Assignment 5: Data Visualization

Zhaoxin Zhang

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
#Load packages
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 errors
```

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

```
## here() starts at /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

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

```
#verify home directory
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
#Read in data
nutrients <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)
litter <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)
#2
#Check the format
class(nutrients$sampledate)
```

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

```
class(litter$collectDate)
```

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

```
#Both of them are "factor
#Thus we convert them to data
nutrients$sampledate <- ymd(nutrients$sampledate)
litter$collectDate <- ymd(litter$collectDate)
class(nutrients$sampledate)
```

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

```
class(litter$collectDate)
```

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

```
#Now their formats are "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
mytheme <- theme_classic(base_size = 14) +
  theme(plot.background = element_rect(fill = "lightblue", color = NA), #plot background
    plot.title = element_text(color = "darkblue", size = 14, face = "bold"), #plot title
    legend.position = "right",
    legend.background = element_rect(color='black',size=0.3)) #legend
```

```
## Warning: The ‘size‘ argument of ‘element_rect()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
theme_set(mytheme)
```

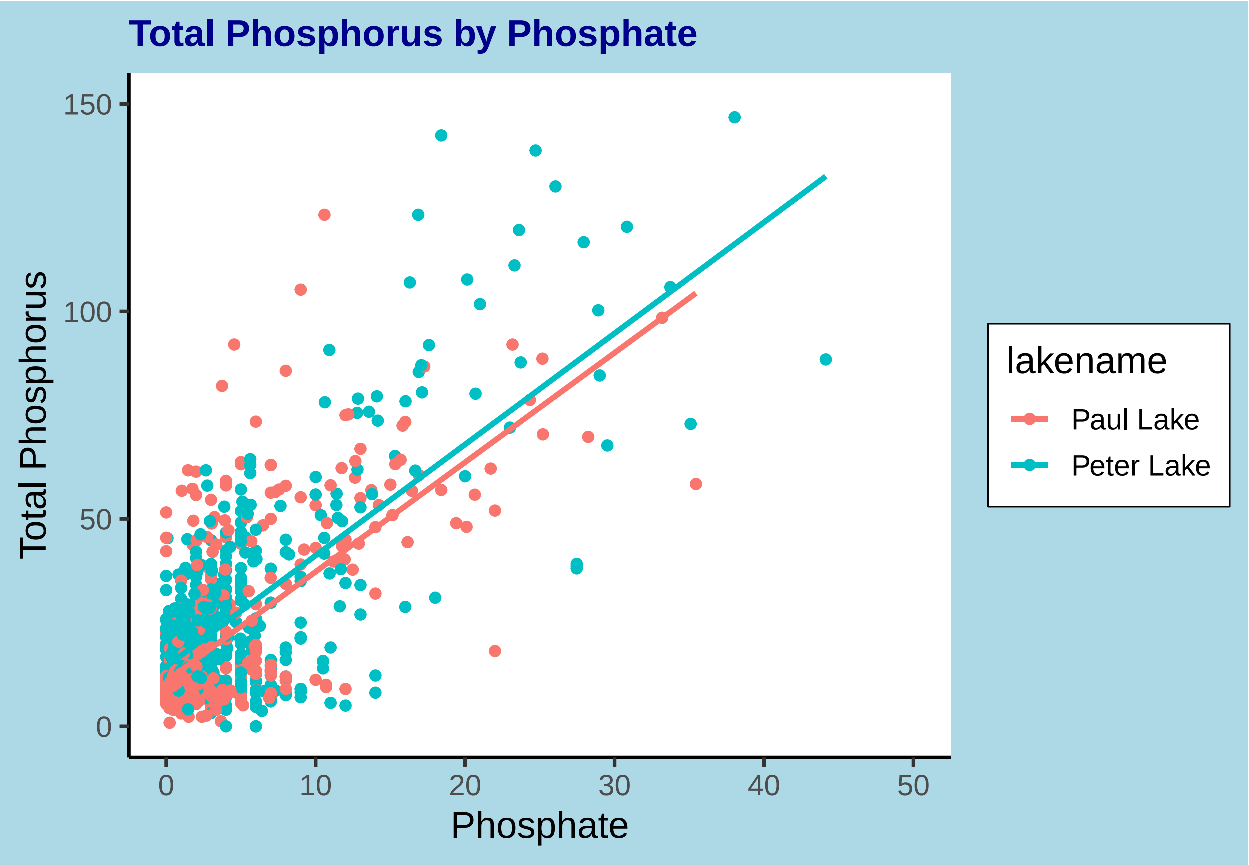
Create graphs

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
total_phosphorus <- 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",
       y = "Total Phosphorus")+
  mytheme

print(total_phosphorus)
```

```
## ‘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()‘).
```

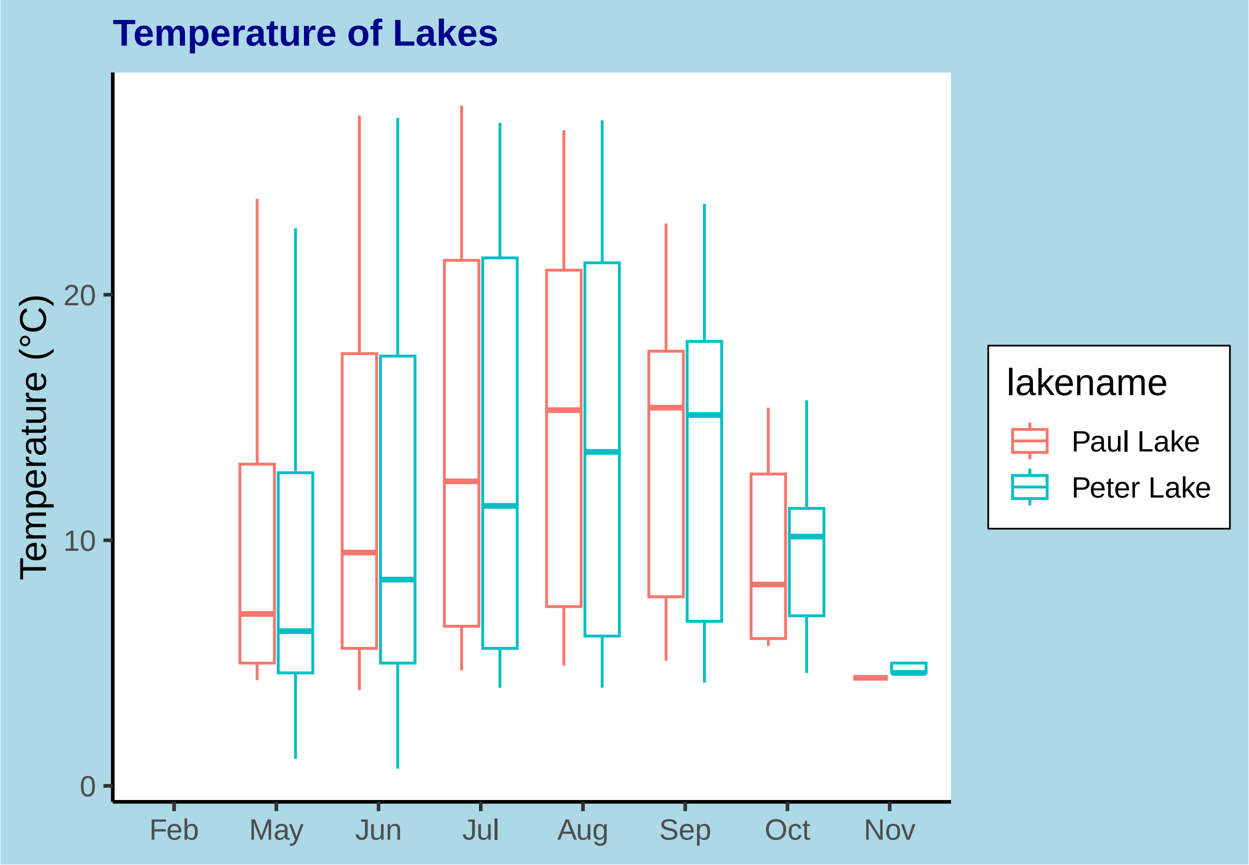


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

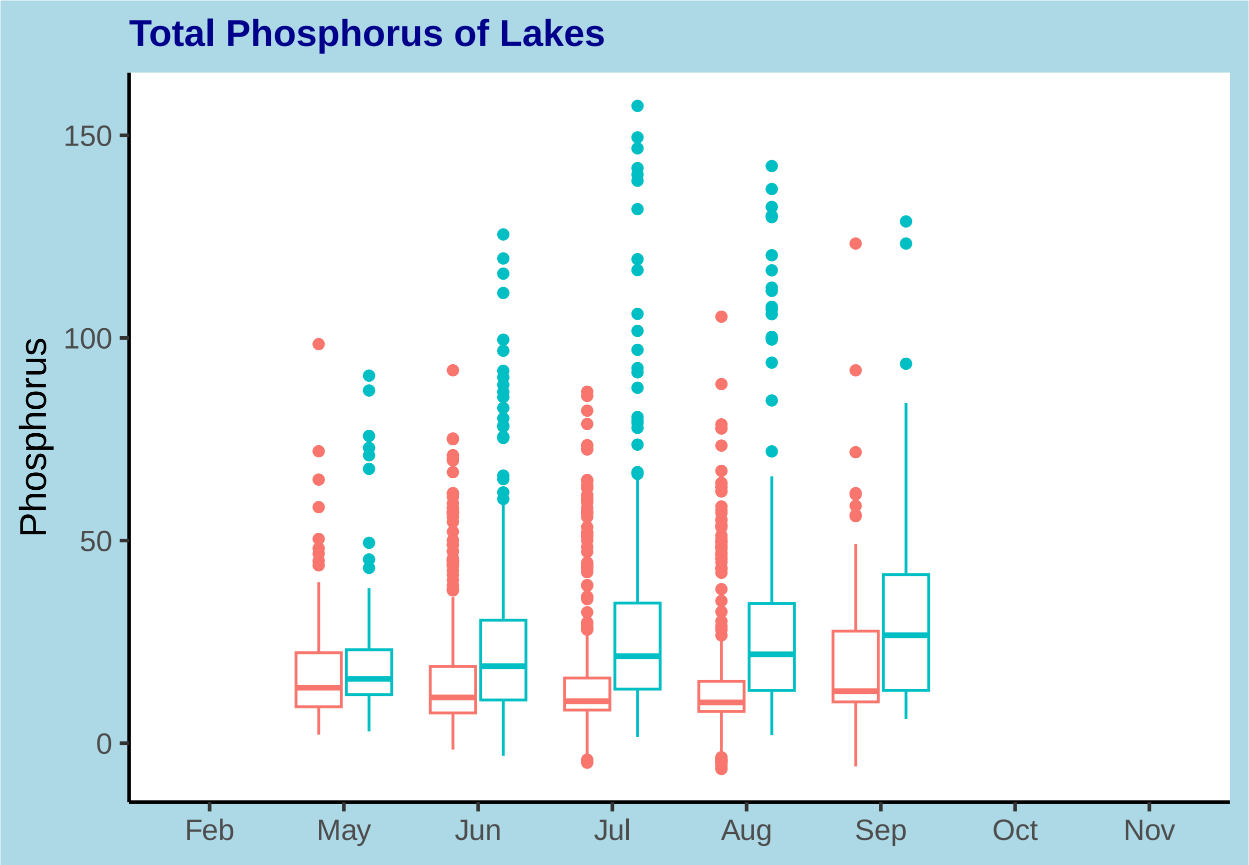
```r
#5
#Convert month to a factor -- with 12 levels, labelled with month names
nutrients$month<-factor(nutrients$month,
       levels=1:12,
       labels = month.abb)
#boxplot of (a) temperature
temperature <- nutrients %>%
  ggplot(aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color=lakename))+
  labs(title = "Temperature of Lakes",
       y = "Temperature (°C)") +
  mytheme+
  theme(axis.title.x = element_blank())

print(temperature)
```

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



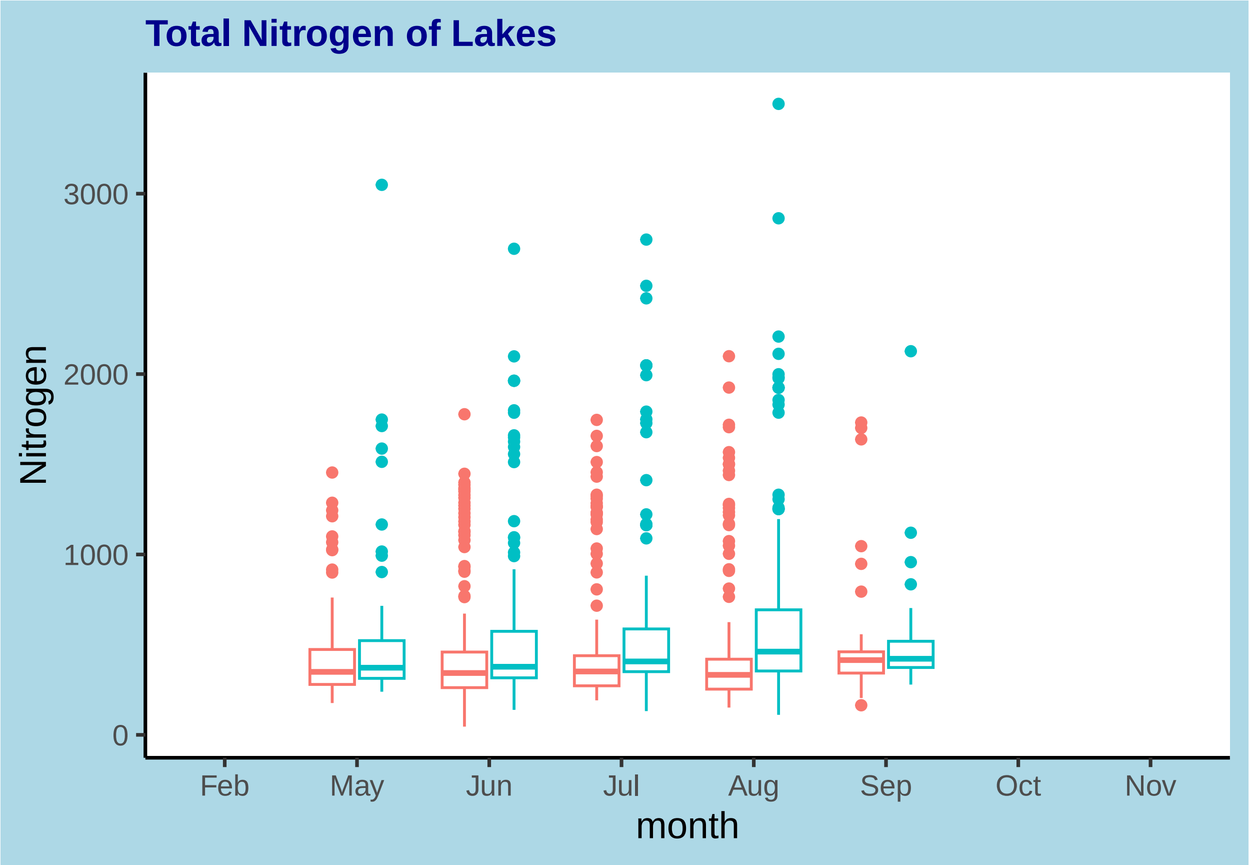
```
#boxplot of (b) TP
TP <- nutrients %>%
  ggplot(aes(x=month, y=tp_ug))+
  geom_boxplot(aes(color=lakename))+
  labs(title = "Total Phosphorus of Lakes",
       y= "Phosphorus")+
  mytheme+
  theme(axis.title.x = element_blank(), legend.position = "none")

print(TP)
```

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



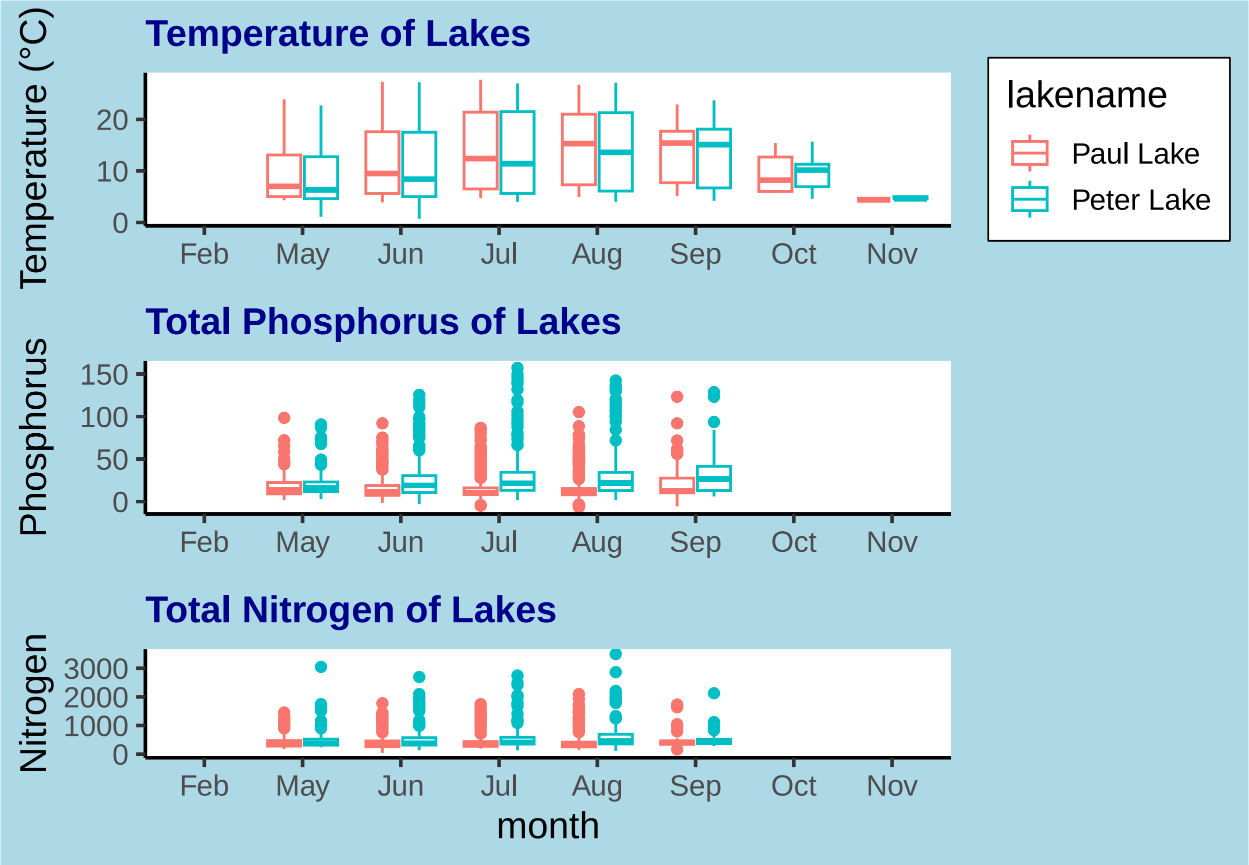
```
#boxplot of (c) TN
TN <- nutrients %>%
  ggplot(aes(x=month, y=tn_ug))+
  geom_boxplot(aes(color=lakename))+
  labs(title = "Total Nitrogen of Lakes",
       y= "Nitrogen")+
  mytheme+
  theme(legend.position = "none")

print(TN)
```

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



```
plot_grid(temperature, TP, TN, nrow = 3, align = "v")
```

```
## 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()‘).
```



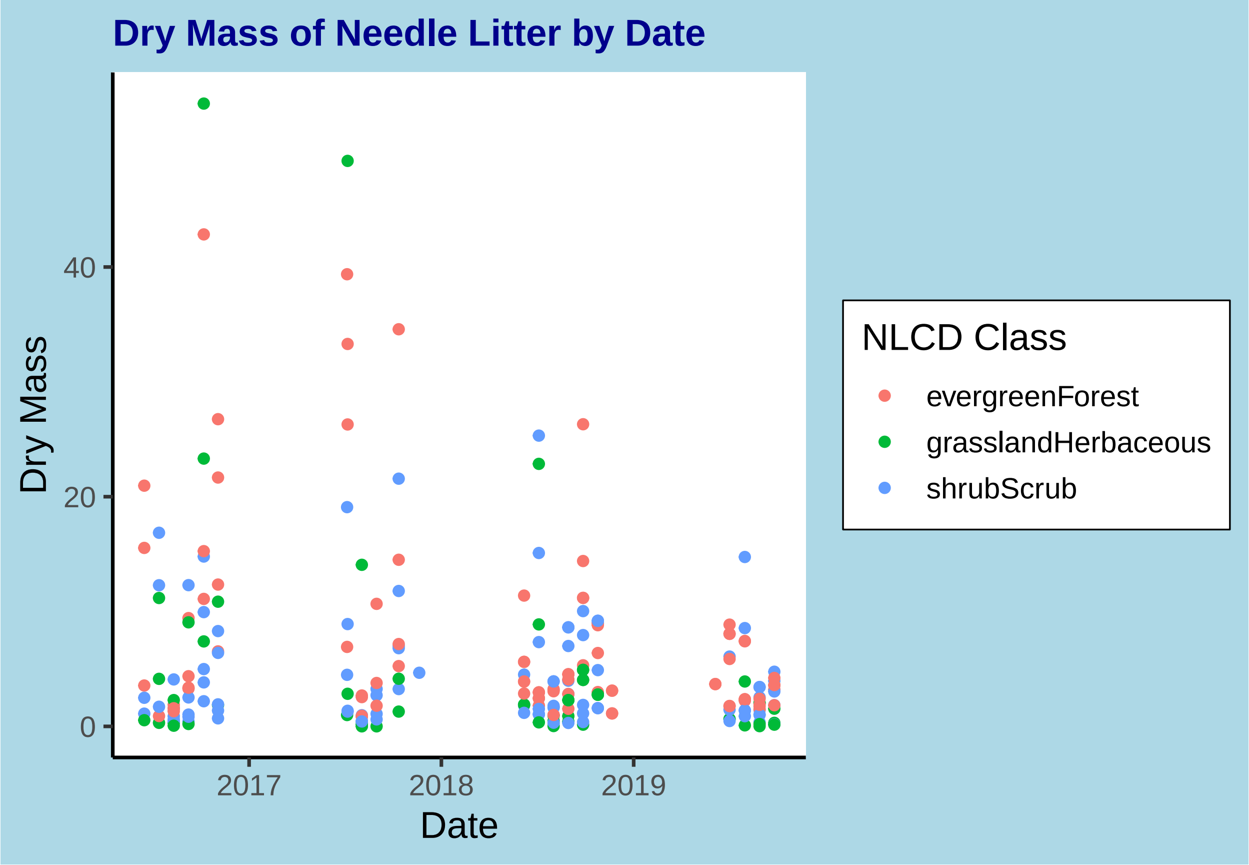
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperature fluctuates a lot through the year, and the temperatues are high during the summer. The total phosphorus in Peter Lake is more during summer time, but the total phosphorus in the Pual Lake is smaller during the summer, and the total phosphorus in Pual Lake is tend to be smaller all season compared with the Peter Lake. For nitrogen, The trends (change) of nitrogen in the two lakes are similar, so do the medians.

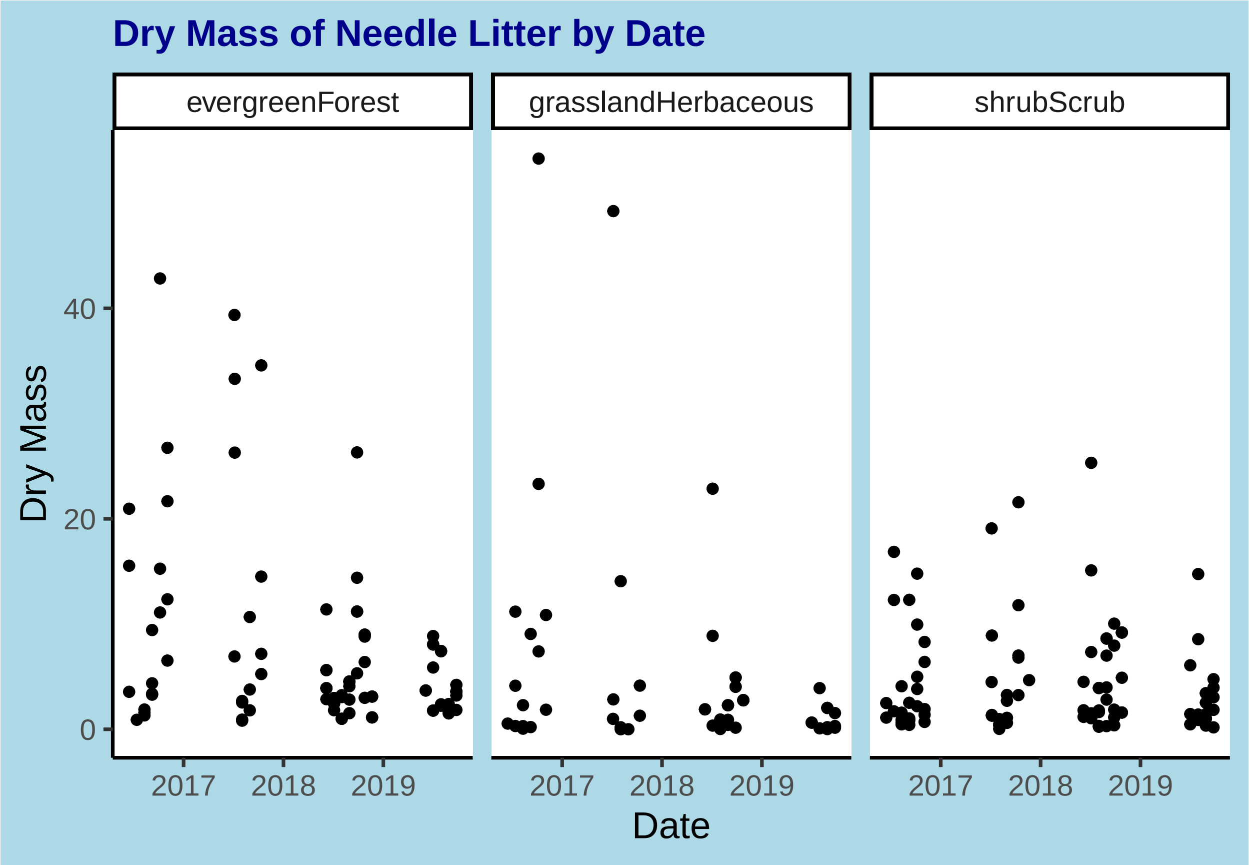
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes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

```
#6
litter_new<-filter(litter,functionalGroup == "Needles")

Dry_Mass<-
  ggplot(litter_new,aes(y=dryMass, x=collectDate, color=nlcdClass))+
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by Date",
       x = "Date",
       y = "Dry Mass",
       color = "NLCD Class")+
  mytheme
```

```
print(Dry_Mass)
```



```
#7
Dry_Mass_new<-
  ggplot(litter_new,aes(y=dryMass, x=collectDate))+
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by Date",
       x = "Date",
       y = "Dry Mass")+
  facet_wrap(facets = vars(nlcdClass),nrow=1)+
  mytheme

print(Dry_Mass_new)
```



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

Answer: I think the plot for question 7 is more effective becasue it is hard to read and differentiate the three different NLCD class when they are combined together in one graph. The second graph shows the three different NLCD class facets side by side, which can help readers clearly see their similarities and difference.