Assignment 5: Data Visualization

Lauren Xu

Spring 2025

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
#Load libraries
library(tidyverse);library(lubridate);library(here);library(cowplot)
```

```
## -- 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 error
## here() starts at /home/guest/EDE_Spring2025
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
#Assign a variable to the processed data folder location
processed_data = "./Data/Processed_KEY"

#Read in data(1)
PeterPaul.chem.nutrients <- read.csv(
  here(processed_data,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)
#Read in data(2)
Litter.mass.trap.1 <- read.csv(
  here(processed_data,"NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
#Convert date columns to be date objects
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate, format="%Y/%m/%d")

Litter.mass.trap.1$collectDate <- as.Date(as.character(Litter.mass.trap.1$collectDate), format="%Y-%m-%
```

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
# Define a custom theme
my_theme <- theme(
  plot.background = element_rect(fill = "white", color = NA),
  plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
  axis.title = element_text(face = "italic", size = 12),
  axis.text = element_text(size = 10, color = "black"),

)
```

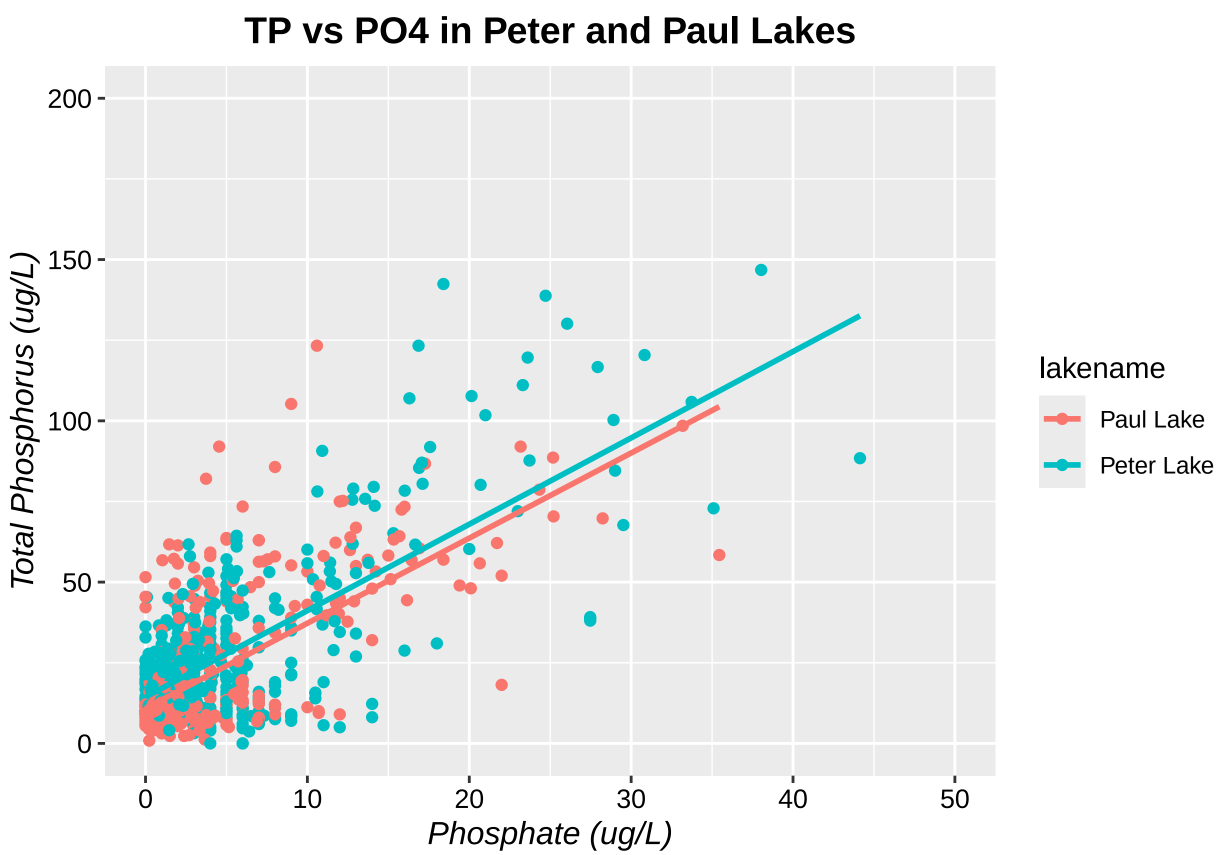
Create graphs

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
library(ggplot2)

# Create the plot
ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  xlim(0, 50) +
  ylim(0, 200) +
  labs(x = "Phosphate (ug/L)", y = "Total Phosphorus (ug/L)",
       title = "TP vs PO4 in Peter and Paul Lakes") +
  my_theme
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21948 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 21948 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



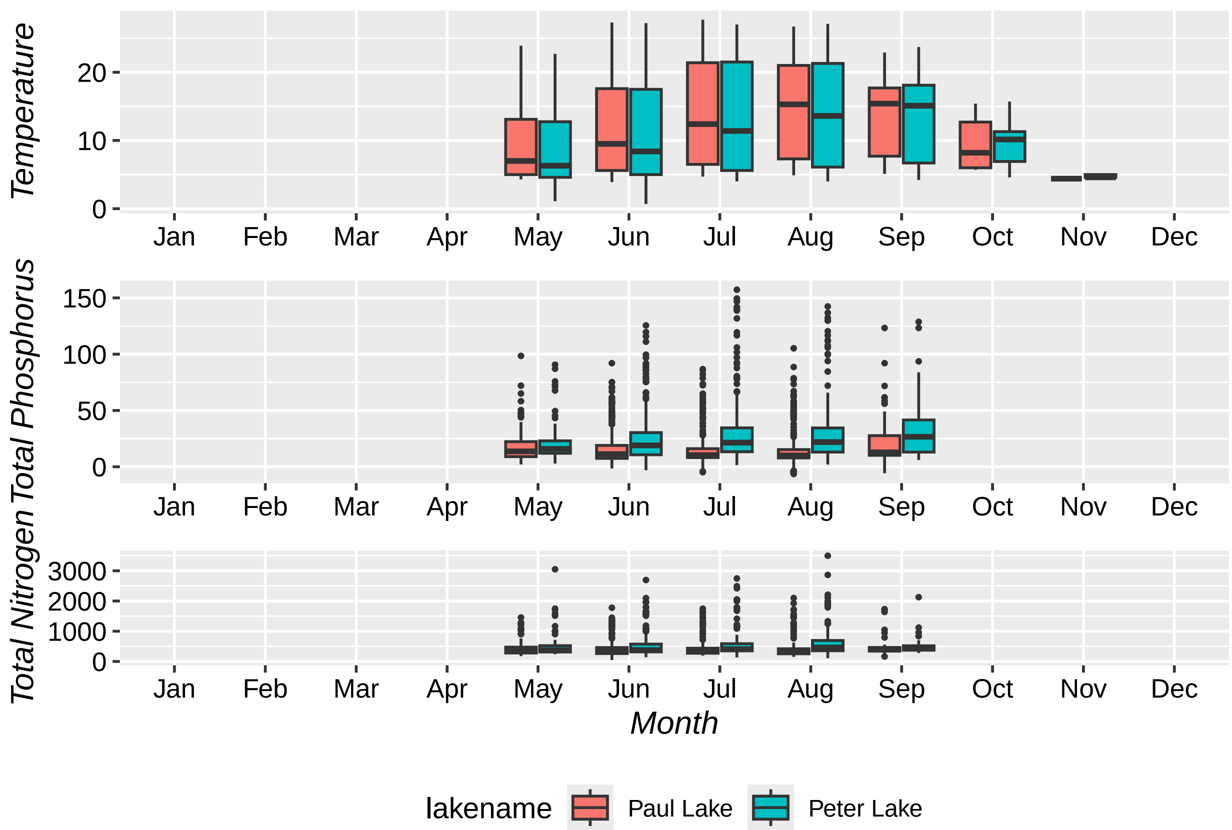
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

```r
#5
# Ensure month is treated as a factor for proper ordering
PeterPaul.chem.nutrients$month <-
  factor(PeterPaul.chem.nutrients$month, levels = 1:12, labels = month.abb)

# Boxplot for Temperature (a)
plot_a <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C, fill = lakename)) +
        scale_x_discrete(drop=F) +
  geom_boxplot(outlier.size = 0.5) +
  labs(y = "Temperature", x = NULL) +
  my_theme +
  theme(
    legend.position = "none",
    axis.title.x = element_blank()
  )

# Boxplot for Total Phosphorus (b)
plot_b <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug, fill = lakename)) +
        scale_x_discrete(drop=F) +
  geom_boxplot(outlier.size = 0.5) +
  labs(y = "Total Phosphorus", x = NULL) +
  my_theme +
  theme(
    legend.position = "none",
    axis.title.x = element_blank()
  )

# Boxplot for Total Nitrogen (c)
plot_c <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug, fill = lakename)) +
        scale_x_discrete(drop=F) +
  geom_boxplot(outlier.size = 0.5) +
  labs(y = "Total Nitrogen", x = "Month") +
  my_theme +
  theme(legend.position = "bottom")

# Combine plots using cowplot
final_plot <- plot_grid(
  plot_a, plot_b, plot_c,
  align = "v",
  ncol = 1,
  rel_heights = c(1, 1, 1.2)
)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

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

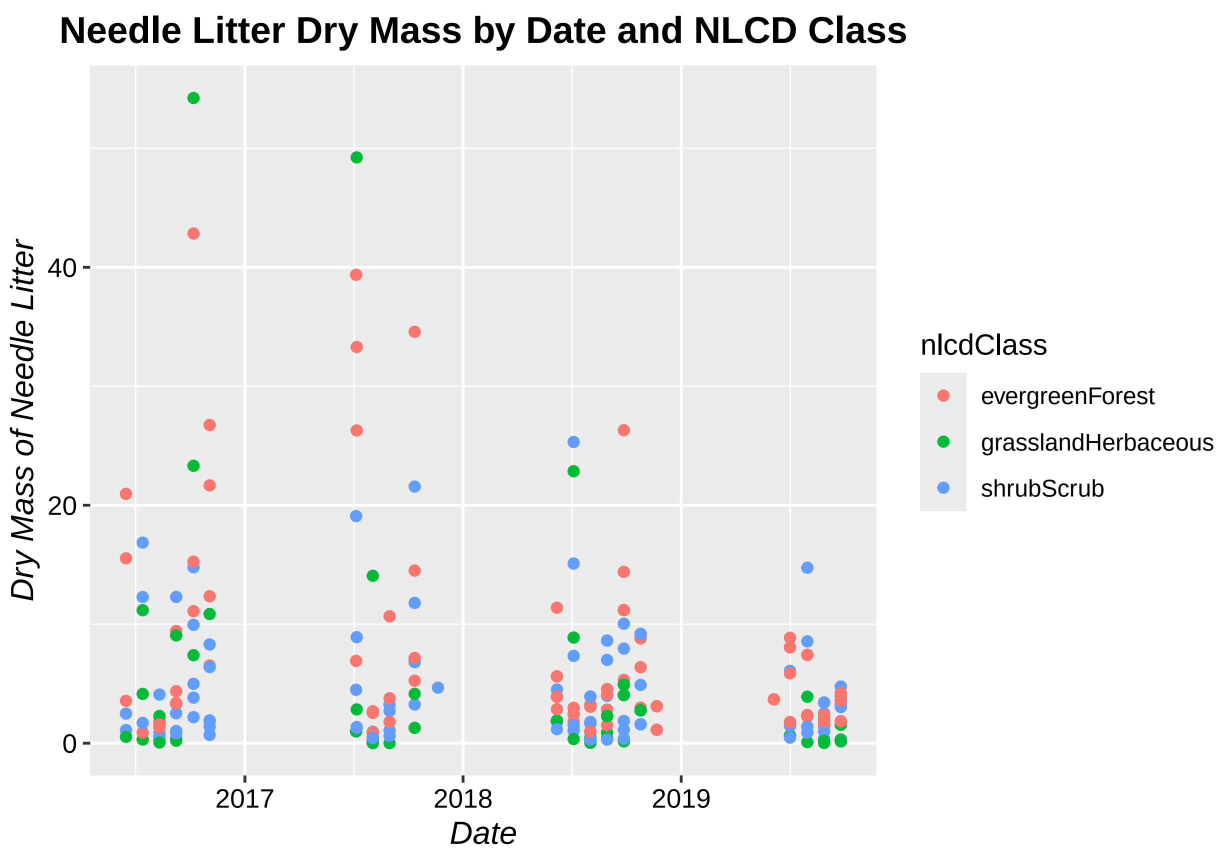


Question: What do you observe about the variables of interest over seasons and between lakes?

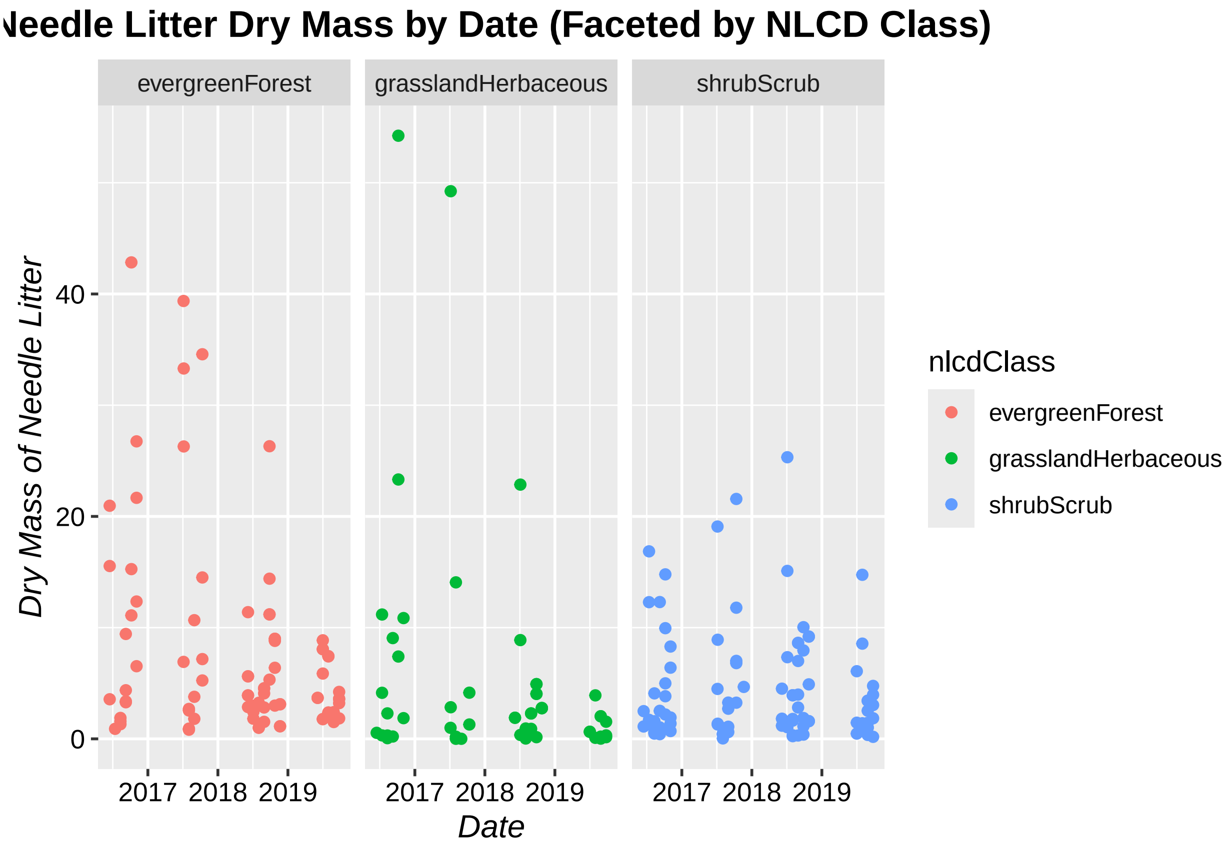
> Answer: Temperature rises from May to August and decreasing afterward, with similar trends in both lakes. Total Phosphorus (TP) and Total Nitrogen (TN) show higher variability in summer months. Peter Lake generally exhibits higher nutrient concentrations than Paul Lake.

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.

```r
#6
# Filter for the "Needles" functional group
needle_data <- subset(Litter.mass.trap.1, functionalGroup == "Needles" &
                      !is.na(collectDate) & !is.na(dryMass) & dryMass > 0)

# Plot dry mass of needle litter by date, separated by NLCD class with color aesthetic
Plot_Ex6 <-
  ggplot(needle_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(x = "Date", y = "Dry Mass of Needle Litter", title = "Needle Litter Dry Mass by Date and NLCD Cla
  my_theme

print(Plot_Ex6)
```




```r
#7
# Plot the same data with NLCD classes separated into three facets
Plot_Ex7 <-
ggplot(needle_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  facet_wrap(~ nlcdClass) +
  labs(x = "Date", y = "Dry Mass of Needle Litter",
       title = "Needle Litter Dry Mass by Date (Faceted by NLCD Class)") +
  my_theme

print(Plot_Ex7)
```



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

Answer: Plot 7 is likely more effective. Because by having separate panels for each NLCD class, we can easily see the data trends for each class without any overlap. However, Plot 6 can provide a more compact and comparative view of the overall trends.