Assignment 5: Data Visualization

Ye Khaung Oo

Spring 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); library(lubridate); library(here); library(cowplot); library(ggplot2); library(ggth
```

```
## -- 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   3.4.4     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 /Users/yekhaungoo/Library/CloudStorage/OneDrive-Personal/EDA Class
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
##
##
## Attaching package: 'ggthemes'
##
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
getwd()
```

```
## [1] "/Users/yekhaungoo/Library/CloudStorage/OneDrive-Personal/EDA Class"
```

```
peter_paul_df <- read.csv("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.cs
litter_df <- read.csv("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors

#2
class(peter_paul_df$sampledate)
```

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

```
class(litter_df$collectDate)
```

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

```
peter_paul_df$sampledate <- as.Date((peter_paul_df$sampledate), format = "%Y-%m-%d")
litter_df$collectDate <- as.Date((litter_df$collectDate), format = "%Y-%m-%d")
```

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
yko_theme <- theme_economist(base_size = 7) +
  theme(axis.text = element_text(color = "blue"),
        axis.text.x = element_text(color = "darkblue", size = 8),
        axis.text.y = element_text(color = "darkblue", size = 8),
        plot.title =        element_text(color = "red", size = 9),
        axis.ticks =        element_line(6),
        legend.position = "right",
        legend.direction = "vertical",
        legend.background = element_rect(color='lightyellow'
    ),
    legend.title = element_text(
      color="darkgrey", size = 8
    )
  )

theme_set(yko_theme)
```

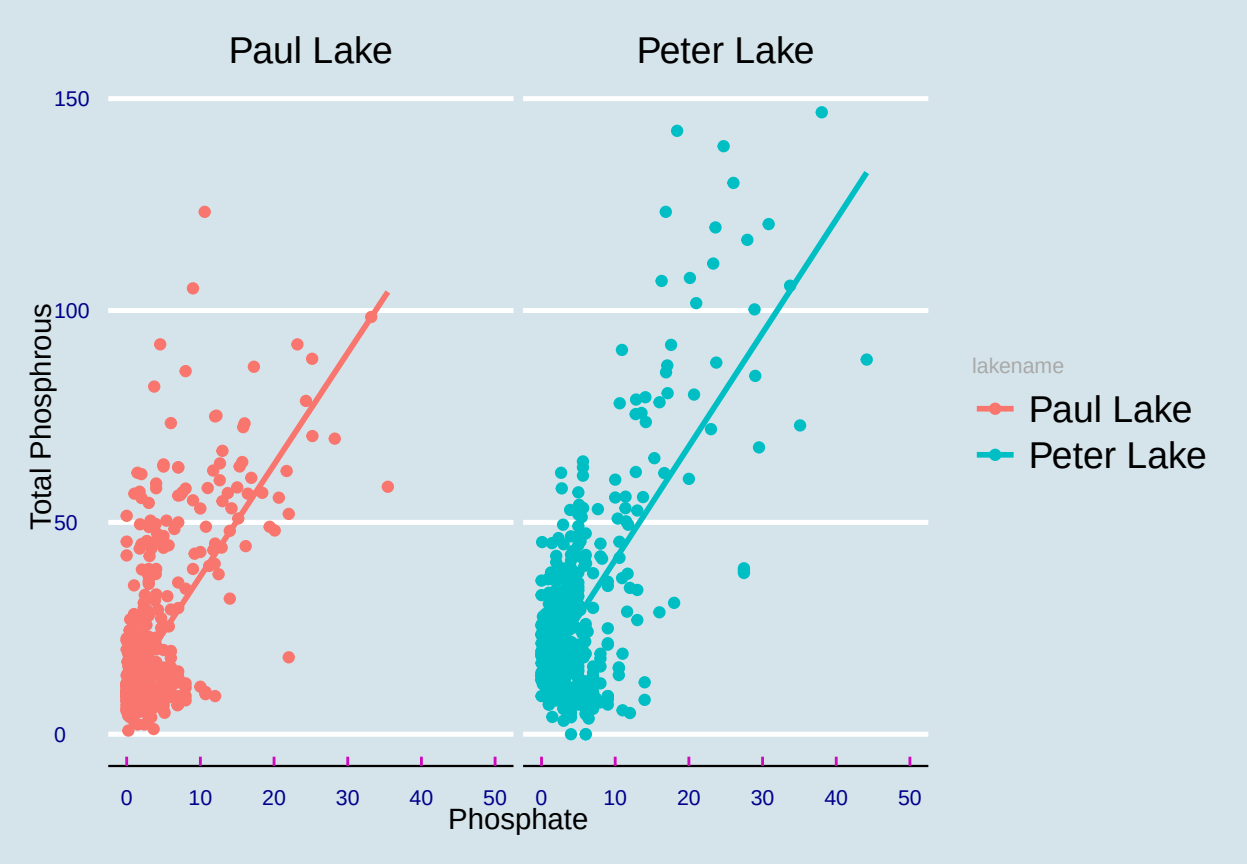
Create graphs

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
peter_paul_df %>%
  ggplot(aes(x=po4,y=tp_ug, color=lakename)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(vars(lakename))+
  ylab("Total Phosphrous")+
  xlab("Phosphate") +
  xlim(0,50) +
  ylim(0,150)
```

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

## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21948 rows containing missing values (`geom_point()`).
```

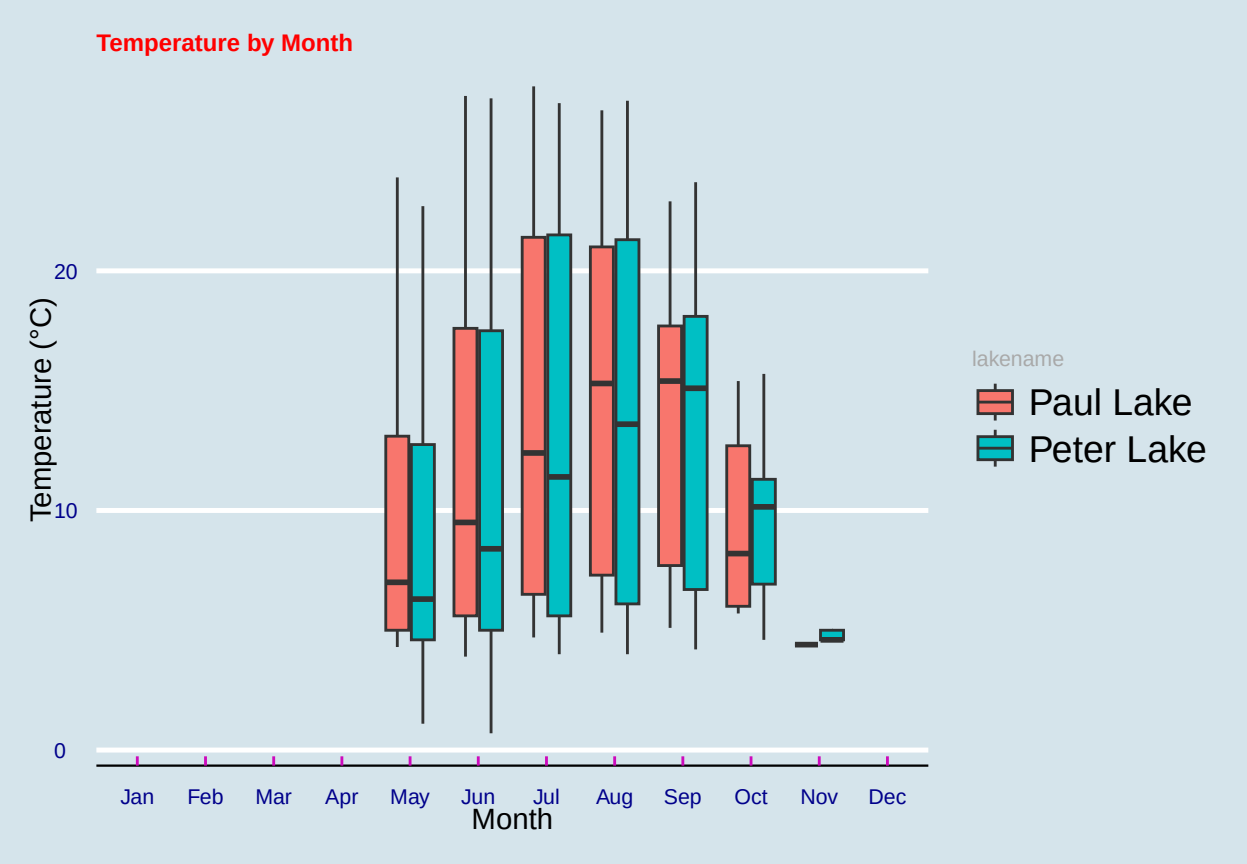


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

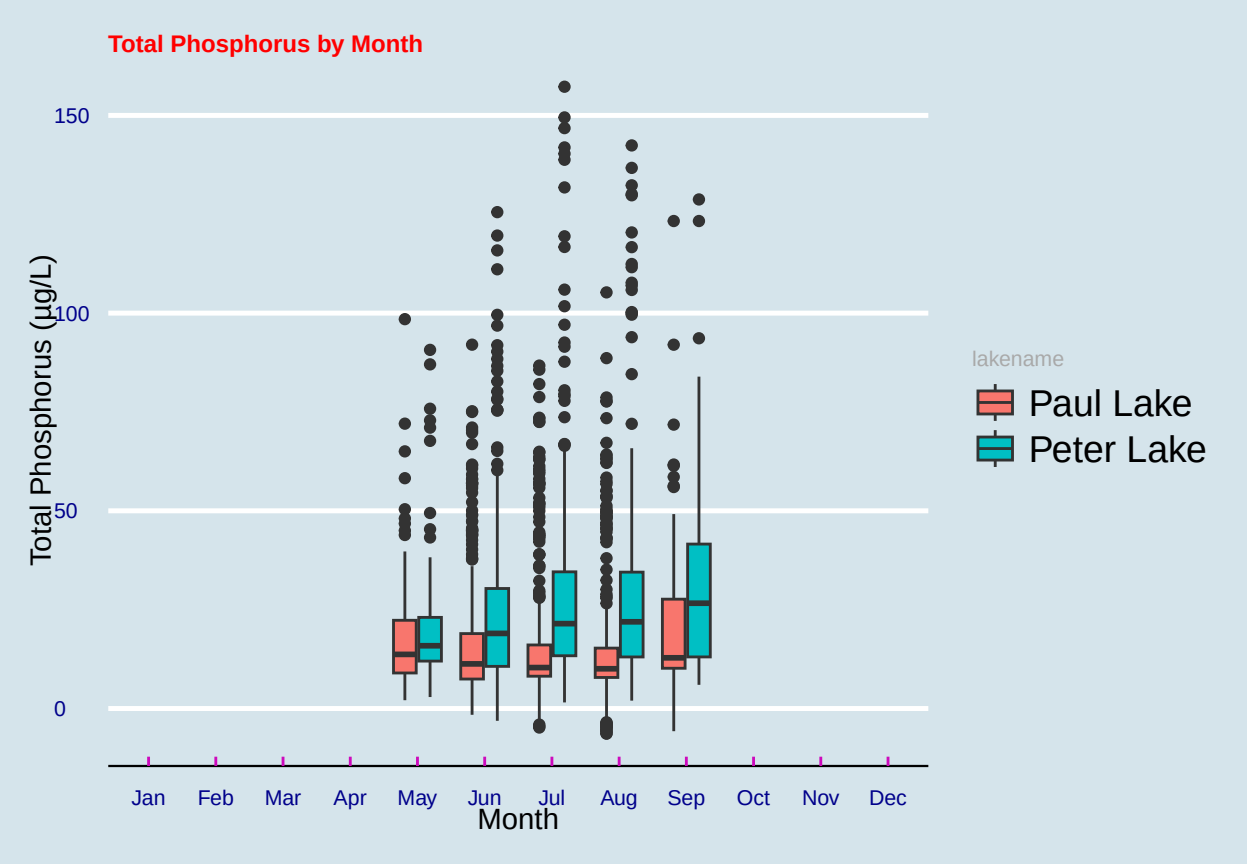
```
#5
#turning months into factor
peter_paul_df$month_f <- factor(
  peter_paul_df$month,
  levels=1:12,
  labels=month.abb,
  )

# Boxplot for Temperature
box_temp <- ggplot(peter_paul_df, aes(x = month_f, y = temperature_C, fill = lakename,)) +
  geom_boxplot() +
  scale_x_discrete(drop=F) +
  labs(title = "Temperature by Month",
       x = "Month",
       y = "Temperature (°C)")
box_temp
```

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



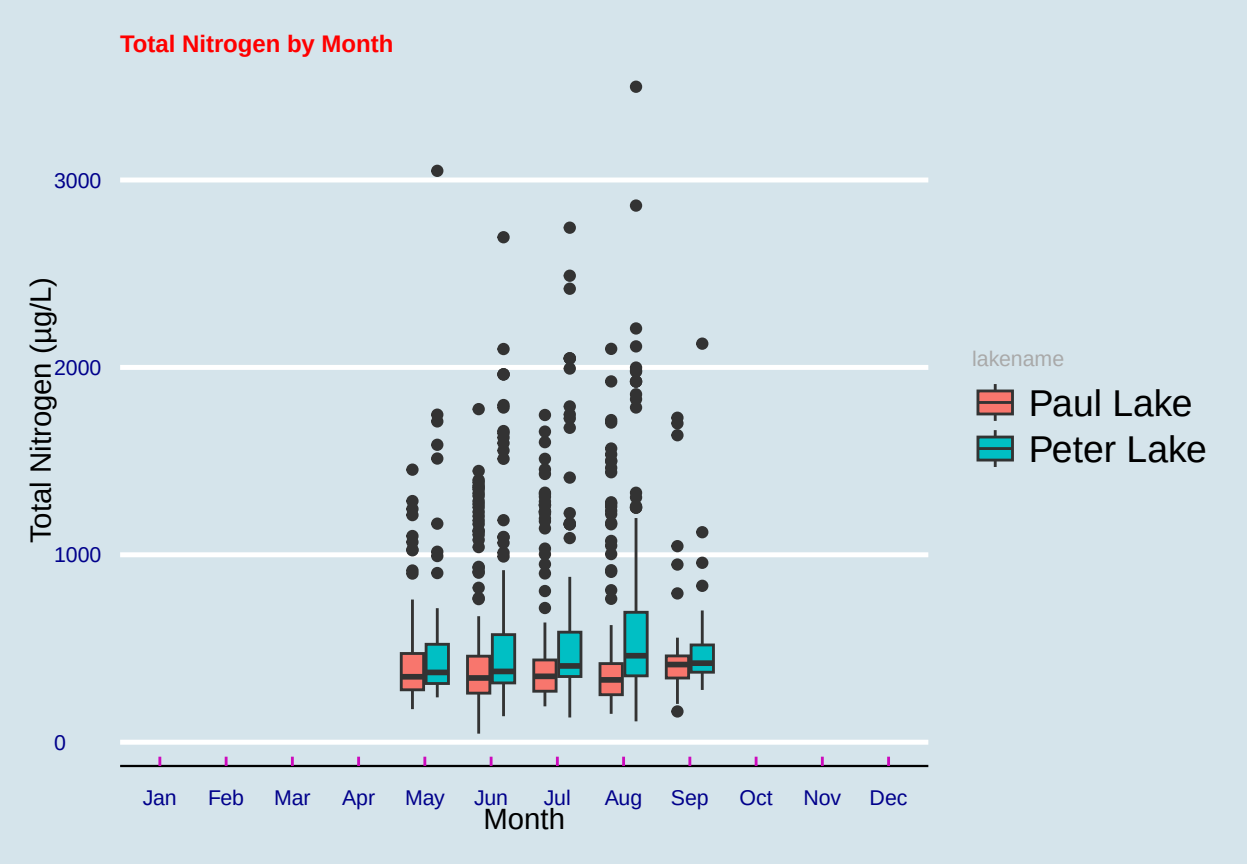

```
#boxplot for Phosphrous
box_tp <- ggplot(peter_paul_df, aes(x = month_f, y = tp_ug, fill = lakename,)) +
  geom_boxplot() +
  scale_x_discrete(drop=F) +
  labs(title = "Total Phosphorus by Month",
       x = "Month",
       y = "Total Phosphorus (µg/L)")
box_tp
```

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




```
#boxplot for Nitrogen
box_tn <- ggplot(peter_paul_df, aes(x = month_f, y = tn_ug, fill = lakename,)) +
  geom_boxplot() +
  scale_x_discrete(drop=F) +
  labs(title = "Total Nitrogen by Month",
       x = "Month",
       y = "Total Nitrogen (µg/L)")
box_tn
```

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



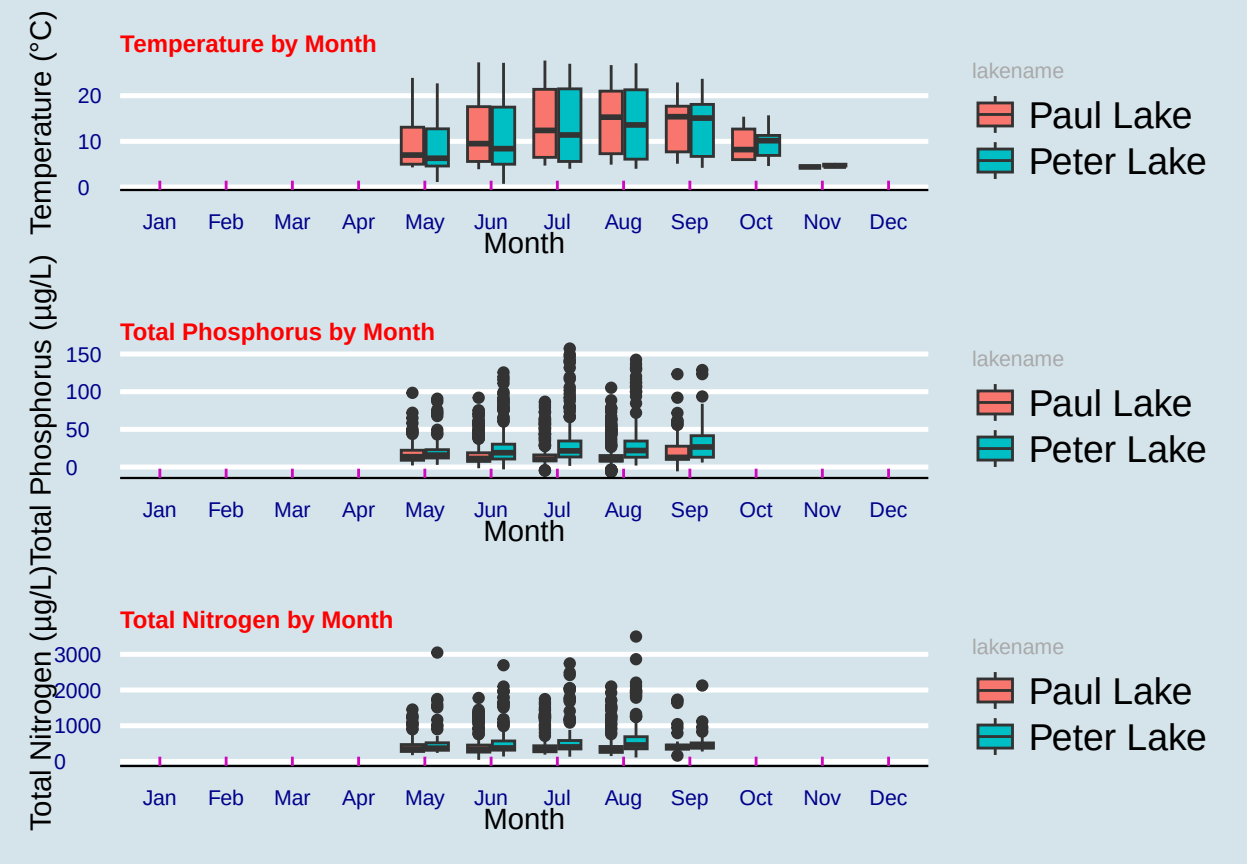

```
# Combine the three plots using cowplot
three_box <- plot_grid(
  box_temp, box_tp, box_tn,
  align = "hv", ncol = 1)
```

```
## 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()‘).
```

```
# Remove legends from individual plots
three_box <- three_box + theme(axis.title = element_blank(), legend.position = "none")

# Display the combined plot
three_box
```



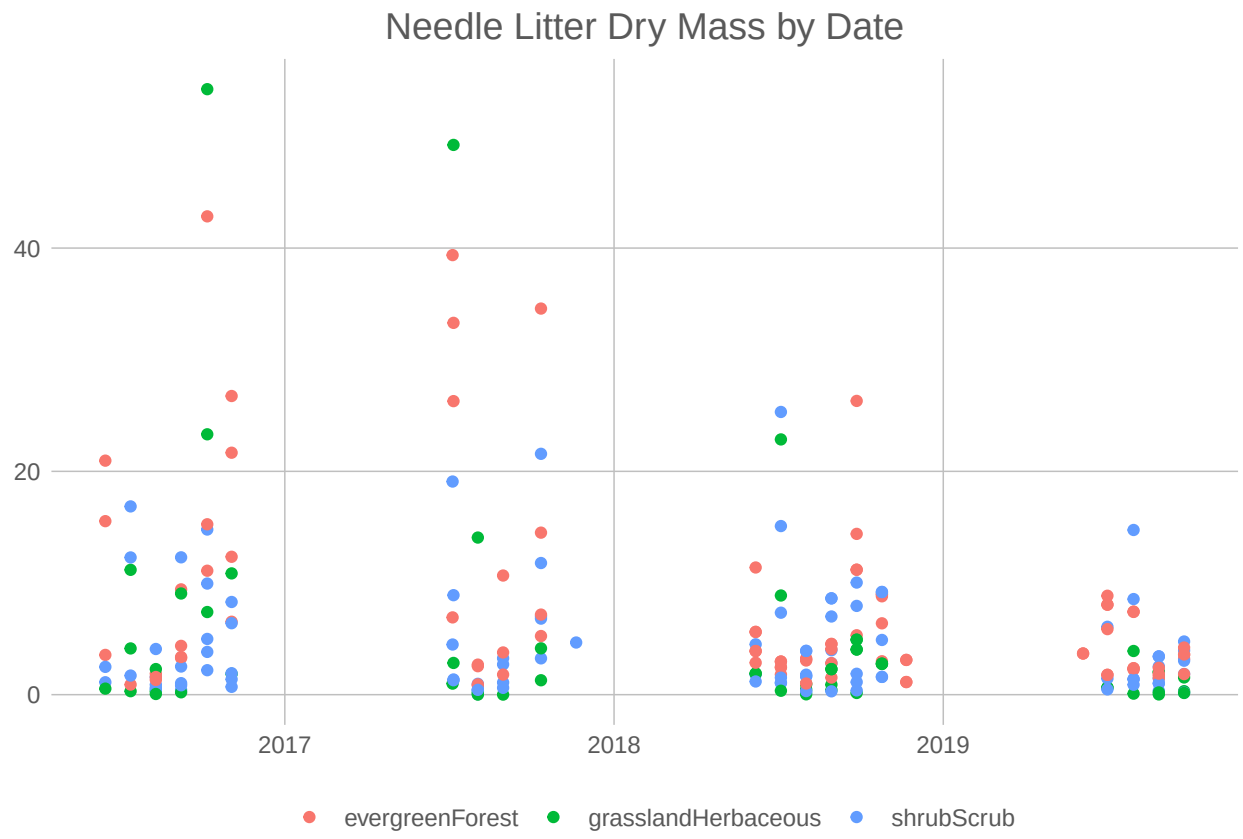
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Phosphrous and Nitrogen seem to rise with rising temperatures in the summer months of July and August. Peter Lake seems to have more nutrients amount in the water.

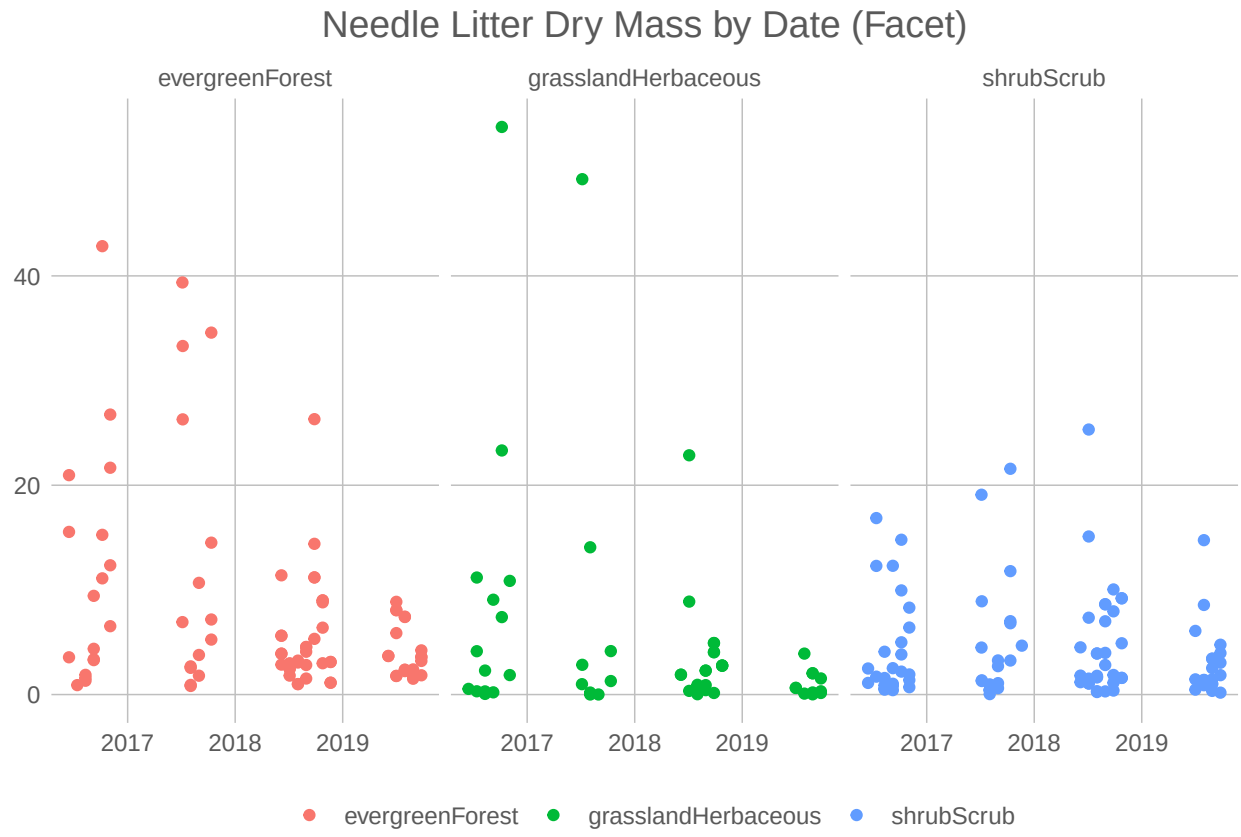
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
needles_df <- litter_df %>%
  filter(functionalGroup == "Needles")

# Create scatter plot of dry mass by date, colored by NLCD class
ggplot(needles_df, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = "Needle Litter Dry Mass by Date",
       x = "Date",
       y = "Dry Mass (g)") +
  theme_excel_new()
```




```
#7
ggplot(needles_df, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  labs(title = "Needle Litter Dry Mass by Date (Facet)",
       x = "Date",
       y = "Dry Mass (g)") +
  theme_excel_new()
```



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

Answer: Plot 7 is more effective since the visualization gives out separate timeline for each category of needle group while Plot 6 only provides a timeline with all three needle groups mixed. Plot 6 is much harder to interpret.