Assignment 5: Data Visualization

Emily Kuhlmann

Fall 2023

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) ; library(lubridate) ; library(here) ; library(cowplot)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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
```

```
## here() starts at /Users/emilykuhlmann/Documents/Documents/Fall23/EDA/EDE_Fall2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
getwd()
```

```
## [1] "/Users/emilykuhlmann/Documents/Documents/Fall23/EDA/EDE_Fall2023"
```

```
PeterPaul.chem.nutrients <-
  read.csv(
    here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
Niwot.Litter <- read.csv(
  here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
            stringsAsFactors = T)
#2
PeterPaul.date <- class(PeterPaul.chem.nutrients$sampledate)
PeterPaul.date
```

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

```
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)

Niwot.date <- class(Niwot.Litter$collectDate)
Niwot.date
```

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

```
Niwot.Litter$collectDate <- ymd(Niwot.Litter$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
my_theme <- theme_classic()  +
  theme(axis.text = element_text(color = "black"),
        plot.title = element_text(color = "darkblue"), legend.position = 'top')
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.

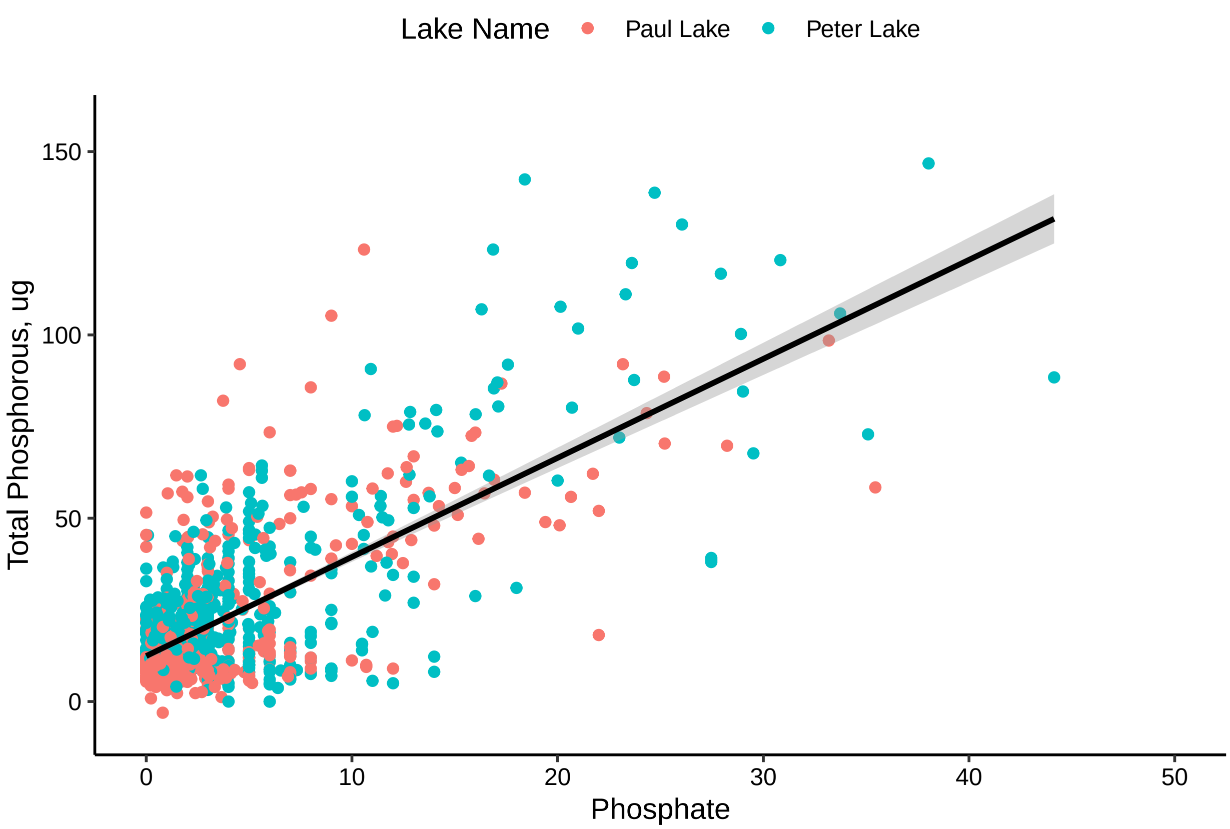
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Plot1 <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  labs(color = "Lake Name") +
  scale_y_continuous(name = "Total Phosphorous, ug") +
  scale_x_continuous(name = "Phosphate", limits = c(0, 50))
Plot1
```

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

## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21947 rows containing missing values (`geom_point()`).
```

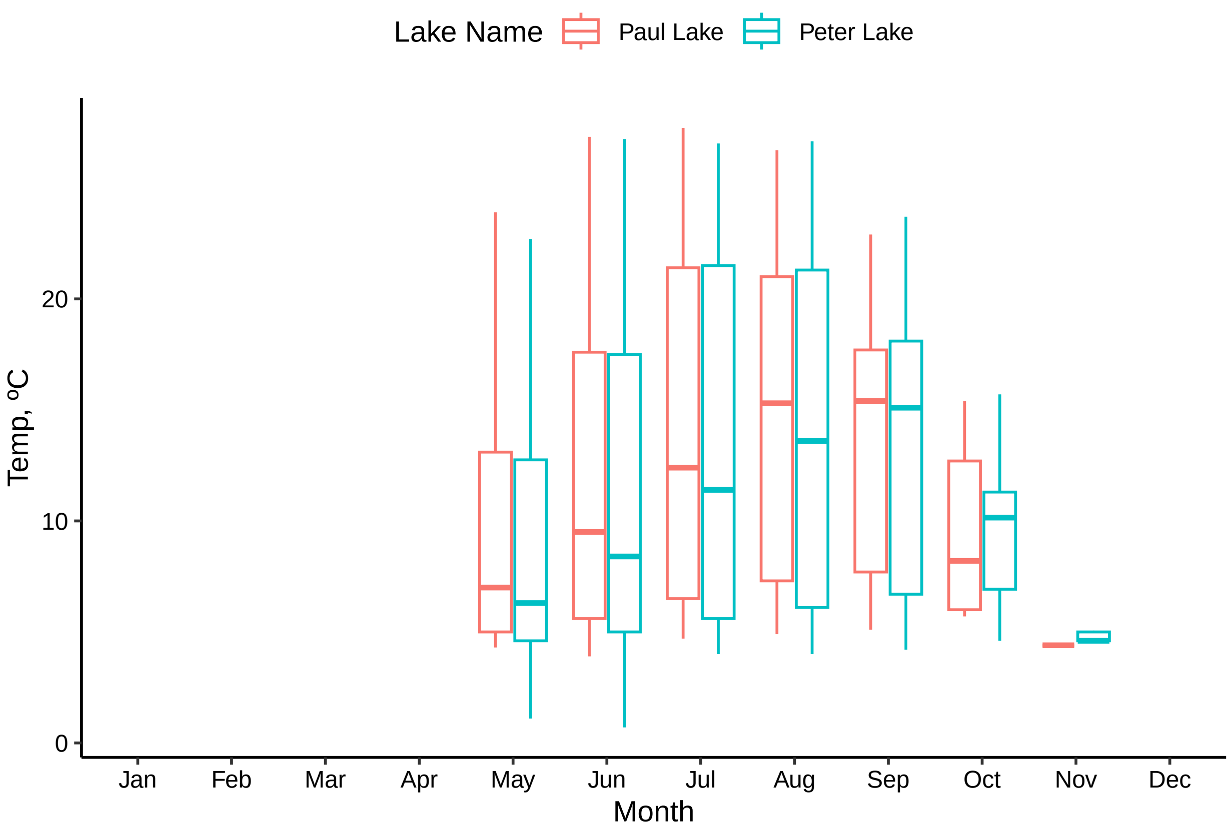


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

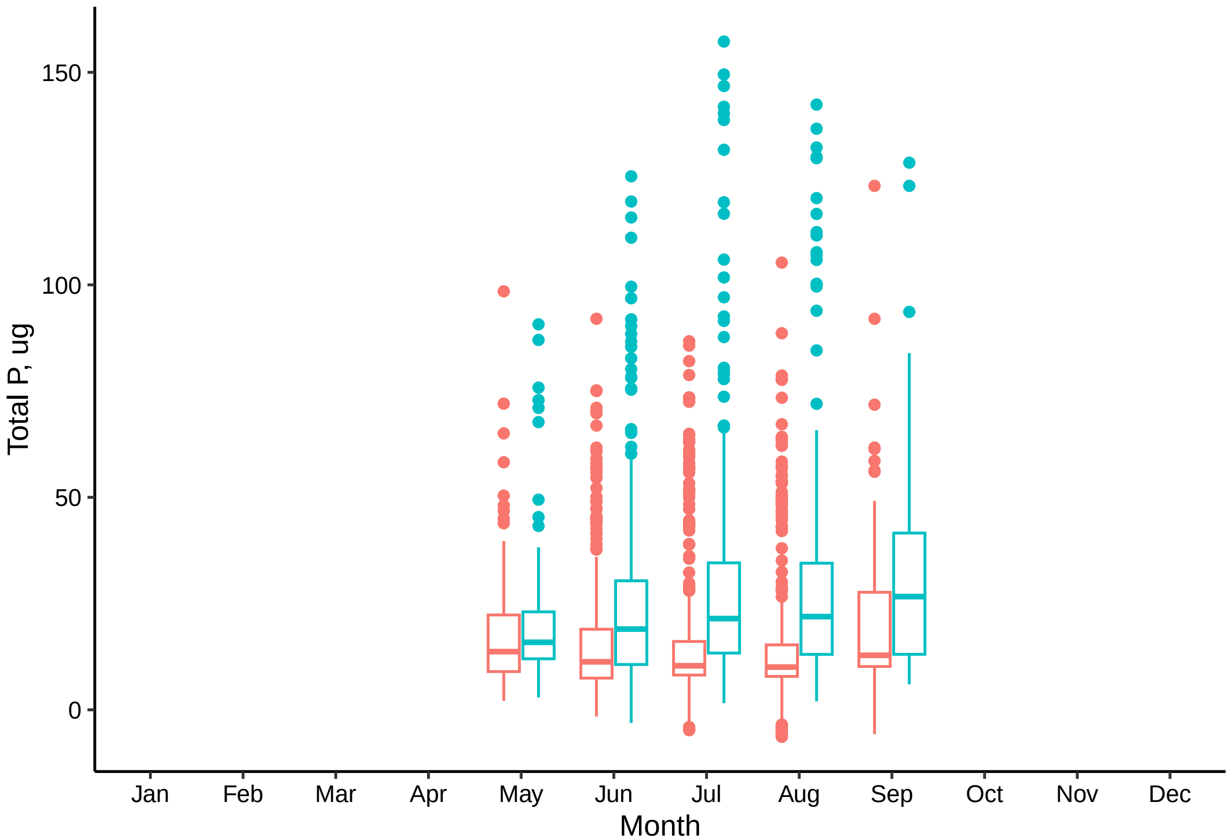
```
#5
Temp.Boxplot <- ggplot(PeterPaul.chem.nutrients,
                       aes(x=factor(month, levels = 1:12, labels = month.abb),
                           y = temperature_C, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop=F) +
  labs(color = "Lake Name") +
  scale_y_continuous(name = "Temp, ºC")
Temp.Boxplot
```

```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```



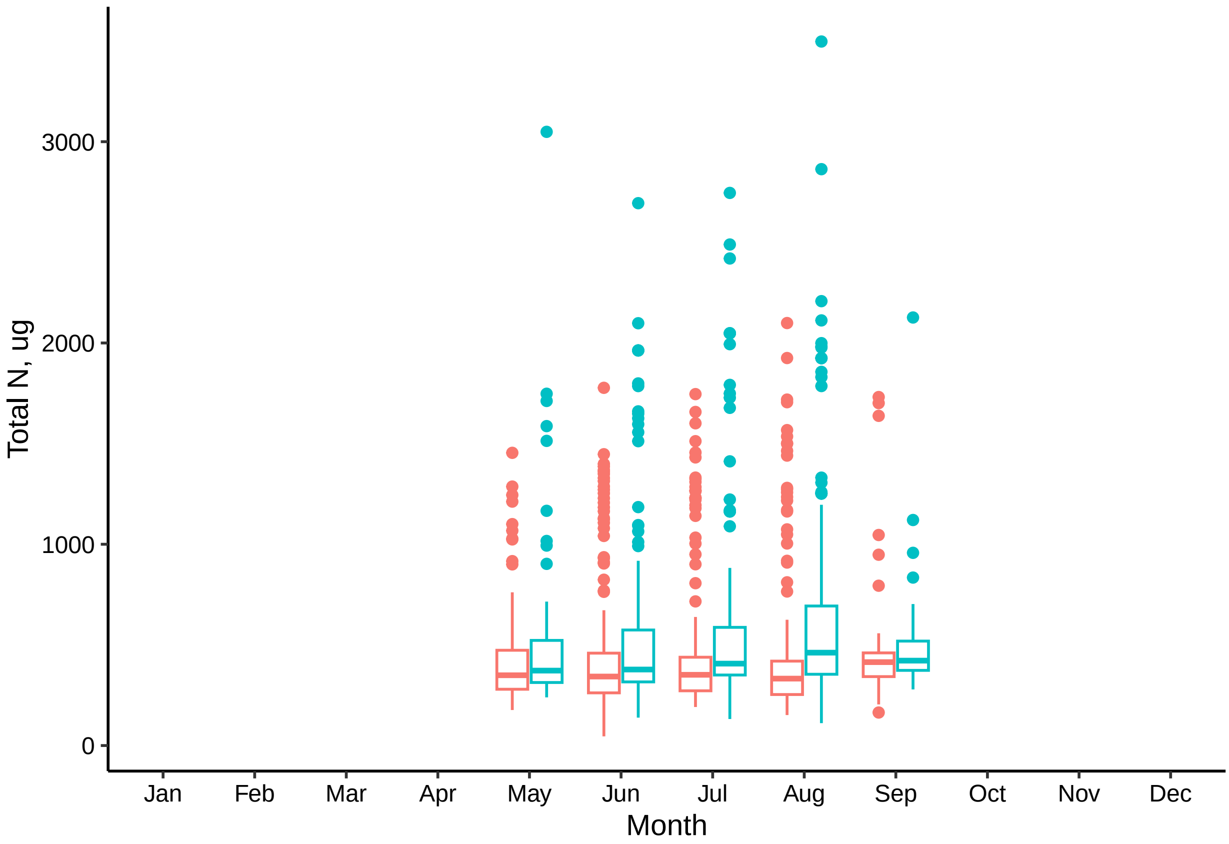

```
TP.Boxplot <- ggplot(PeterPaul.chem.nutrients,
                     aes(x=factor(month, levels = 1:12, labels = month.abb),
                         y = tp_ug, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop=F) +
  theme(legend.position = "none") +
  scale_y_continuous(name = "Total P, ug")
TP.Boxplot
```

```
## Warning: Removed 20729 rows containing non-finite values ('stat_boxplot()').
```




```r
TN.Boxplot <- ggplot(PeterPaul.chem.nutrients,
                     aes(x=factor(month, levels = 1:12, labels = month.abb),
                         y = tn_ug, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop=F) +
  theme(legend.position = "none") +
  scale_y_continuous(name = "Total N, ug")
TN.Boxplot
```

```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```



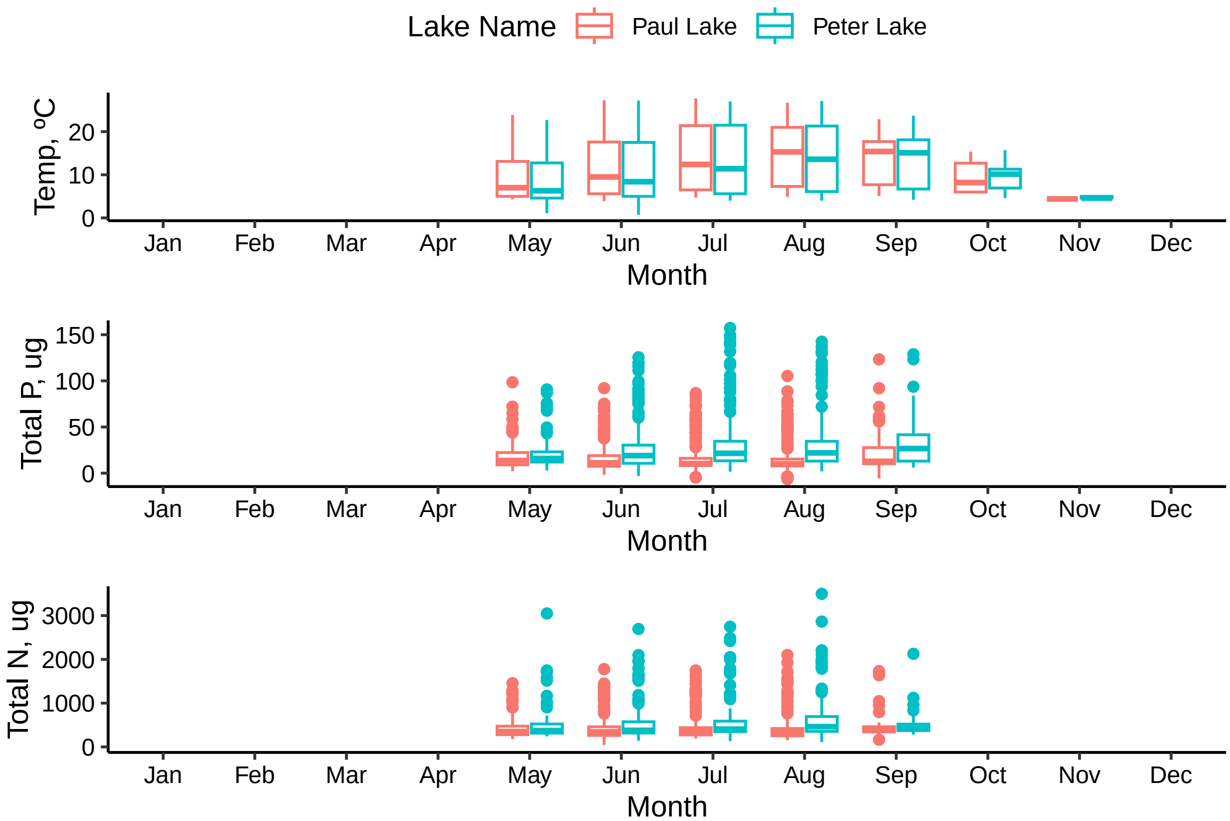

```
PeterPaul.Temp.TP.TN <- plot_grid(Temp.Boxplot, TP.Boxplot, TN.Boxplot,
                                  nrow = 3, align = "hv", rel_heights = c(1.25, 1, 1),
                                  axis = "l")
```

```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

```
PeterPaul.Temp.TP.TN
```

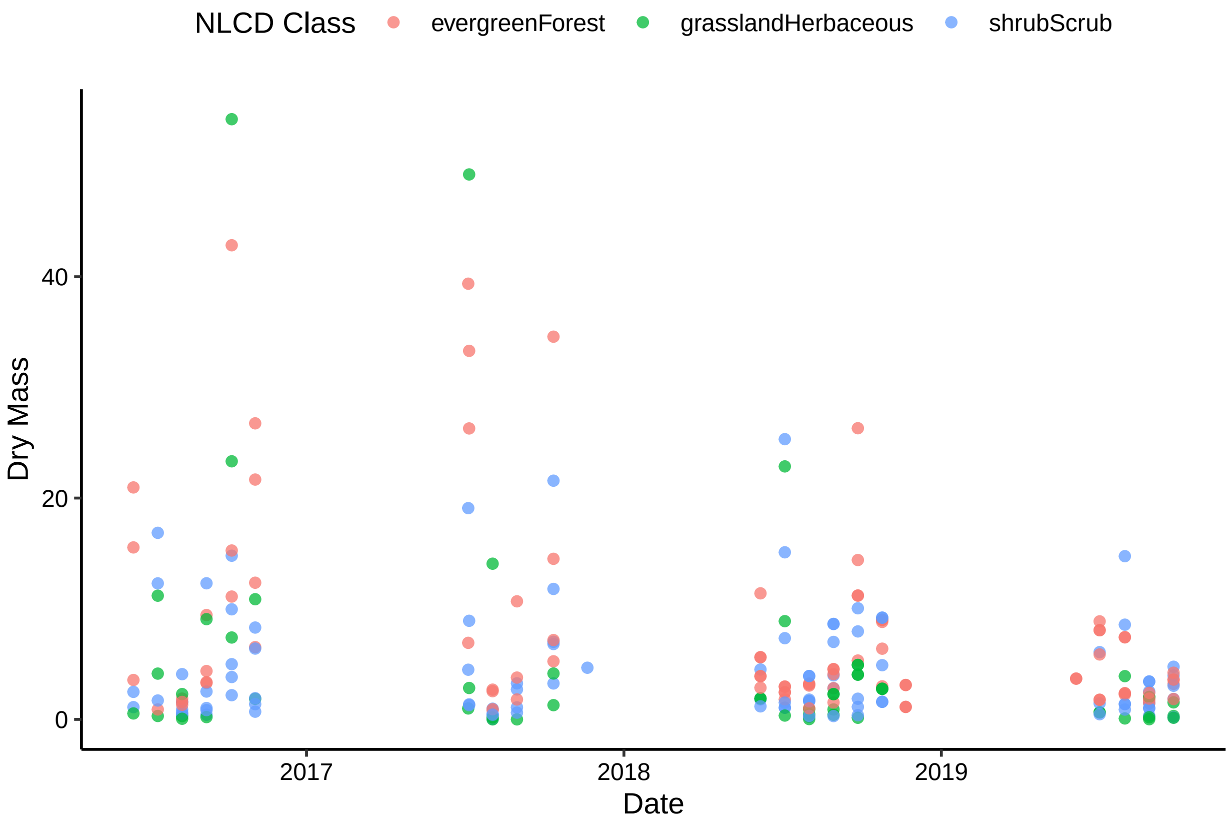


Question: What do you observe about the variables of interest over seasons and between lakes?

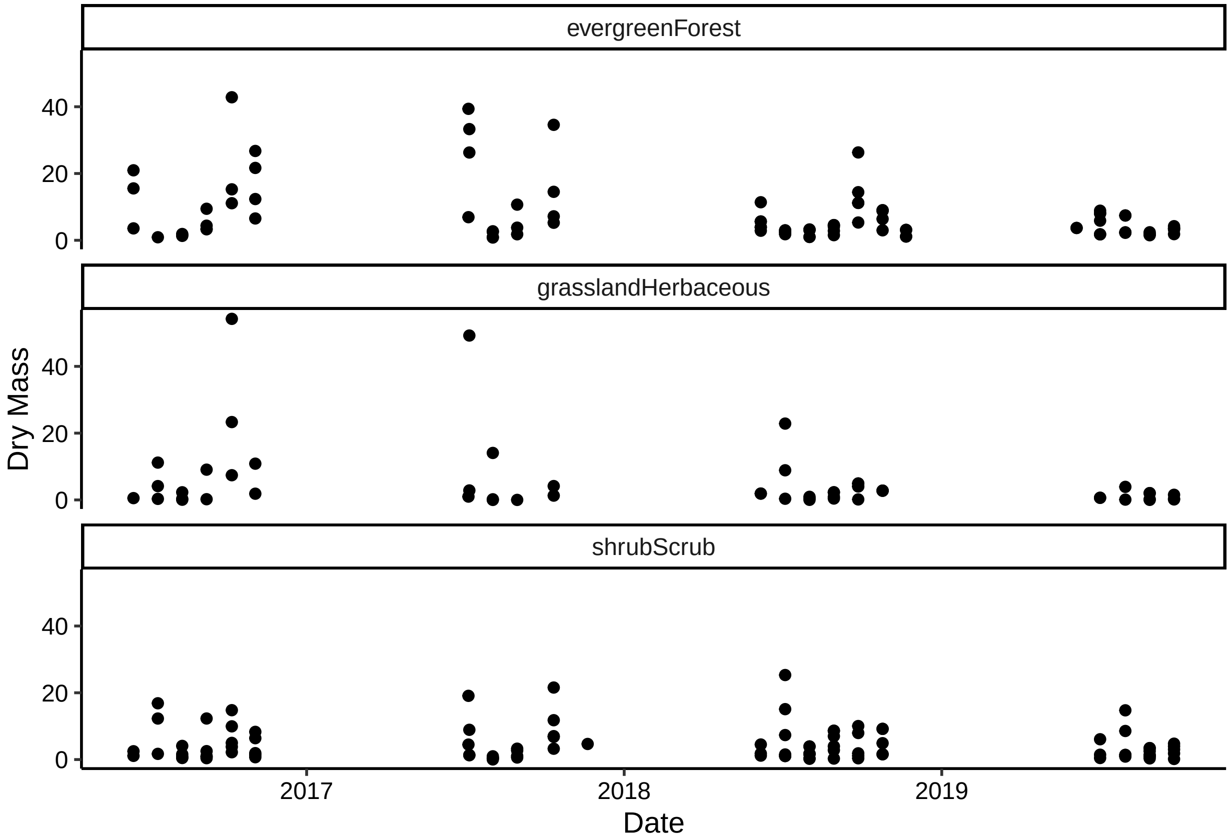
> Answer: The temperature of both lakes rises throughout the summer and then decreases beginning in October, following the seasons. The temperatures of both lakes are very similar. The phosphorous measurements in Paul Lake remain more consistent throughout the summer than in Peter Lake. The Peter Lake phosphorous measurements increase from May to September and are mostly higher than in Paul Lake. The nitrogen measurements in both lakes do not seem to change much with the seasons where samples were collected as they remain fairly consistent from May through September. Peter Lake has slightly higher nitrogen measurements and more high outliers 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.

```
#6
Niwot.needle.mass <- ggplot(Niwot.Litter %>% filter(functionalGroup == "Needles"),
                            aes(x = collectDate, y = dryMass,
                                color = nlcdClass)) +
  geom_point(alpha=0.75) +
  scale_x_date(name = "Date") +
  scale_y_continuous(name = "Dry Mass") +
  labs(color = "NLCD Class")
Niwot.needle.mass
```




```
#7
Niwot.needle.mass2 <- ggplot(Niwot.Litter %>% filter(functionalGroup == "Needles"),
                              aes(x = collectDate, y = dryMass)) +
  geom_point() +
  scale_x_date(name = "Date") +
  scale_y_continuous(name = "Dry Mass") +
  facet_wrap(vars(nlcdClass), nrow = 3)
Niwot.needle.mass2
```



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

> Answer: I think that plot 7 is more effective for these data becasuse it is easier to view the points from each NLCD class since there is a lot of overlap at the lower masses. The facets allow you to more easily see the trends of each NLCD class over time and compare them to the other classes.