# Assignment 5: Data Visualization

Tori Newton

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.

```r
#1: Loading libraries
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 error
```

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

```
## here() starts at /home/guest/EDA_Spring2025
```

```
library(cowplot)
```

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

```
#Verify home directory
here()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
#Read in datasets
NTL_LTER_Nutrients <- read.csv(
file=here
("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
,
  stringsAsFactors = TRUE
)

NTL_LTER_Litter <- read.csv(
  file=here
  ("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)

#2: Dates as date format?
class(NTL_LTER_Litter$collectDate) #originally a factor
```

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

```
class(NTL_LTER_Nutrients$sampledate) #originally a factor
```

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

```
#Change the format to date
NTL_LTER_Litter$collectDate <- ymd (NTL_LTER_Litter$collectDate)
NTL_LTER_Nutrients$sampledate <- ymd (NTL_LTER_Nutrients$sampledate)

#Check again
class(NTL_LTER_Litter$collectDate) #Now a date
```

```
## [1] "Date"
```

```
class(NTL_LTER_Nutrients$sampledate) #Now a date
```

```
## [1] "Date"
```

## 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: Build a theme
my_theme <- theme_bw() +
  theme(
    plot.background = element_rect(
      fill = "lightblue"
    ),
    plot.title = element_text(
      color = "darkblue",
    ),
    legend.title = element_text(
      color='blue'
    )

  )

#Set as default theme
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.

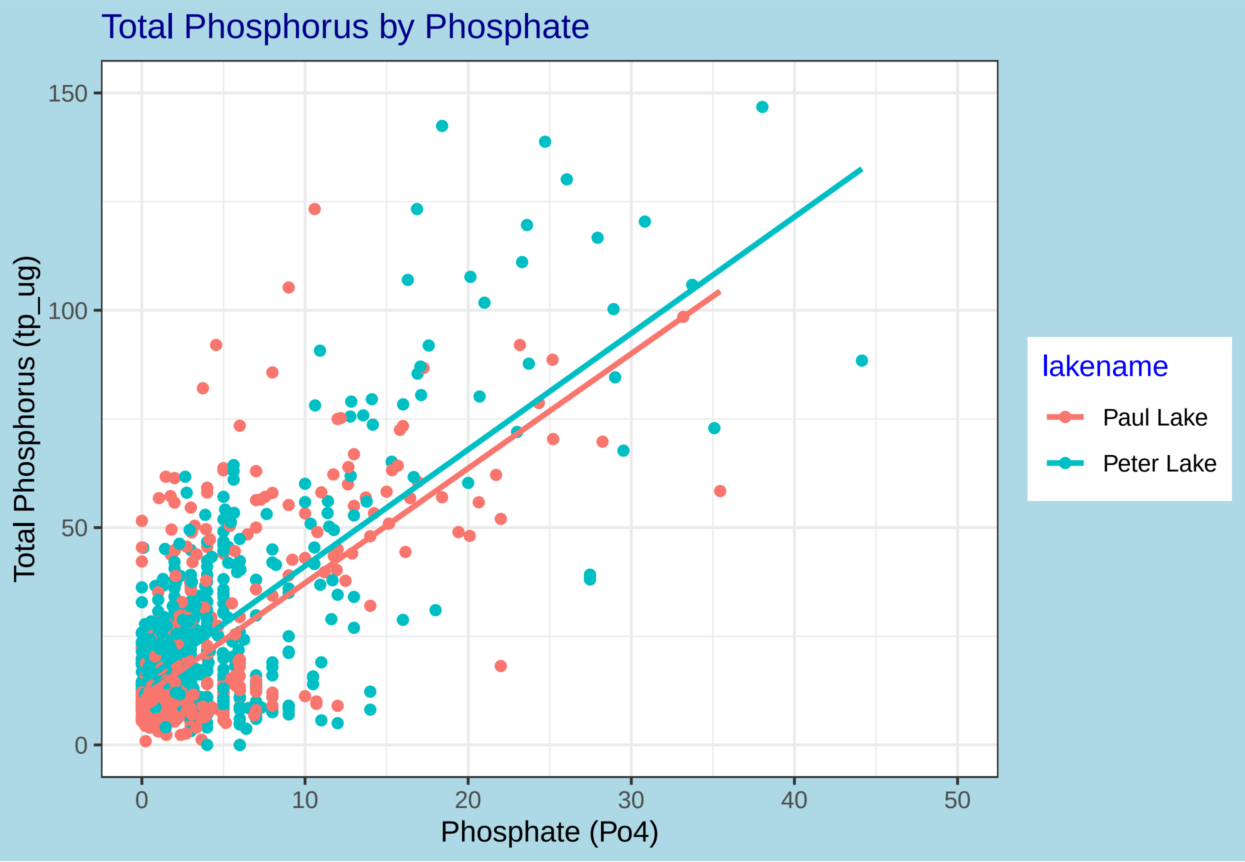
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4: Plot of total phosphorus by phosphate
NTL_LTER_Nutrients %>%
  ggplot(aes(x=po4, y=tp_ug, color=lakename)) +
  geom_point() +
  geom_smooth(method = lm, se=FALSE) +
  xlim(0, 50) +
  ylim(0, 150) +
  labs(title = "Total Phosphorus by Phosphate",
       x = "Phosphate (Po4)",
       y = "Total Phosphorus (tp_ug)")
```

```
## `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()`).
```

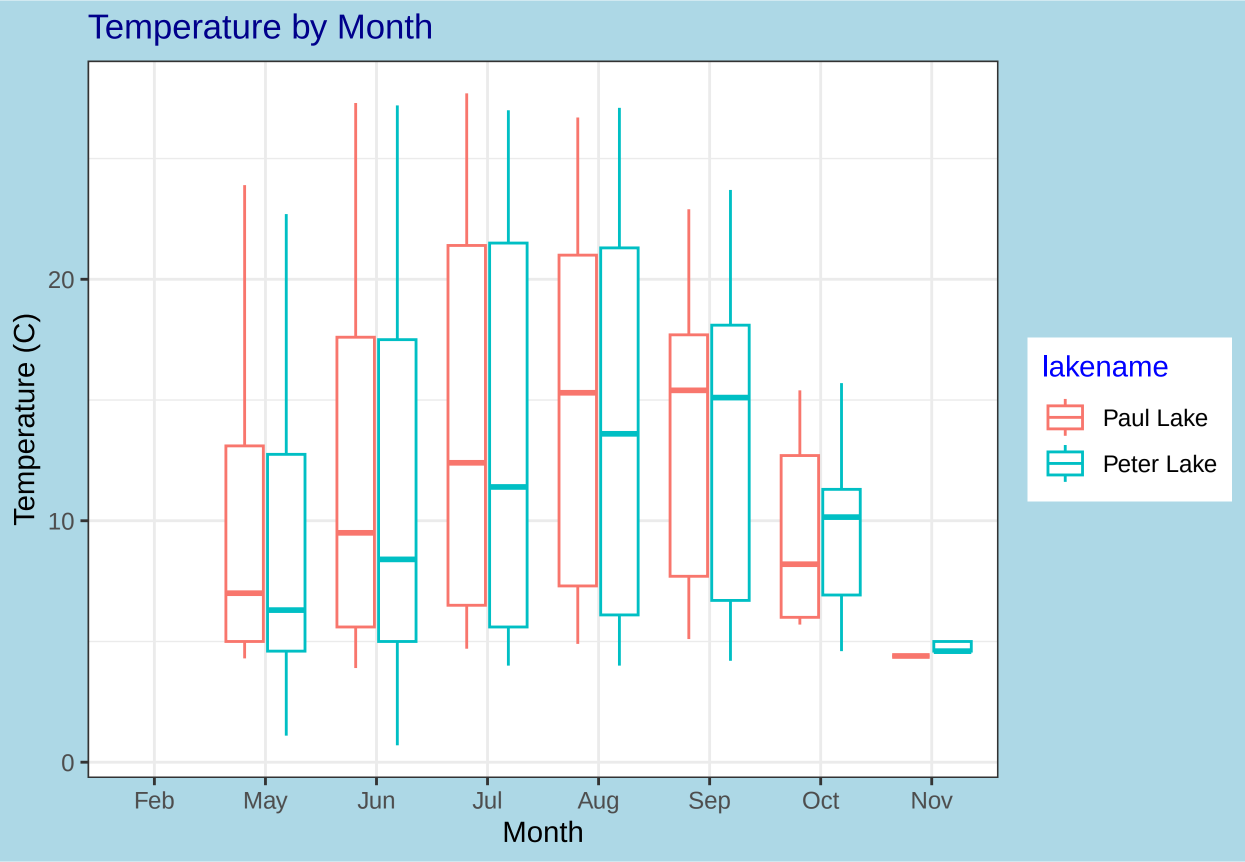


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
#Boxplot of temperature
plot1 <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=temperature_C, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name = "Month") +
  labs(title = "Temperature by Month",
       y = "Temperature (C)")
print(plot1)
```

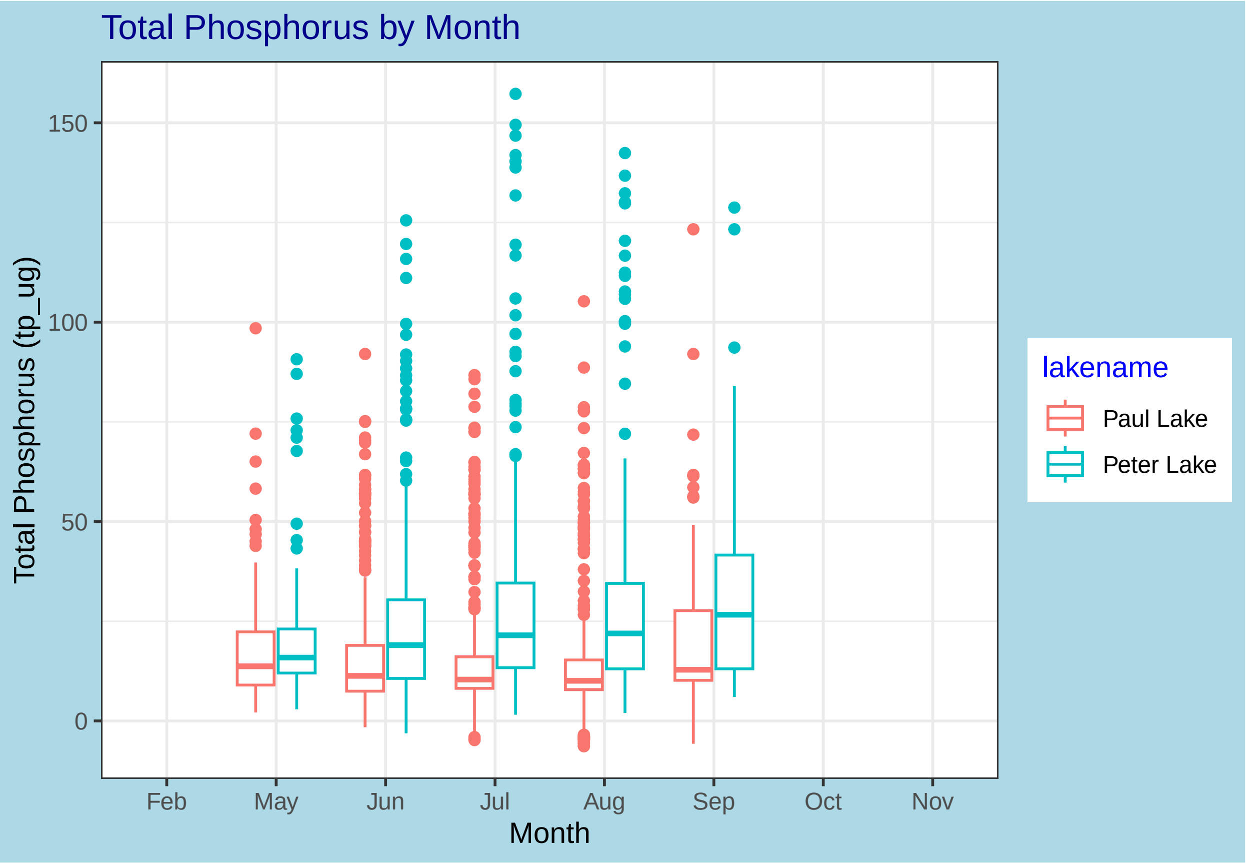
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```
#Take x axis label and legend away from plot 1 and make y axis label smaller
plot1_edited <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=temperature_C, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name = element_blank()) +
  labs(title = "Temperature by Month",
       y = "Temperature (C)") +
 theme(legend.position = "none",
      axis.title.y = element_text(size = 10))

#Boxplot of TP
plot2 <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=tp_ug, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name="Month") +
  labs(title = "Total Phosphorus by Month",
       y = "Total Phosphorus (tp_ug)")
print(plot2)
```

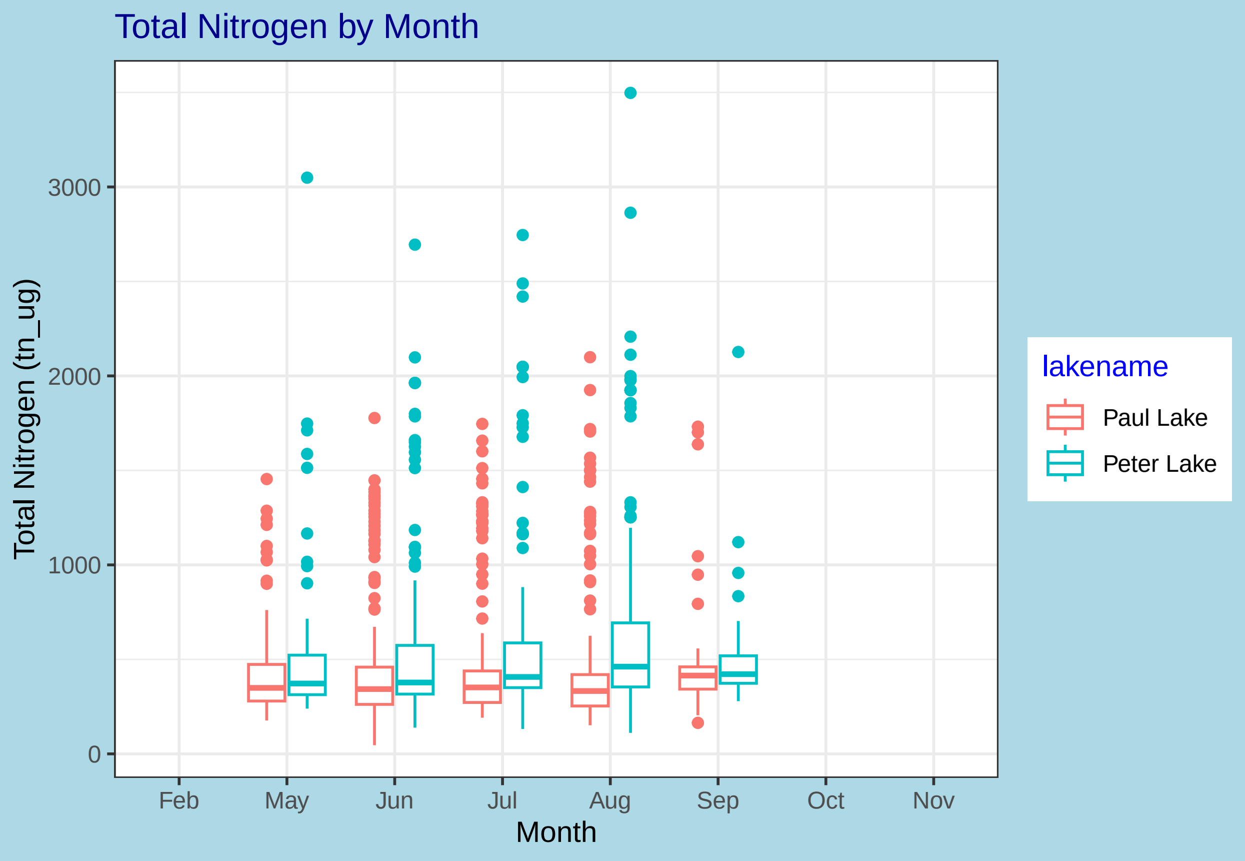
```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```
#Take x axis label and legend away from plot 2 and make y axis label smaller
plot2_edited <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=tp_ug, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name = element_blank()) +
  labs(title = "Total Phosphorus by Month",
       y = "Total Phosphorus (tp_ug)") +
 theme(legend.position = "none",
       axis.title.y = element_text(size = 8))

#Boxplot of TN
plot3 <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=tn_ug, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name="Month") +
  labs(title = "Total Nitrogen by Month",
       y = "Total Nitrogen (tn_ug)")
print(plot3)
```

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



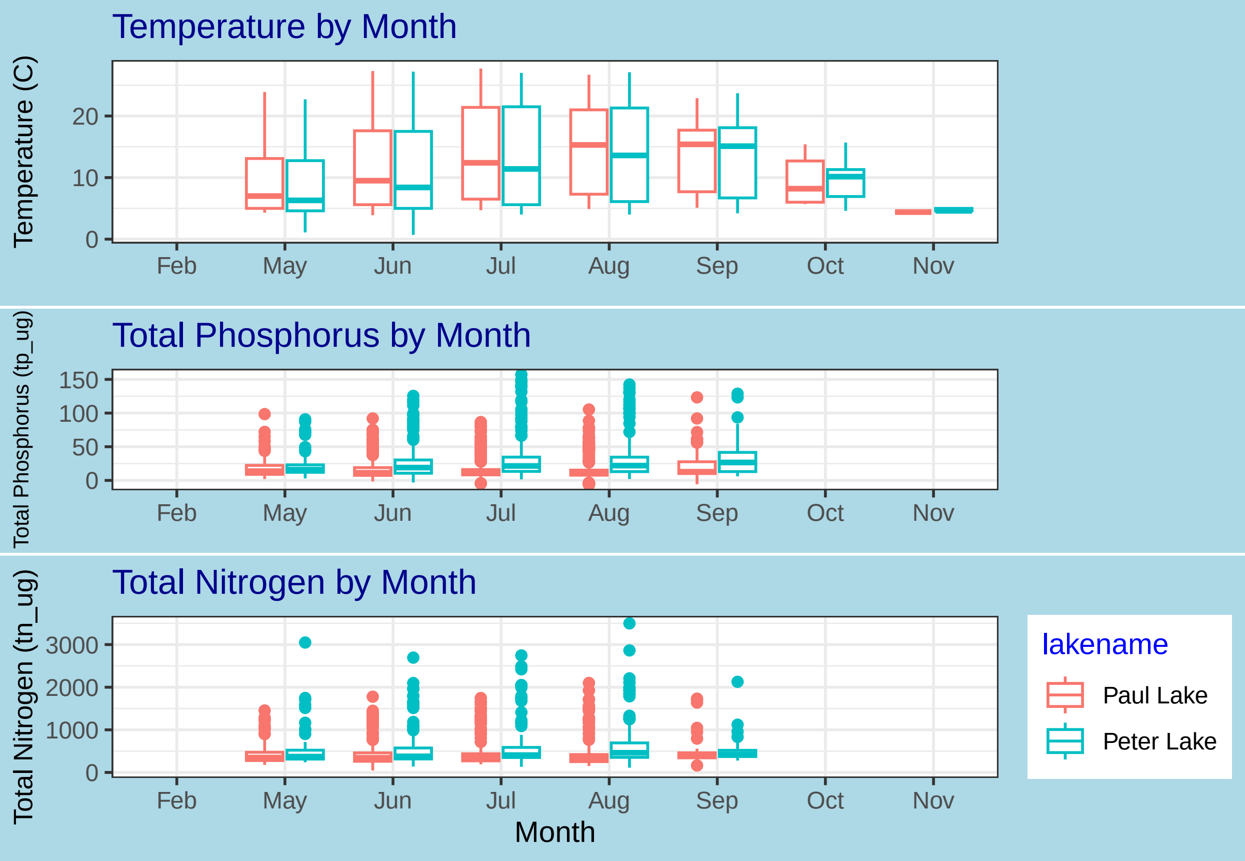
```
#Make y axis label smaller
plot3_edited <- NTL_LTER_Nutrients %>%
  ggplot(aes(x=month, y=tn_ug, color=lakename)) +
  geom_boxplot(aes(x=factor(month,levels =1:12,labels=month.abb))) +
  scale_x_discrete(name="Month") +
  labs(title = "Total Nitrogen by Month",
       y = "Total Nitrogen (tn_ug)") +
  theme(axis.title.y = element_text(size = 10))


#Cowplot that combines all graphs
plot_grid(plot1_edited, plot2_edited, plot3_edited, ncol = 1, align = 'v',
          rel_heights = c(1.25, 1))
```

```
## 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()`).
```



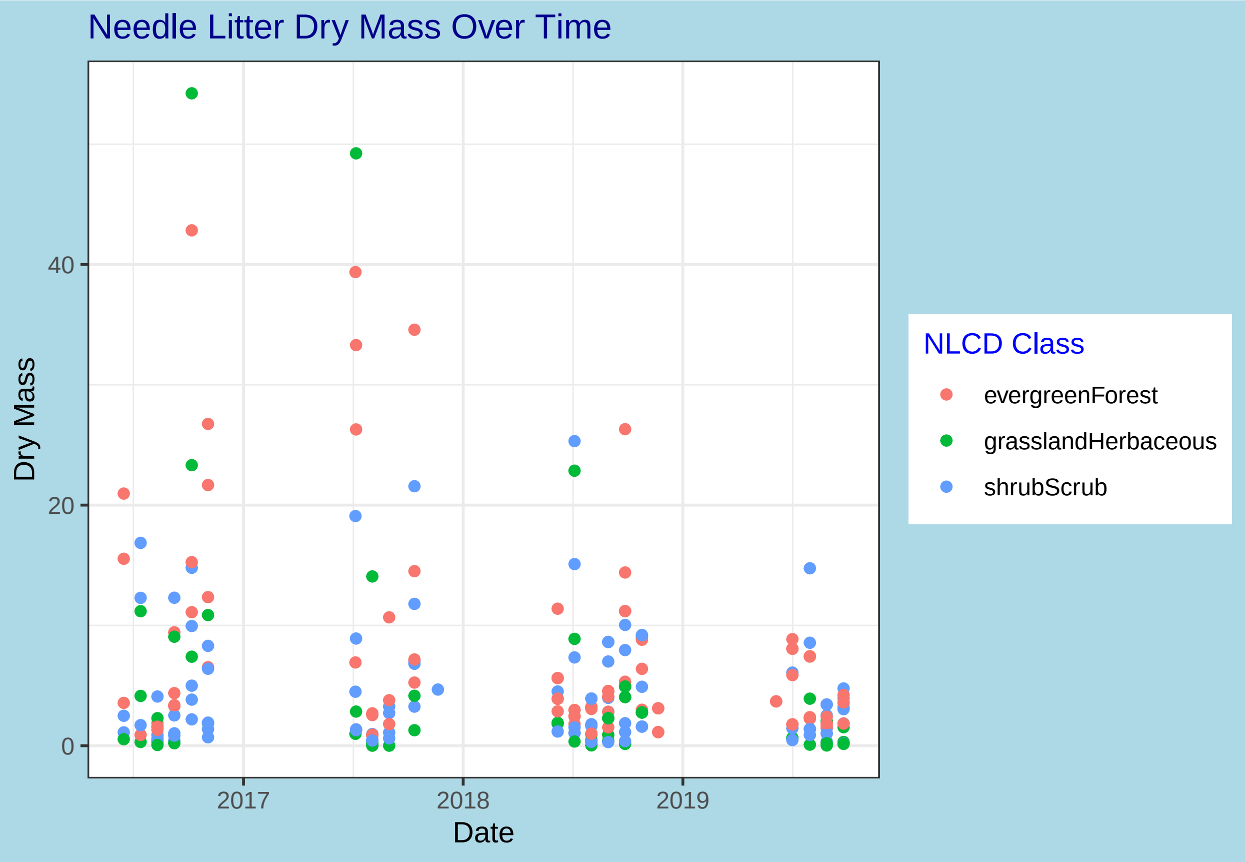
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: It is seen that temperature increases for both lakes in summer months and is particularly warm in August. The temperature for both lakes cools down in October. Total phosphorus and total nitrogen appear to be overall higher in Peter Lake than in Paul Lake.

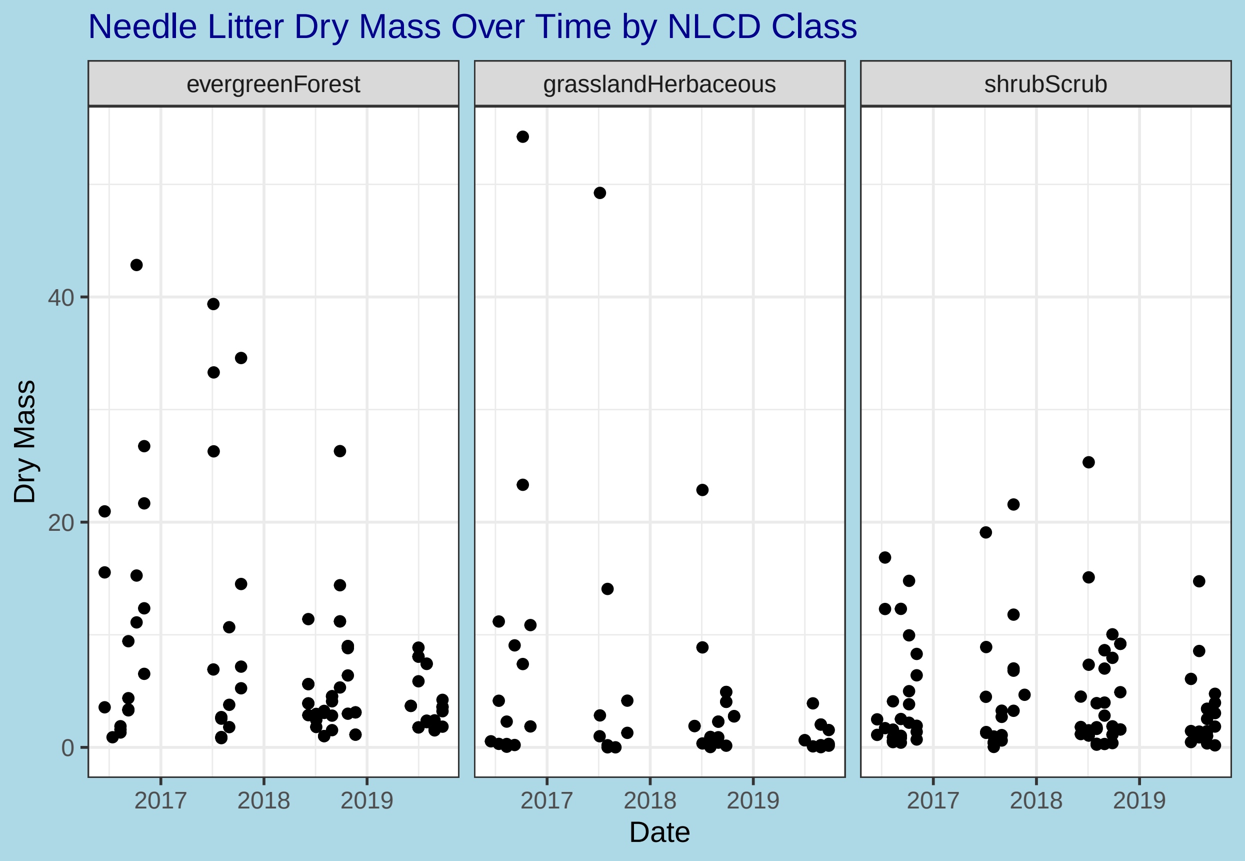
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
#Filter for only Needles
needles_subset <- NTL_LTER_Litter %>%
  filter(functionalGroup == "Needles")

#Create the plot
needles_subset %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(
    title = "Needle Litter Dry Mass Over Time",
    x = "Date",
    y = "Dry Mass",
    color = "NLCD Class"
  )
```



```
#7
#Separate NLCD classes into three facets
needles_subset %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  geom_point() +
  labs(
    title = "Needle Litter Dry Mass Over Time by NLCD Class",
    x = "Date",
    y = "Dry Mass",
  ) +
    facet_wrap(~ nlcdClass)
```



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

Answer: I think it depends what you are trying to analyze. I think plot 6 is better to compare trends across NLCD classes directly and plot 7 is better to analyze trends within each NLCD class separately.