Assignment 5: Data Visualization

Yin-Chia Yang

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)
getwd()
```

```
## [1] "/home/guest/EDA_Spring2024/EDA_forked_Spring2024"
```

```
PeterPaul <- read.csv(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)
NEON <- read.csv("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
```

```
                 stringsAsFactors = TRUE)

#2
#class(PeterPaul$sampledate)
#class(NEON$collectDate)
PeterPaul$sampledate <- as.Date(PeterPaul$sampledate, format = "%Y-%m-%d")
NEON$collectDate <- as.Date(NEON$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
mytheme  <- theme_classic(base_size = 12) +
   theme(
    line = element_line(
      color='magenta',
      linewidth =0.5
      ),
    legend.background = element_rect(
      color='lightgrey',
      fill = NULL
    ),
    legend.title = element_text(
      color='black'
    ),
    legend.position = "top",
    axis.text = element_text(
      color = "black"
      )
  )
theme_set(mytheme)
```

Create graphs

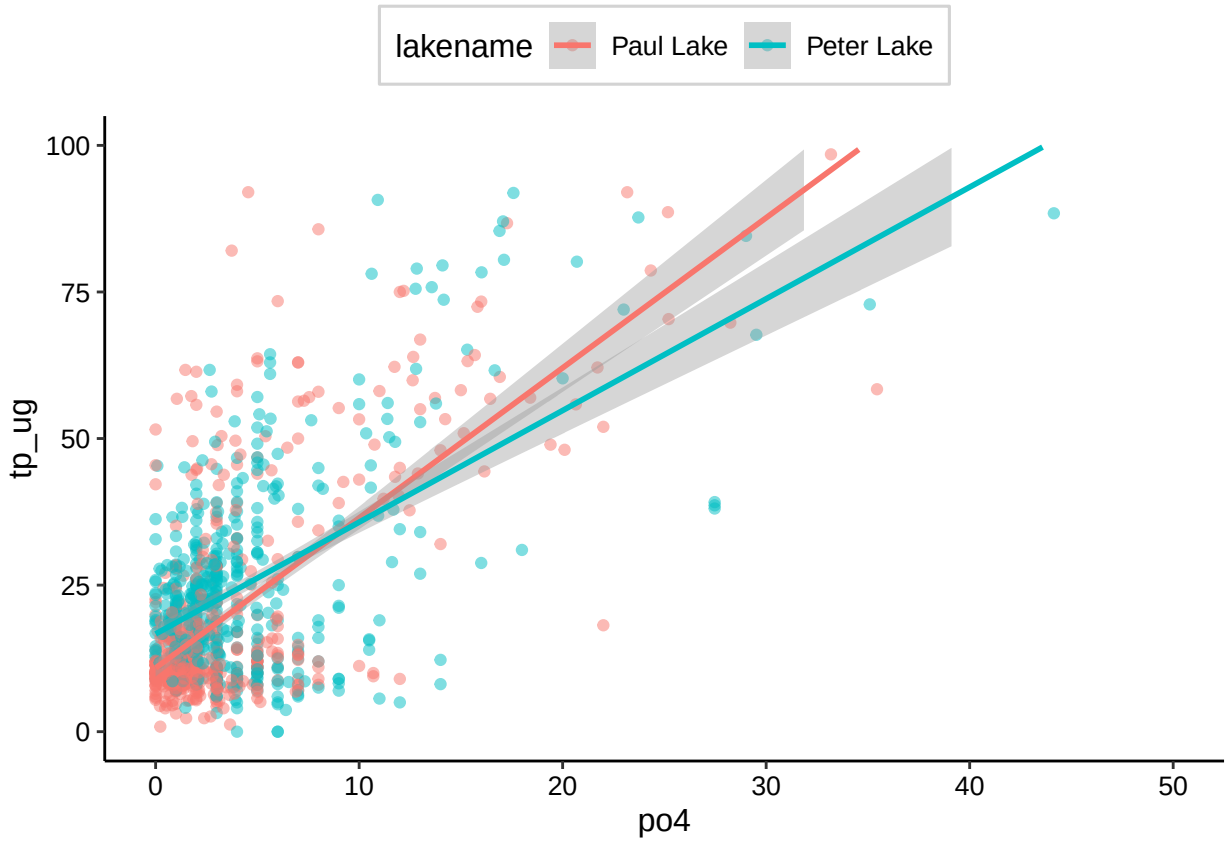
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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
NTL_phos_plot <- PeterPaul %>%
```

```
  ggplot(
     mapping =
       aes(x = po4, y= tp_ug, color = lakename),
  ) +
  geom_point(alpha = 0.5) +
  geom_smooth(method="lm") +
  xlim(0,50) +
  ylim(0,100)

print(NTL_phos_plot)
```

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

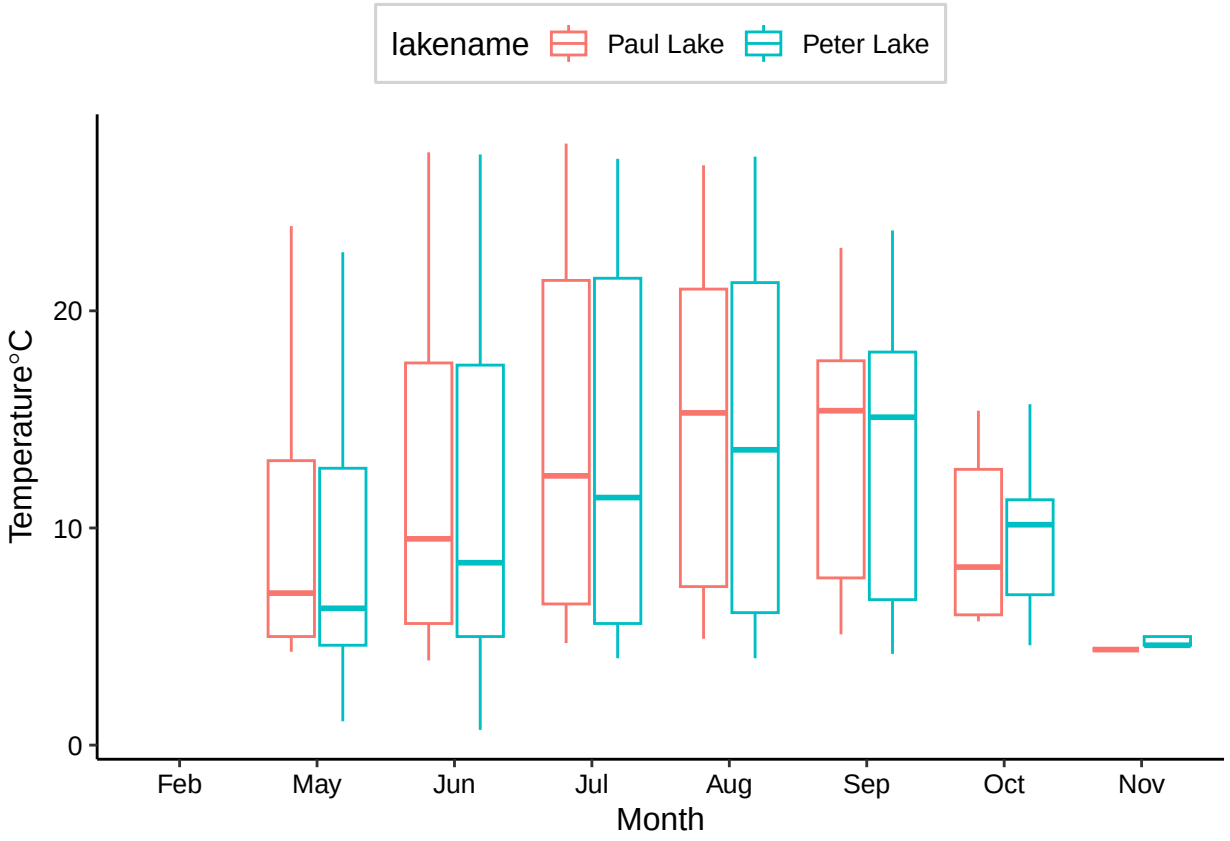


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

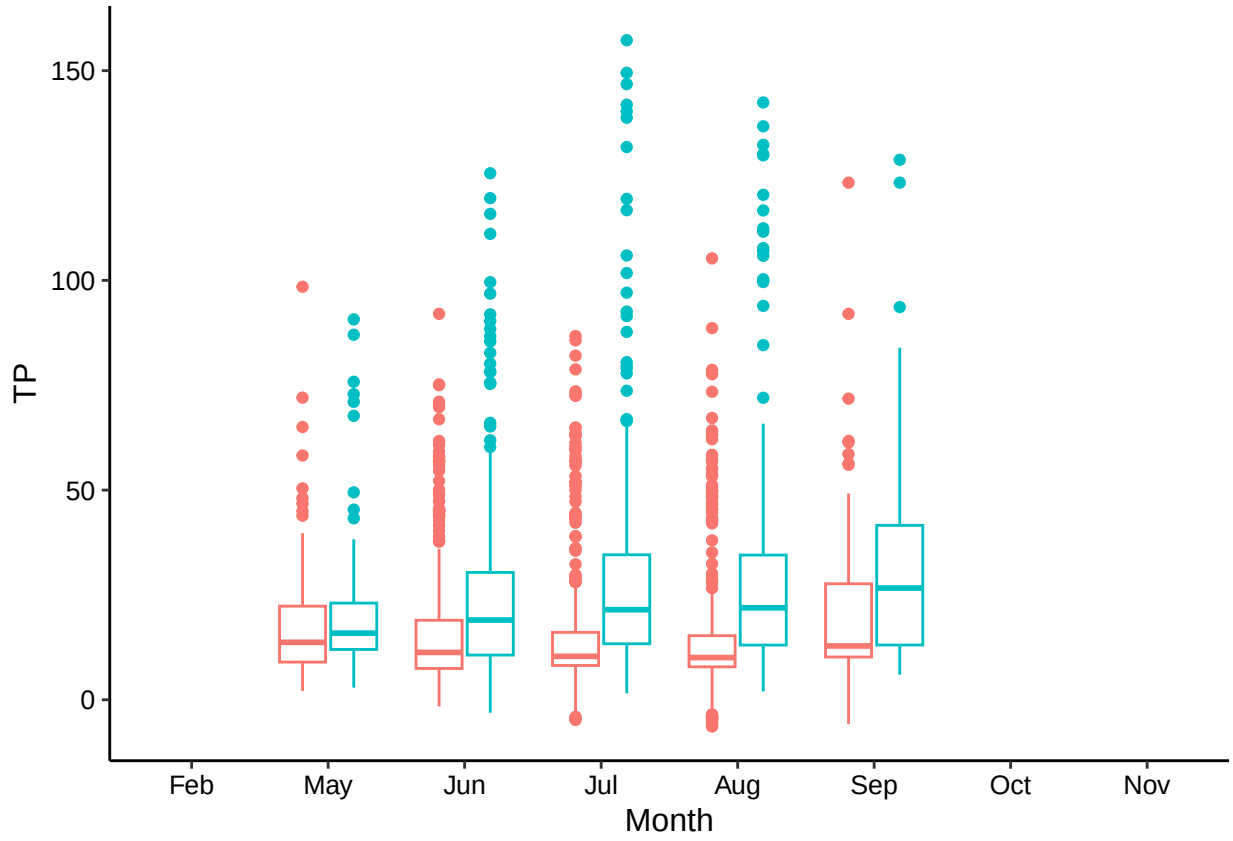
```r
#5
PeterPaul$month_factor <- factor(
  PeterPaul$month,
  levels = 1:12,
  labels = month.abb)

NTL_boxpolt_temp <- ggplot(PeterPaul, mapping = aes(
  x= month_factor, y= temperature_C, color = lakename)) +
  xlab("Month") +
  ylab(expression(paste("Temperature", degree, "C"))) +
  geom_boxplot(show.legend = T)
  print(NTL_boxpolt_temp)
```

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



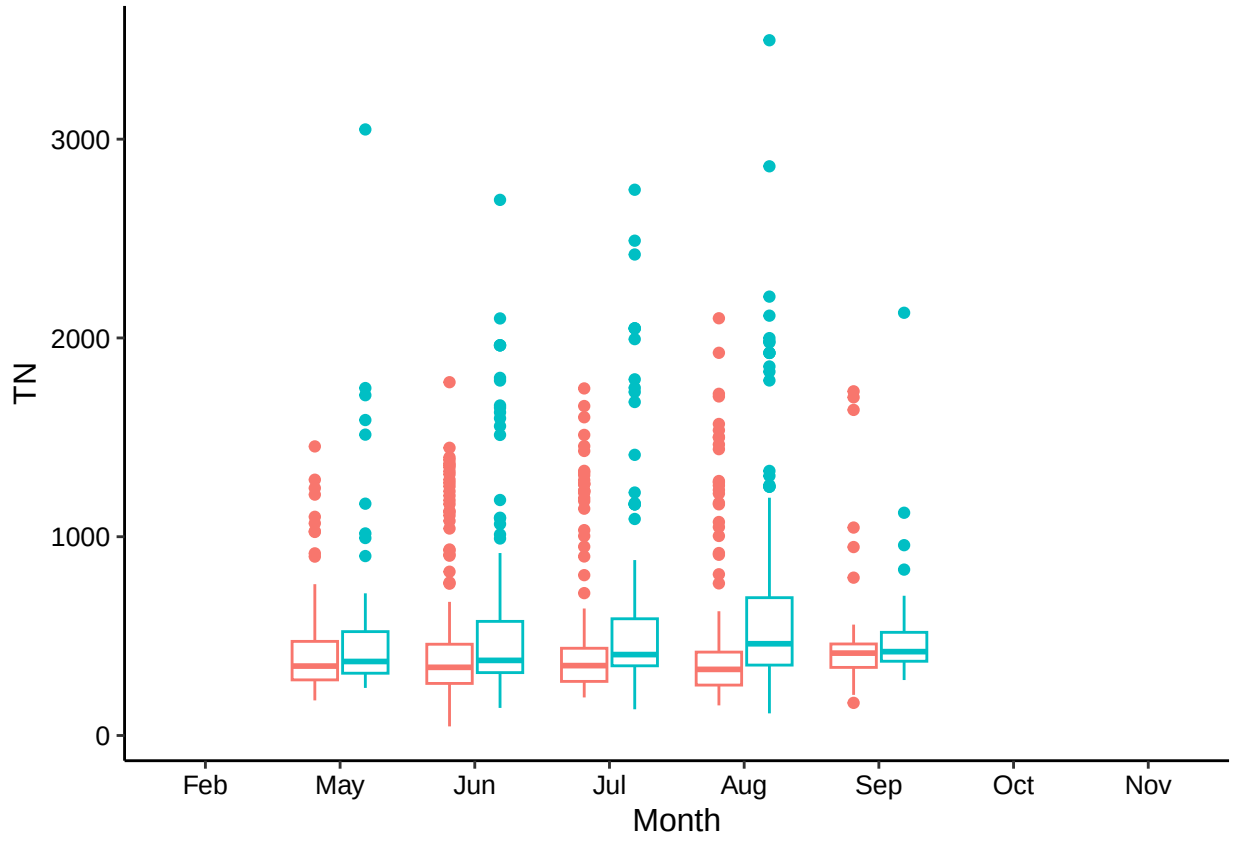

```r
NTL_boxpolt_tp <- ggplot(PeterPaul, mapping = aes(
  x= month_factor, y= tp_ug, color = lakename)) +
  xlab("Month") +
  ylab("TP") +
  geom_boxplot(show.legend = F)
  print(NTL_boxpolt_tp)
```

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




```
NTL_boxpolt_tn <- ggplot(PeterPaul, mapping = aes(
  x= month_factor, y= tn_ug, color = lakename)) +
  xlab("Month") +
  ylab("TN") +
  geom_boxplot(show.legend = F)
  print(NTL_boxpolt_tn)
```

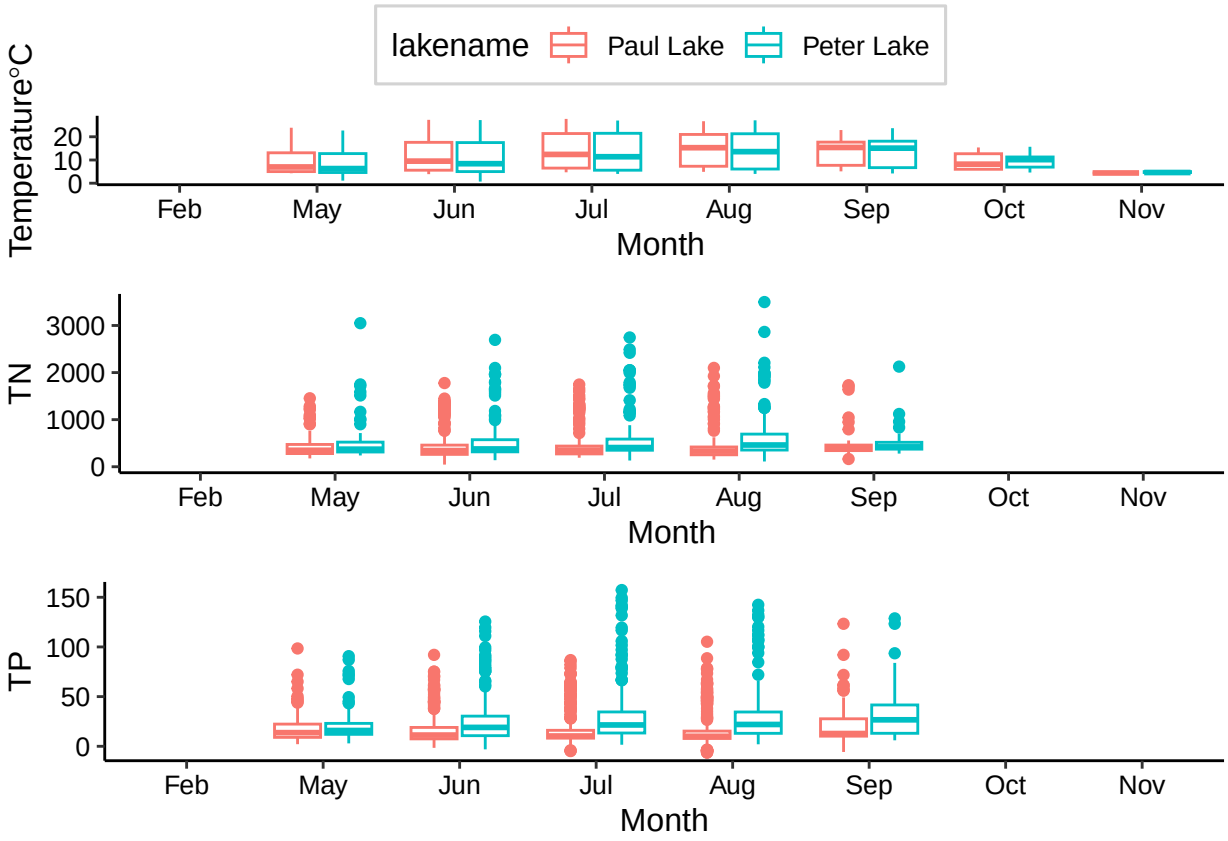
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```




```
library(cowplot)
NTL_cowplot <- plot_grid(NTL_boxpolt_temp, NTL_boxpolt_tn, NTL_boxpolt_tp, nrow = 3, align = "h")
```

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

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

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

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

```
print(NTL_cowplot)
```



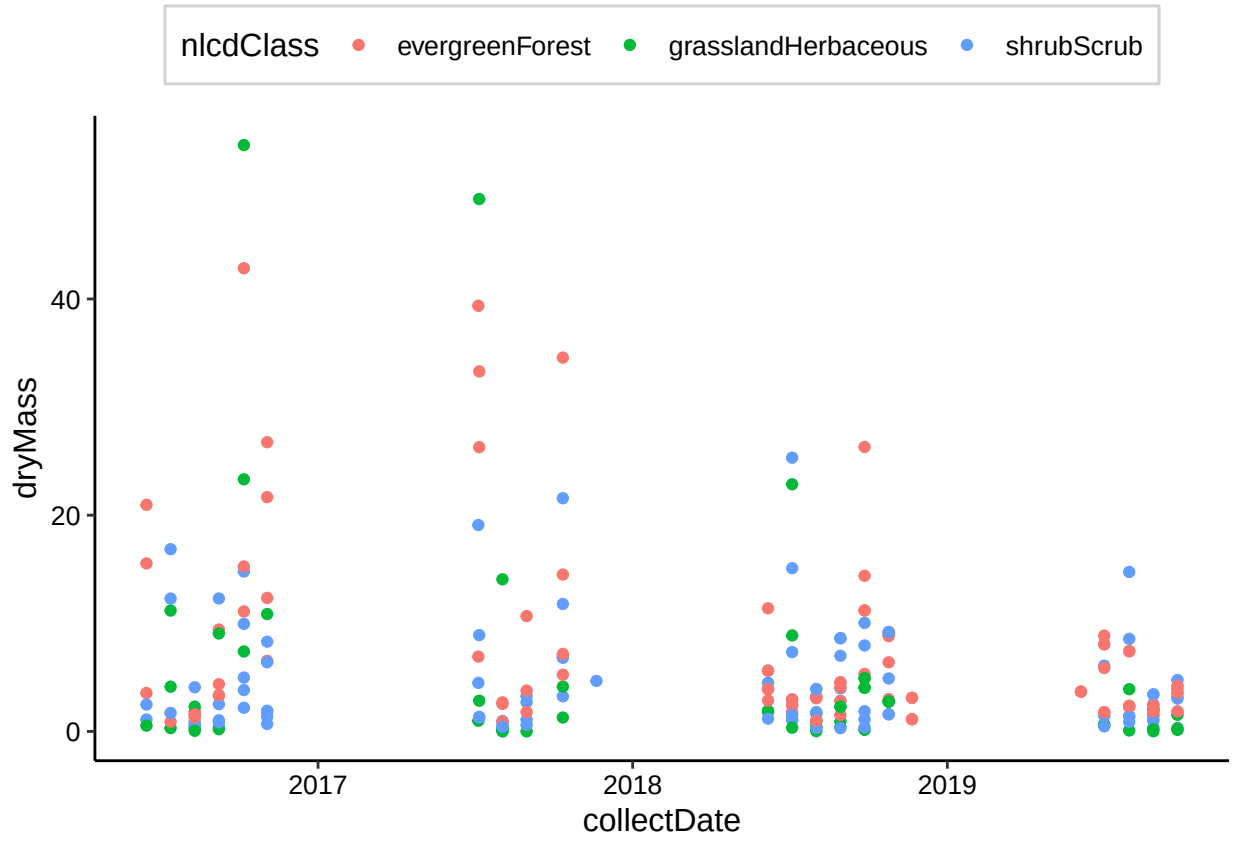
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperature was trending similarly between lakes, with average temperature higher in summer and lower in wither. As for TP and TN, both indicators were generally higher in Peter Lake, and more extreme values were observered in summer.

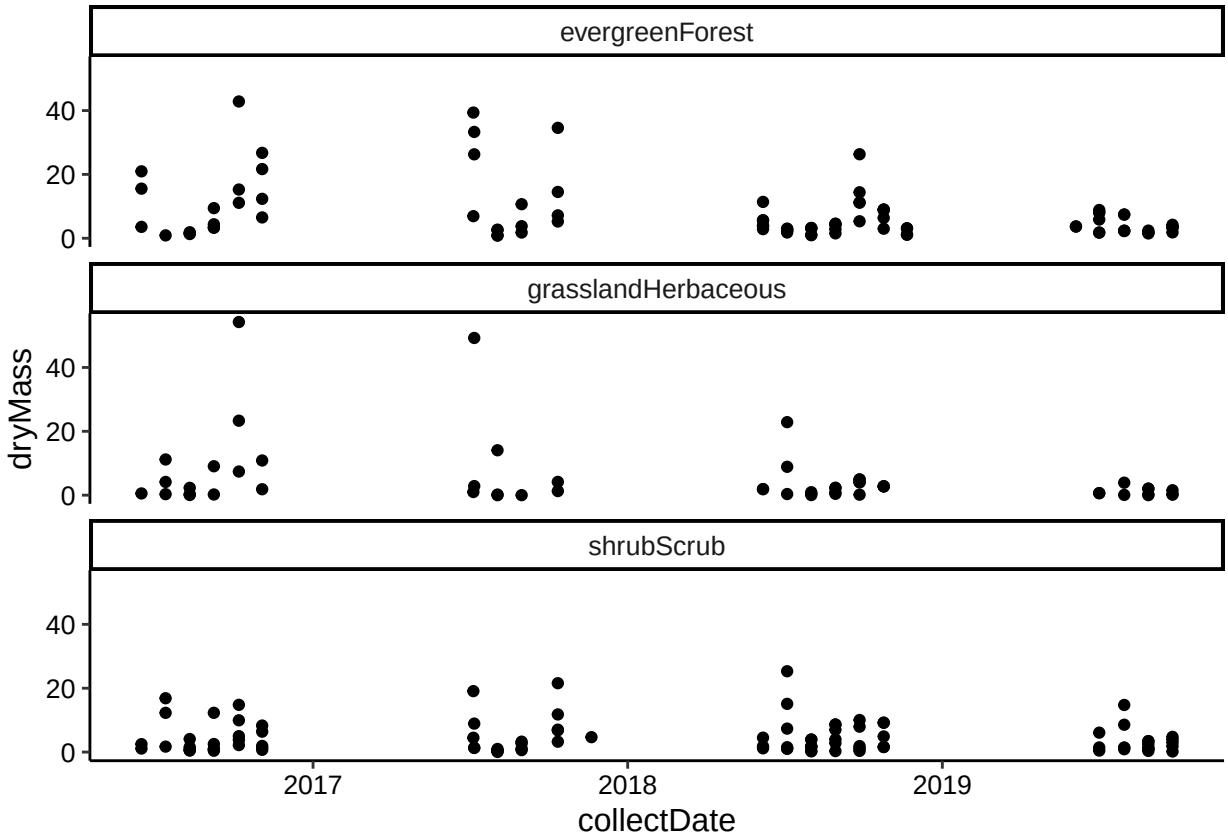
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NEON_needles_dryMass_plot1 <- NEON %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y= dryMass, color = nlcdClass)) +
  geom_point()
print(NEON_needles_dryMass_plot1)
```




```
#7
NEON_needles_dryMass_plot2 <- NEON %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y= dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(NEON_needles_dryMass_plot2)
```



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

Answer: I think plot 6 (NEON_needles_dryMass_polt1) is more effective because I was able to identify and compare between the three by NLCD classes, and I think showing them with color is more visual attractive.