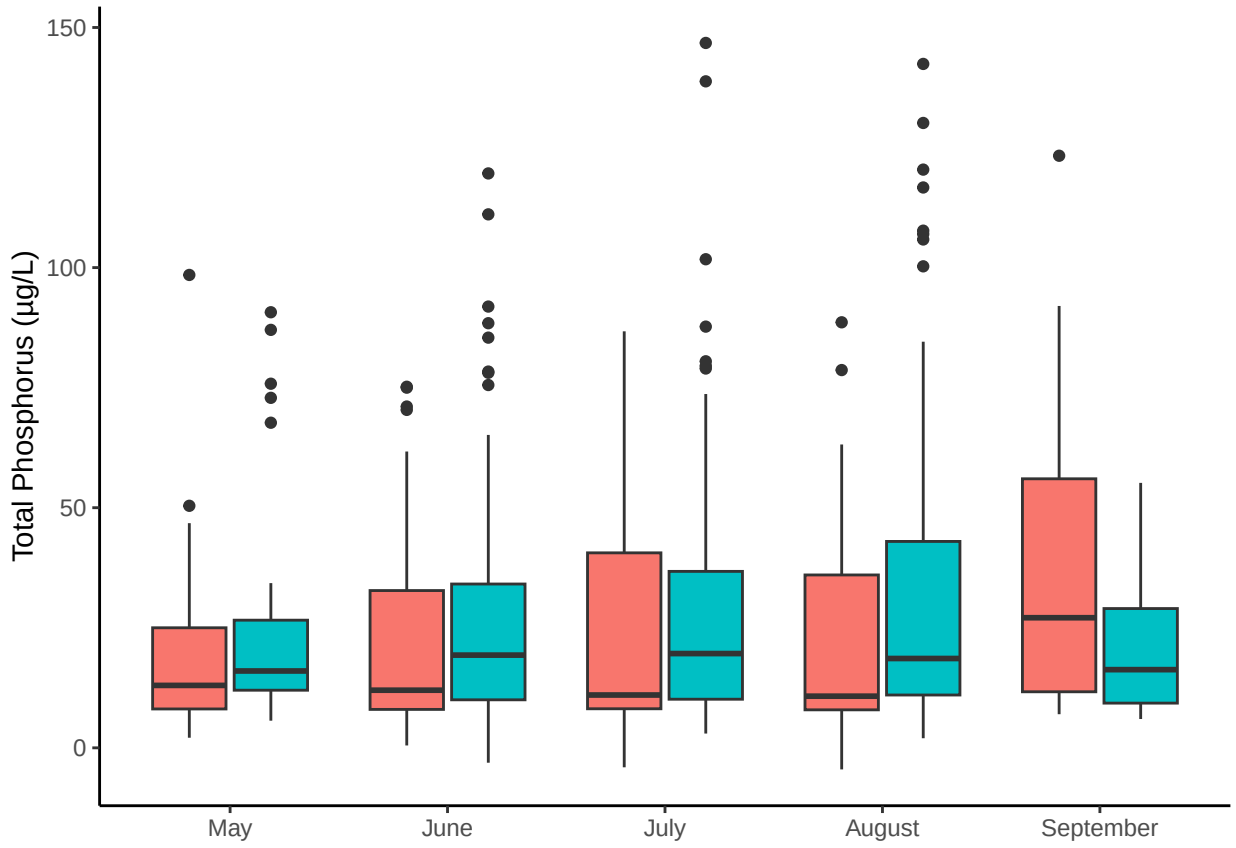
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5

NTL_LTER_PeterPaul$month <- factor(NTL_LTER_PeterPaul$month, levels = 1:12, labels = month.name)

NTL_LTER_PeterPaul_finite <- NTL_LTER_PeterPaul %>%
  filter(
    is.finite(temperature_C),
    is.finite(tp_ug),
    is.finite(tn_ug)
  )
```

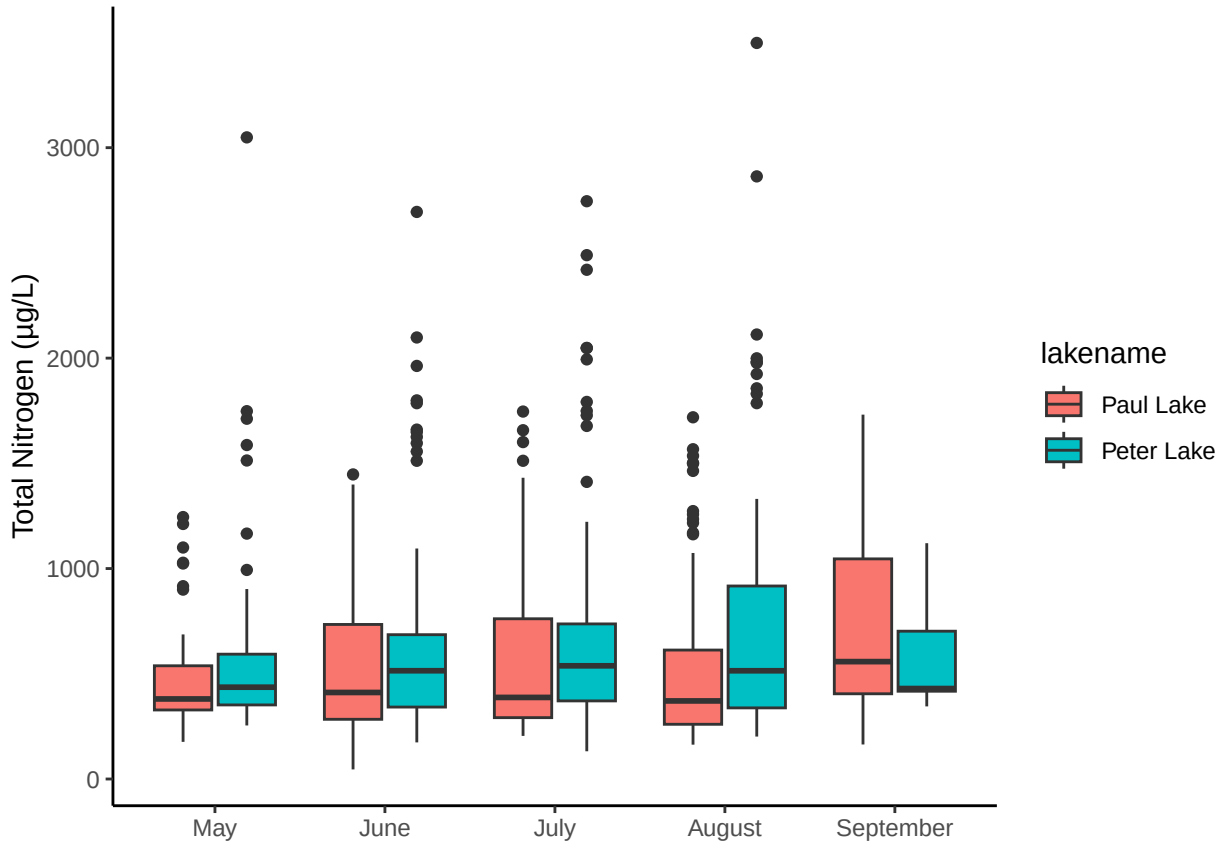
```r
# Plot 1: Temperature
plot_temp <- ggplot(NTL_LTER_PeterPaul_finite, aes(x = month, y = temperature_C, fill = lakename)) +
  geom_boxplot() +
  labs(y = "Temperature (°C)", x = NULL) +  # Remove x axis label for alignment
  theme_classic() +
  theme(axis.title.x = element_blank(),
        legend.position = "none")  # Remove legend

print(plot_temp)
```



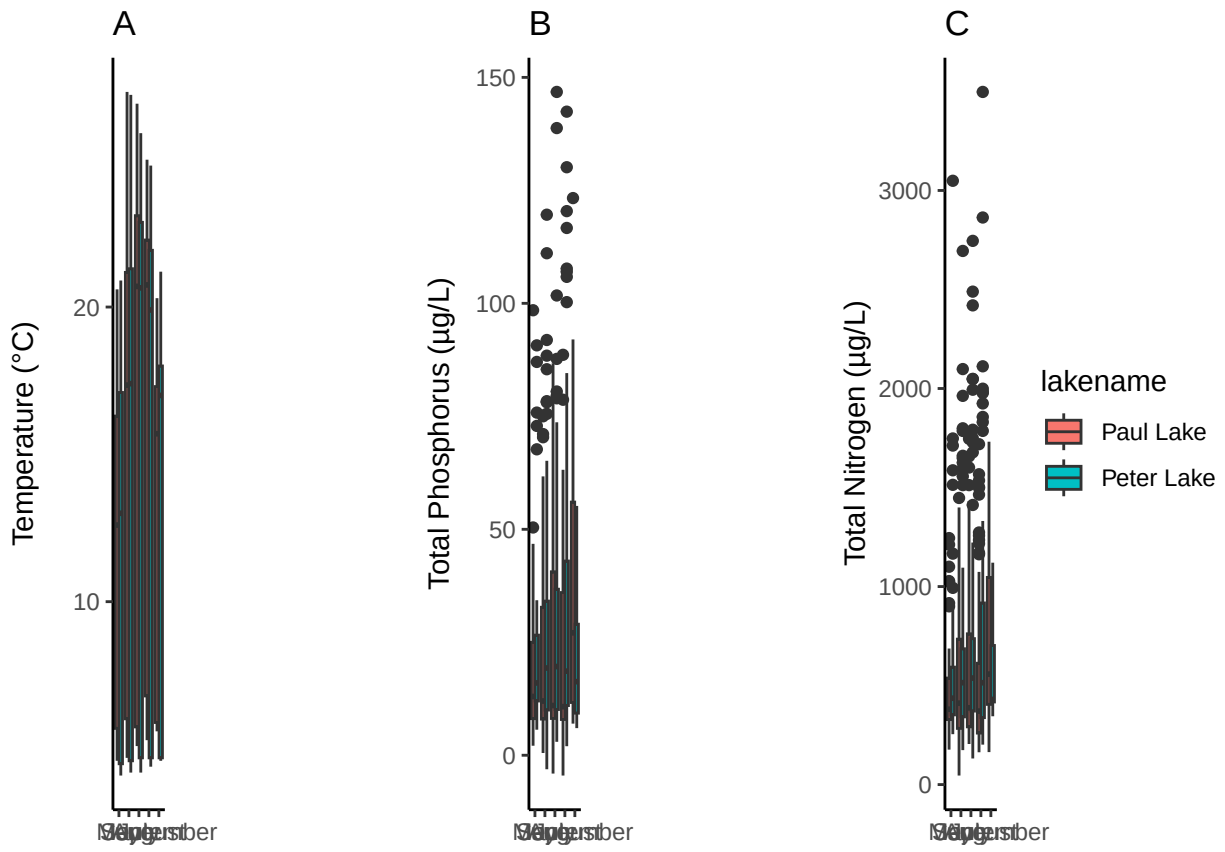

```r
# Plot 2: Total Phosphorus (TP)
plot_tp <- ggplot(NTL_LTER_PeterPaul_finite, aes(x = month, y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  labs(y = "Total Phosphorus (µg/L)", x = NULL) +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        legend.position = "none")

print(plot_tp)
```




```
# Plot 3: Total Nitrogen (TN)
plot_tn <- ggplot(NTL_LTER_PeterPaul_finite, aes(x = month, y = tn_ug, fill = lakename)) +
  geom_boxplot() +
  labs(y = "Total Nitrogen (µg/L)", x = "Month") +
  theme_classic() +
  theme(axis.title.x = element_blank())

print(plot_tn)
```



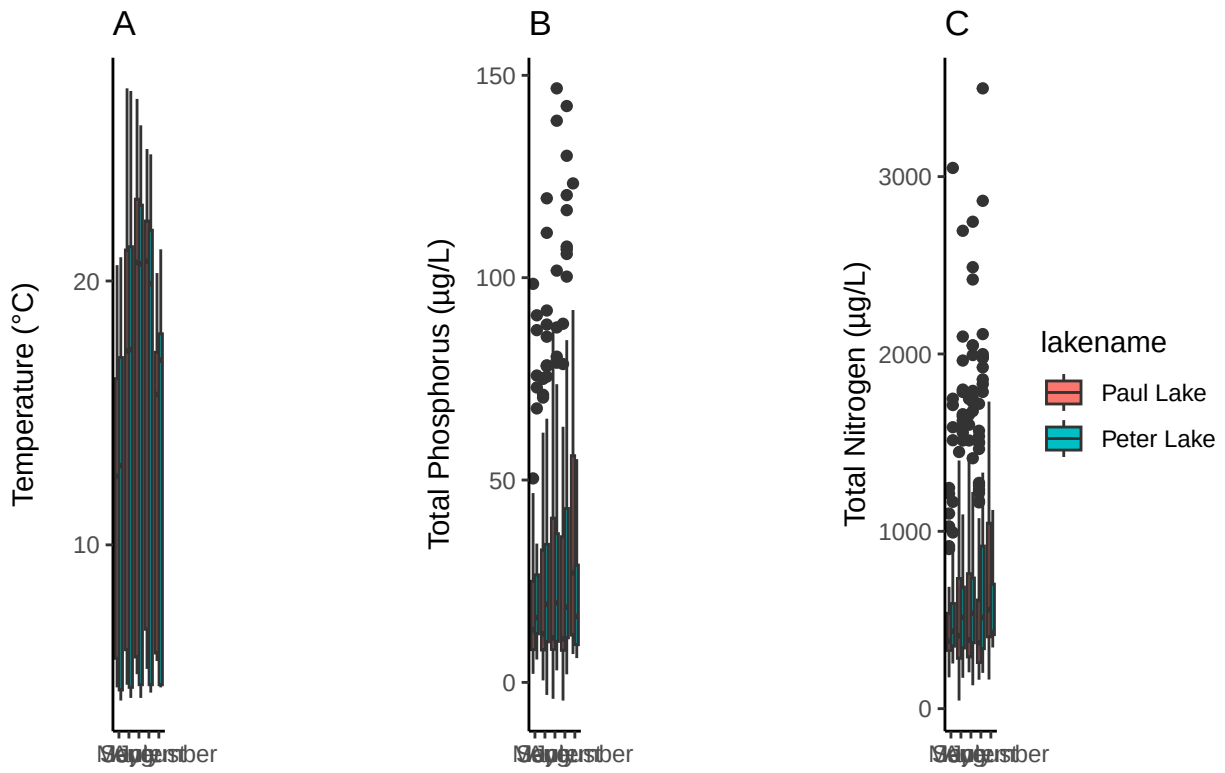

```
# Combine the plots using cowplot
plot_temp <- plot_temp + ggtitle("A")
plot_tp <- plot_tp + ggtitle("B")
plot_tn <- plot_tn + ggtitle("C")
combined_plot <- plot_grid(plot_temp, plot_tp, plot_tn, nrow = 1, align = 'v')

print(combined_plot)
```




```
# Adding a common legend
legend <- get_legend(ggplot(NTL_LTER_PeterPaul_finite, aes(x = month, y = temperature_C, fill = lakenam
                     geom_boxplot() +
                     theme(legend.position = "bottom"))
```

```
## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use 'return_all = TRUE'.
```

```
final_plot <- plot_grid(combined_plot, legend, ncol = 1, rel_heights = c(1, 0.1))

# Print the final combined plot
print(final_plot)
```



Question: What do you observe about the variables of interest over seasons and between lakes?

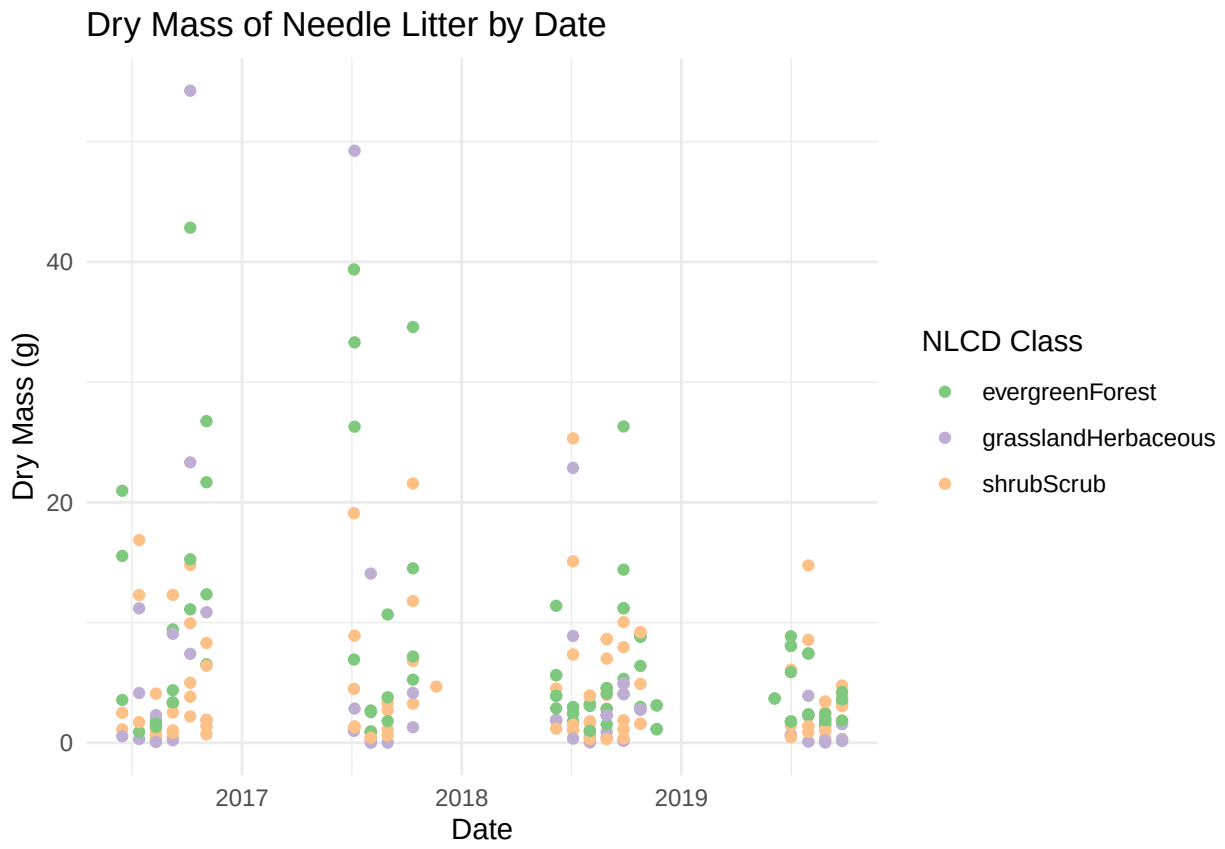
> Answer: The analysis of the plots reveals that neither Paul Lake nor Peter Lake consistently exhibits higher concentrations of total phosphorus or total nitrogen throughout the months from May to September. Both lakes show variability in nutrient levels, reflecting potentially diverse biogeochemical processes, watershed inputs, and management practices affecting each lake differently. Peter Lake does show higher levels from May to August but its levels are lower than Paul Lake in September.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

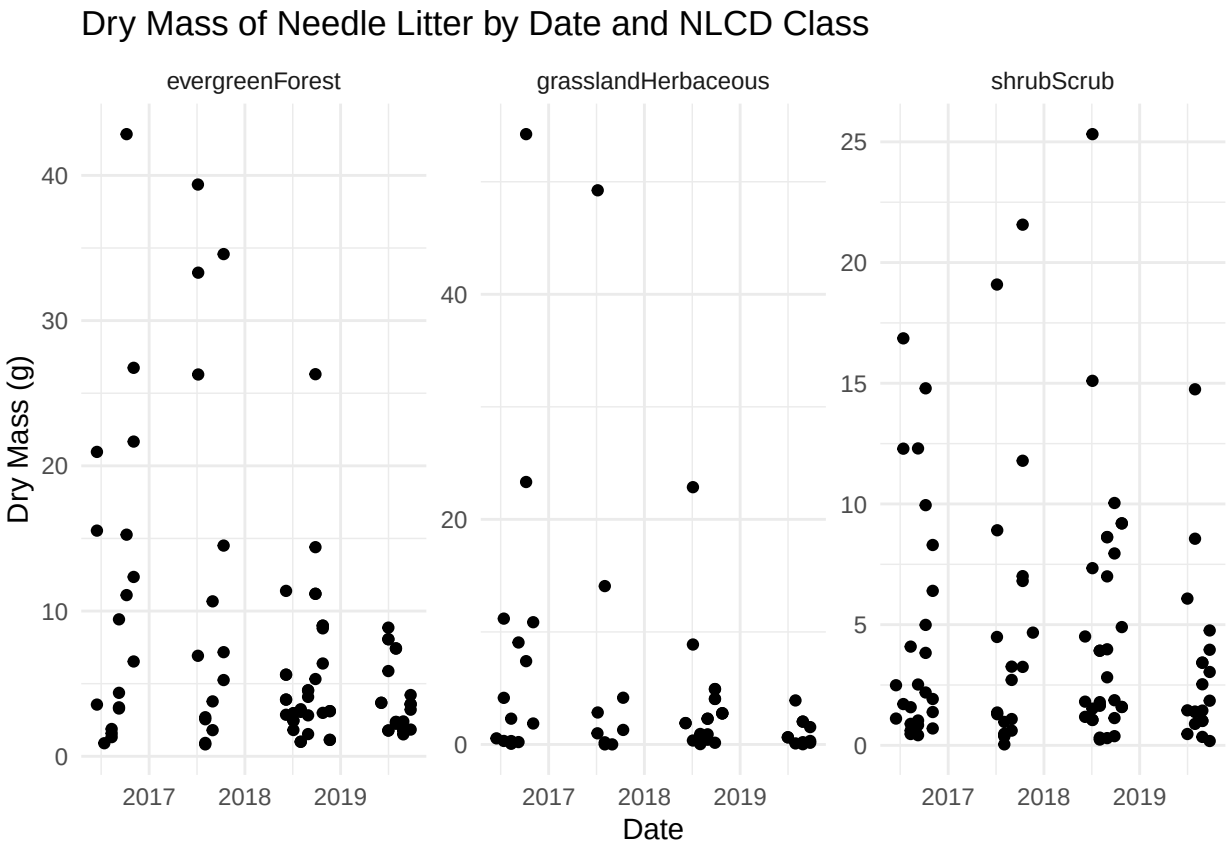
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
needles_data <- NIWOT_ridge_litter %>%
  filter(functionalGroup == "Needles")

plot_needles <- ggplot(needles_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by Date",
       x = "Date",
       y = "Dry Mass (g)",
       color = "NLCD Class") +
```

```
  theme_minimal() +
  scale_color_brewer(type = 'qual')  # using a qualitative color palette for clarity

print(plot_needles)
```




```
#7
plot_needles_facets <- ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by Date and NLCD Class",
       x = "Date",
       y = "Dry Mass (g)") +
  theme_minimal() +
  facet_wrap(~nlcdClass , scales = "free_y")  # Facet by NLCD class with free y-axis scales

# Print the plot
print(plot_needles_facets)
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer:Plot 7 is more effective than Plot 6 because it separates the data into different panels for each land cover type. This layout makes it easier to see and compare the specific trends and unusual data points within each category, without the confusion that overlapping data can cause in Plot 6. This clear separation in Plot 7 helps in understanding and analyzing the data better for each land type.