Assignment 5: Data Visualization

Nusrat Noor

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.

```r
#1
#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("here")
#install.packages("cowplot")

#loading packages
library(tidyverse)
```

```
## -- 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 1.9.3     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 error
```

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

```
## here() starts at /home/guest/R/EDE_Fall2023
```

```r
library(cowplot)
```

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

```r
here() #checked home directory
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```r
Lake <- read.csv(here(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
                 stringsAsFactors = TRUE)
#loaded Lake dataset

Litter <- read.csv(here(
  "./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
                   stringsAsFactors = TRUE)
#loaded Litter dataset

#2
str(Lake) #checked structure of dataset to see date format (it was not date format)
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : Factor w/ 1103 levels "1984-05-27","1984-05-28",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```r
Lake$sampledate <- ymd(Lake$sampledate) #changed format to date

str(Litter) #checked structure of dataset to see date format (it was not date format)
```

```
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 9 8 9 11 7 7 4 4 4 4 ...
##  $ trapID          : Factor w/ 15 levels "NIWO_040_139",..: 11 10 11 13 9 9 5 5 5 5 ...
##  $ collectDate     : Factor w/ 24 levels "2016-06-16","2016-07-14",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ functionalGroup : Factor w/ 8 levels "Flowers","Leaves",..: 6 5 8 6 4 2 2 6 7 8 ...
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 1 1 1 ...
##  $ subplotID       : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : Factor w/ 3 levels "evergreenForest",..: 3 1 3 1 3 3 2 2 2 2 ...
##  $ plotType        : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum   : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...
```

```
Litter$collectDate <- ymd(Litter$collectDate) #changed format to 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

my_theme <- theme_classic() +
  theme(axis.title = element_text(color = "black"), #made axis titles black
        plot.background = element_rect(color = "gray"), #made plot background gray
        axis.text = element_text(color = "black"), #made axis text black
        legend.position = "right") #set legend position to the right of plot
#set theme
theme_set(my_theme)
```

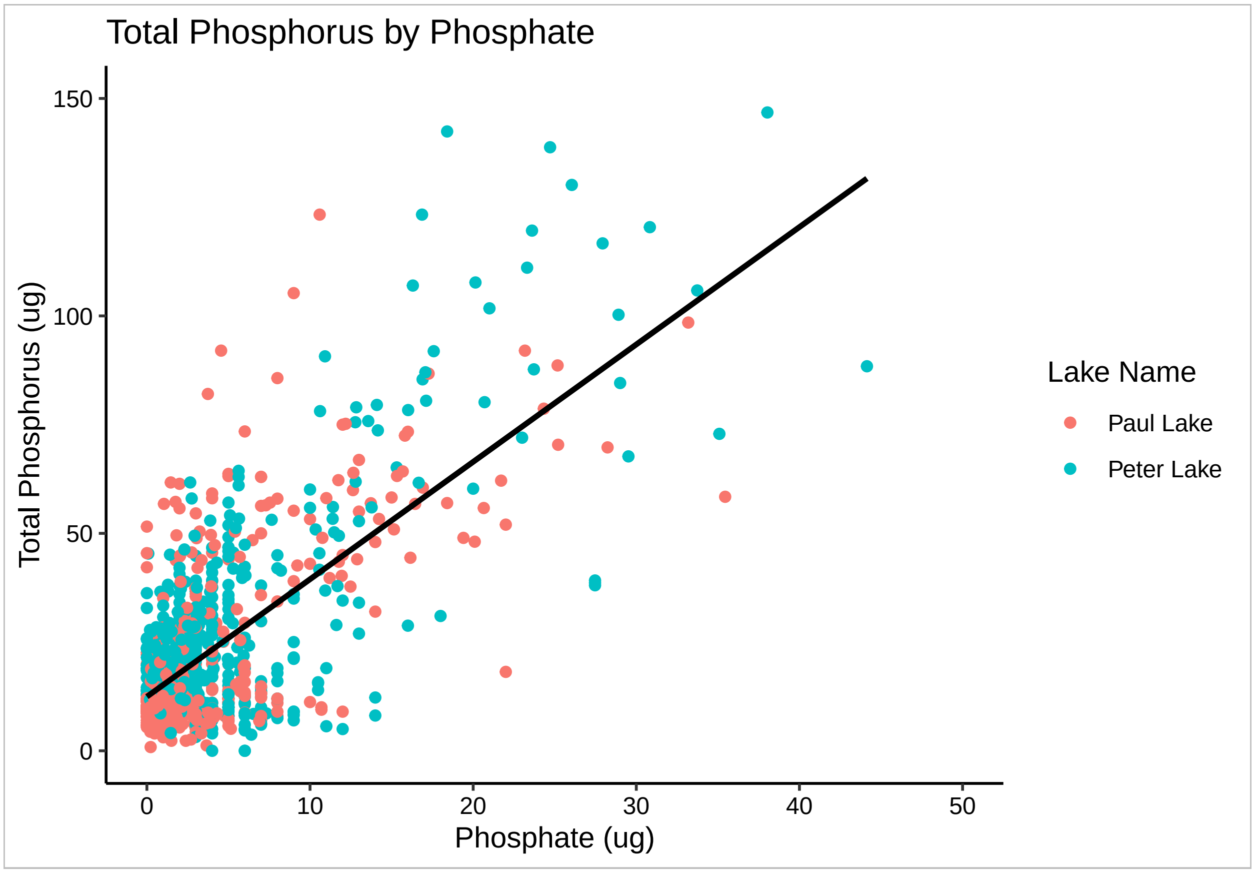
Create graphs

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 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

LakePlot <- ggplot(Lake, aes(x = po4, y = tp_ug, color = lakename)) +
  #created plot for total phosphorus and phosphate
  geom_point(na.rm = TRUE) + #got rid of NAs
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  #added a best fit line and made it black
  xlim(c(0, 50)) + #set limits for x-axis
  ylim(c(0, 150)) + #set limits for y-axis
  labs(x = "Phosphate (ug)", y = "Total Phosphorus (ug)", title = "Total Phosphorus by Phosphate",
       color = "Lake Name") #changed labels

LakePlot #called back the plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).
```

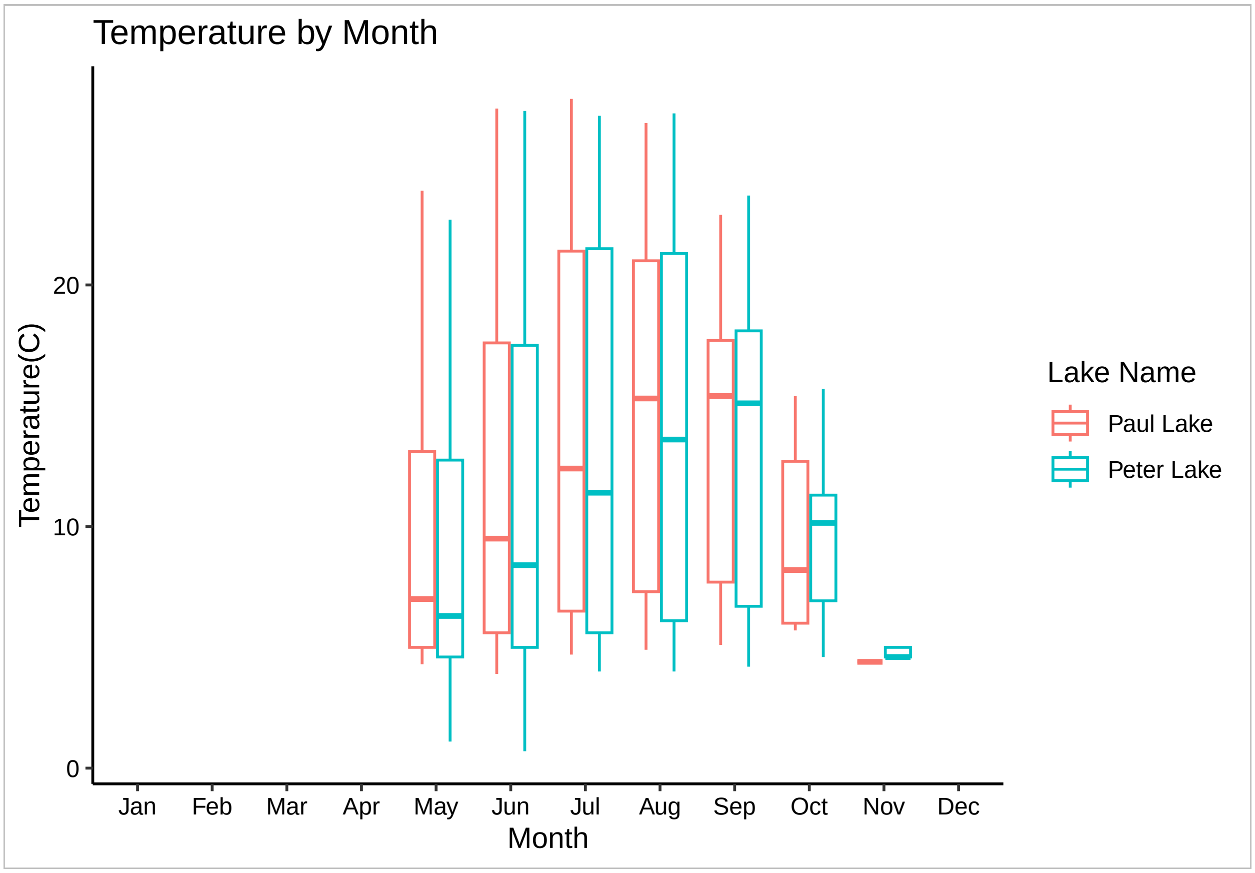


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

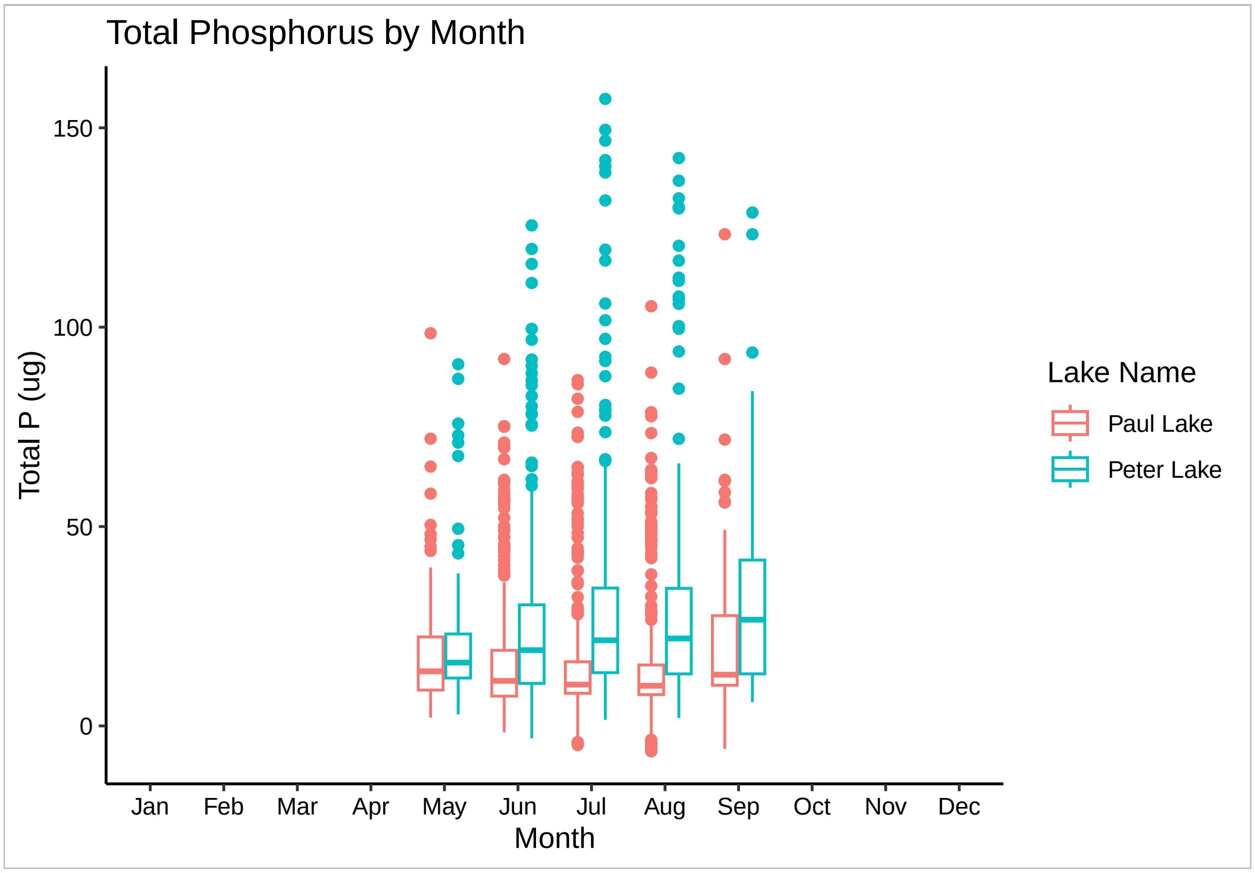
```
#5

temp_plot <- ggplot(Lake, aes(x = month.abb[month], y = temperature_C, color = lakename)) +
  geom_boxplot() + #made boxplot of temperatures by date collected and separated by lakename
  scale_x_discrete(limits = month.abb) + #made x-axis labels show every month in abbreviated form
  labs(x = "Month", y = "Temperature(C)", title = "Temperature by Month", color = "Lake Name")
#changed labels
print(temp_plot) #called back plot
```

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



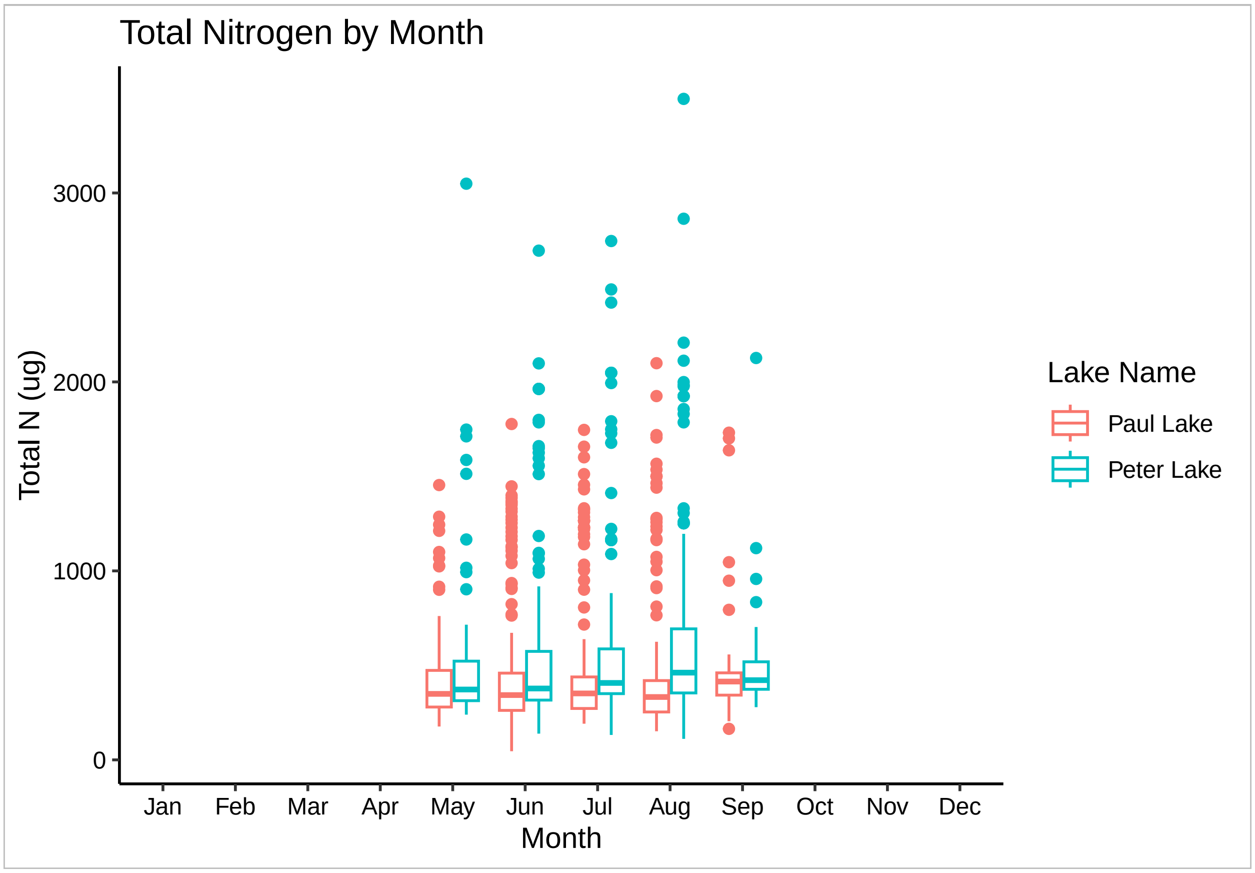
```r
TP_plot <- ggplot(Lake, aes(x = month.abb[month], y = tp_ug, color = lakename)) +
  geom_boxplot() + #created boxplot of total phosphorus by month and separated by lakename
  scale_x_discrete(limits = month.abb) + #made x-axis labels show every month in abbreviated form
  labs(x = "Month", y = "Total P (ug)", title = "Total Phosphorus by Month",
       color = "Lake Name") #changed labels
print(TP_plot) #called back the plot
```

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



```
TN_plot <- ggplot(Lake, aes(x = month.abb[month], y = tn_ug, color = lakename)) +
  geom_boxplot() + #created boxplot of total nitrogen by month and separated by lakename
  scale_x_discrete(limits = month.abb) + ##made x-axis labels show every month in abbreviated form
  labs(x = "Month", y = "Total N (ug)", title = "Total Nitrogen by Month", color = "Lake Name")
#changed labels
print(TN_plot) #called back plot
```

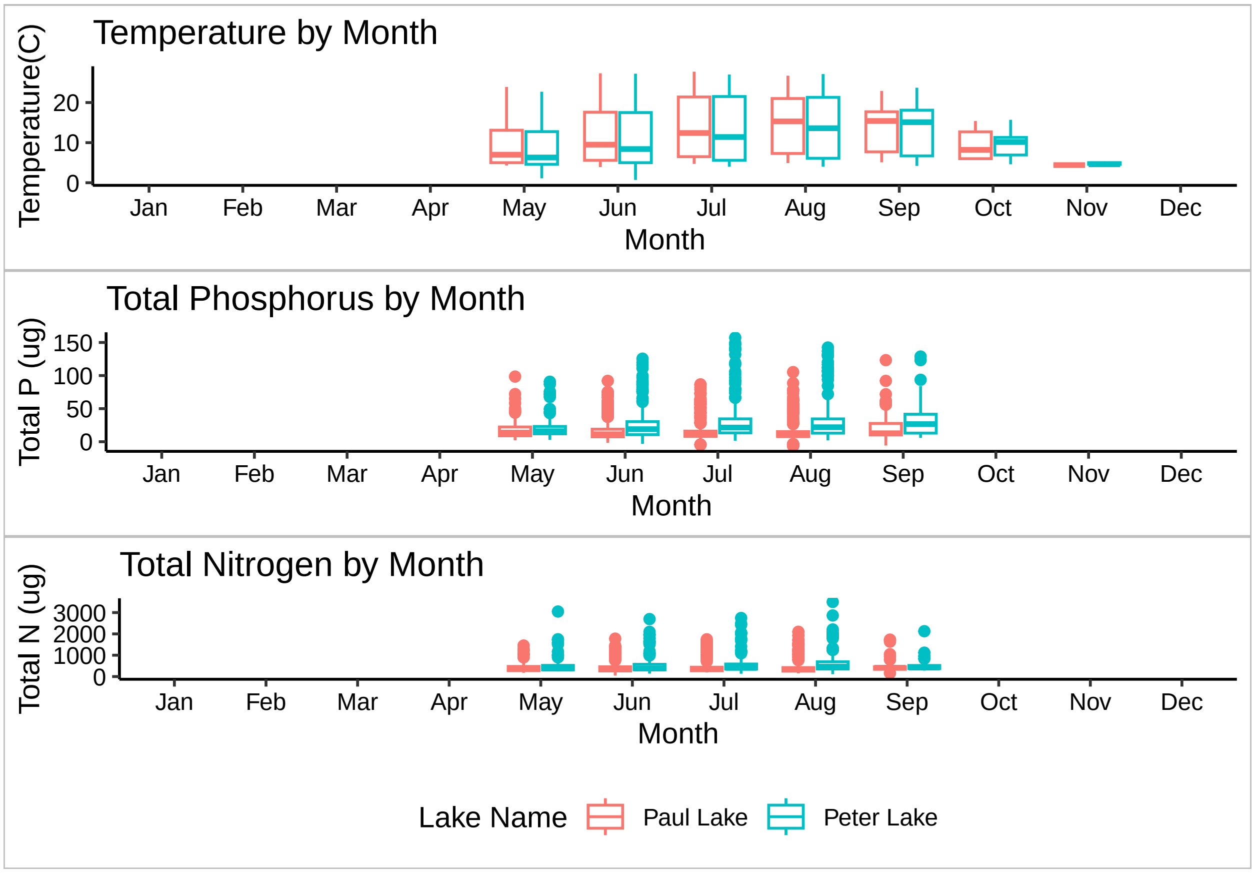
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```



```
#created cowplot of all three above plots

plot_all <- plot_grid(
  temp_plot + theme(legend.position = "none"), #got rid of legend
  TP_plot + theme(legend.position = "none"), #got rid of legend
  TN_plot + theme(legend.position = "bottom"),
  nrow = 3, align = 'h', rel_heights = c(1, 1, 1.25)) +
  #set number of rows, aligned horizontally
  theme(axis.text = element_text(size = 10)) #changed axis text to size 10
```

```
## 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()`).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
plot_all #called back the plot
```

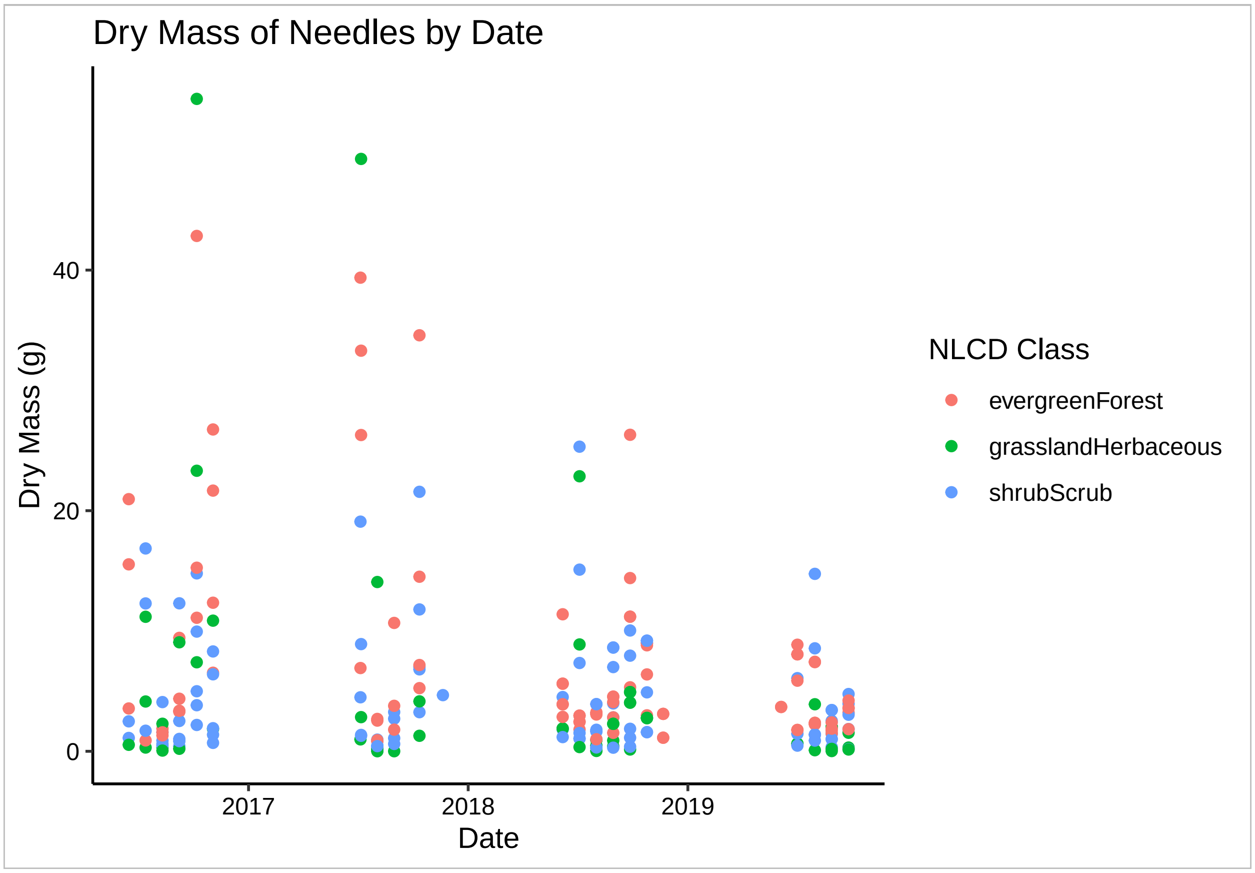


Question: What do you observe about the variables of interest over seasons and between lakes?

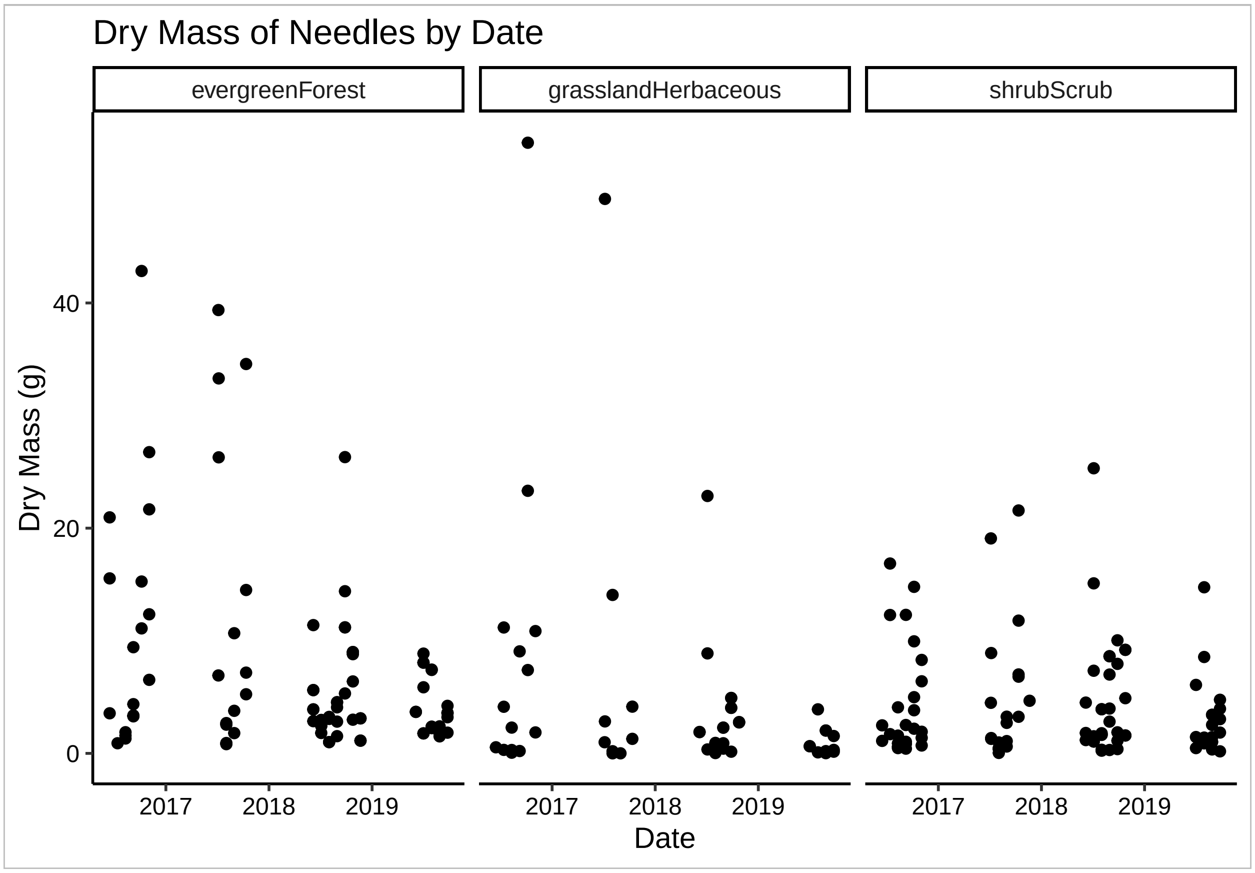
> Answer: I notice that warmer months have higher levels of total phosphorus and total nitrogen and also have a wider spread. The average total nitrogen and phosphorus levels in Peter lake is higher and overall, has more spread.

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
Needles1 <- ggplot(subset(Litter, functionalGroup == "Needles"),
                   aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() + #plotted a subset of only Needles, set axis, and separated by ncld class
  labs(x = "Date", y = "Dry Mass (g)", title = "Dry Mass of Needles by Date", color = "NLCD Class")
#changed labels
print(Needles1) #called back the plot
```



```
#7
Needles2 <- ggplot(subset(Litter, functionalGroup == "Needles"), aes(x = collectDate, y = dryMass)) +
  geom_point() + #plotted a subset of only Needles and set axis
  facet_wrap(vars(nlcdClass), ncol = 3) + #separated the ncld classes into three facets
  labs(x = "Date", y = "Dry Mass (g)", title = "Dry Mass of Needles by Date")
#changed labels and title
print(Needles2) #called out the plot
```



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

> Answer: Plot 7 is more effective because it separates the data much more clearly so that the points not overlapping as much as they are in plot 6.