Assignment 5: Data Visualization

Ana Andino

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.

```
#1

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
getwd()
```

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

```
#2 Read processed data files

PeterPaul.processed <-
  read.csv(
  here("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

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


class(Litter.processed$collectDate)
```

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

```
class(PeterPaul.processed$sampledate)
```

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

```
#change date format

Litter.processed$collectDate <- as.Date(Litter.processed$collectDate,
                                        format("%Y-%m-%d"))
PeterPaul.processed$sampledate <- as.Date(PeterPaul.processed$sampledate,
                                          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

my_theme <- theme_minimal(base_size = 12) +
  theme(axis.text = element_text(color = "black", face = "bold"),
        plot.title = element_text(hjust = 0.5, color = "grey", size = 14, face = "bold"),
        legend.position = "bottom")
```

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.

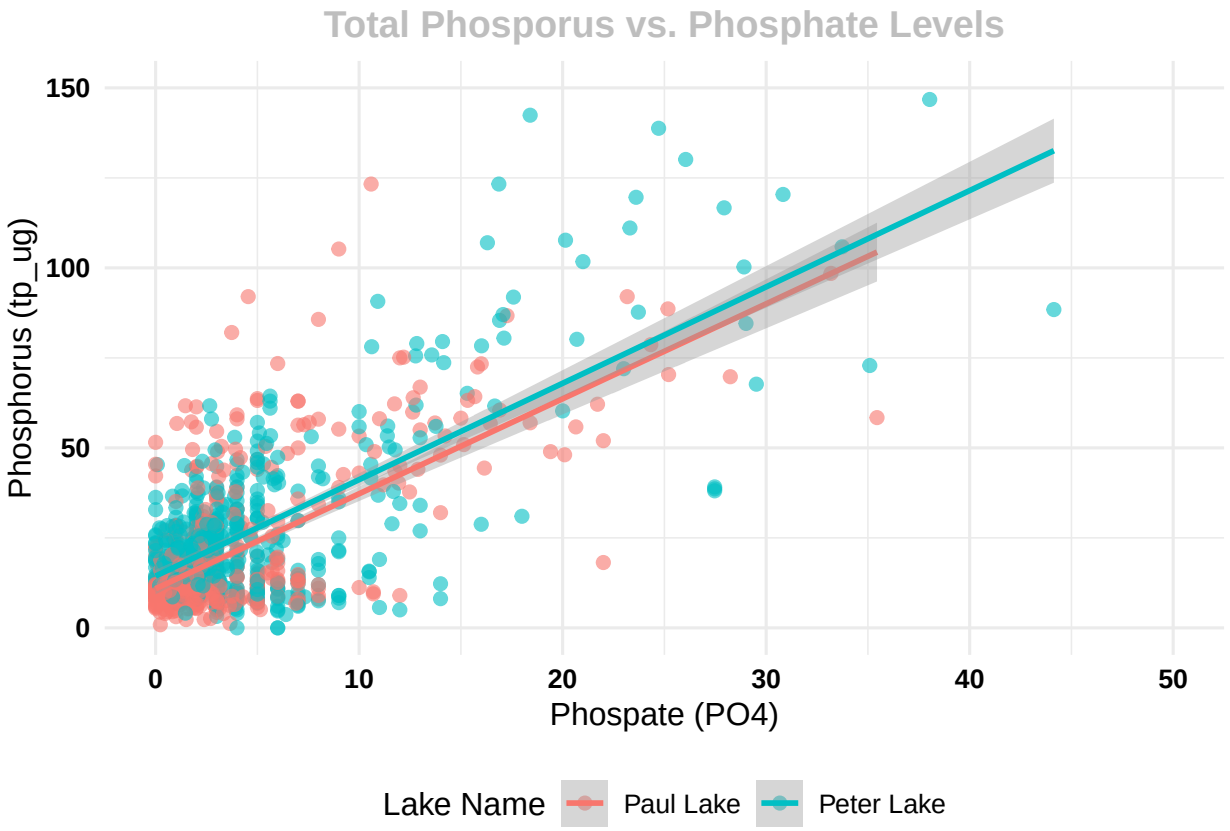
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
library(ggplot2)
library(dplyr)

PeterPaul.processed %>%
  ggplot(aes(x= po4, y=tp_ug, color = lakename)) +
          geom_point(alpha = 0.6, size = 2) +  #Added points to catch outliers
           geom_smooth(method = "lm", se = TRUE) + #Line with CI
  xlim(0, 50) + #added axis limits according to data on x and y axis
  ylim(0, 150) +
  labs(title = "Total Phosporus vs. Phosphate Levels",
       x = "Phospate (PO4)",
       y= "Phosphorus (tp_ug)",
       color = "Lake Name") +
  my_theme #Applied my custom theme
```

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

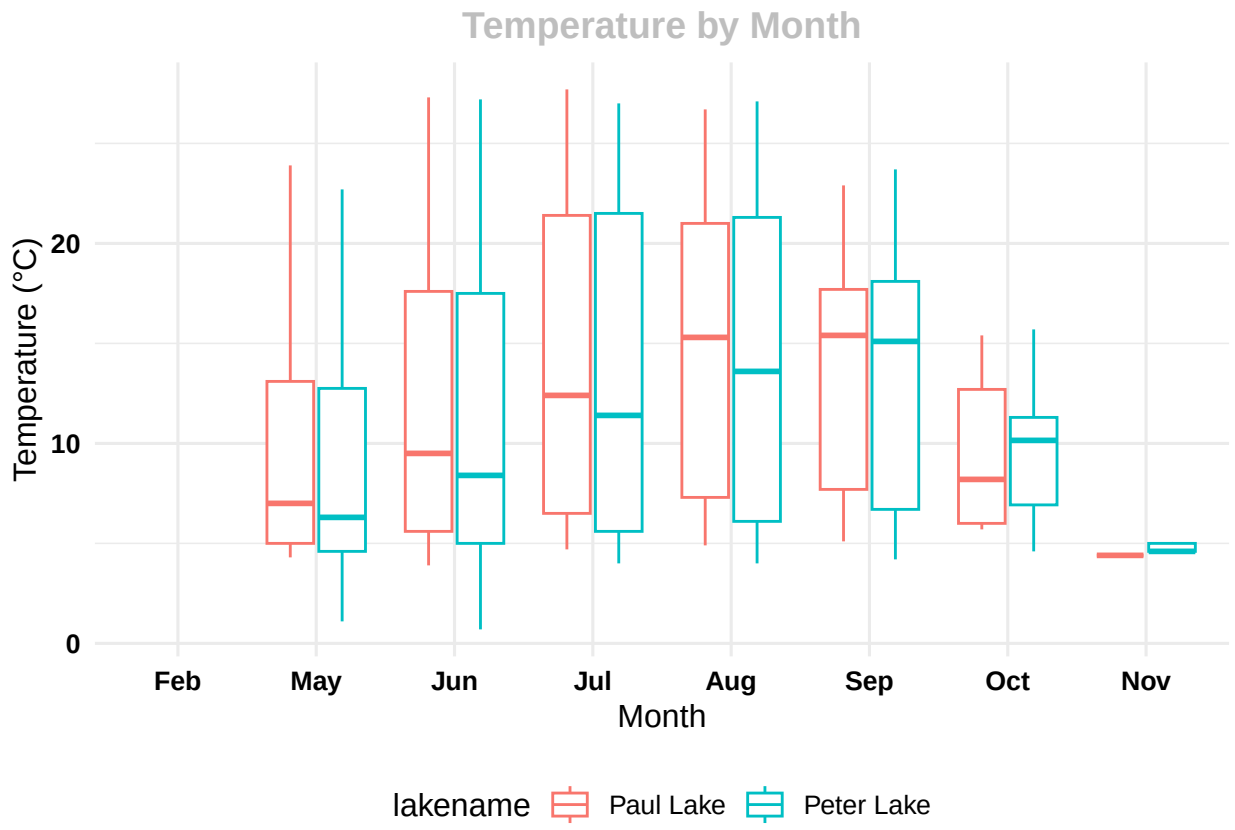


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

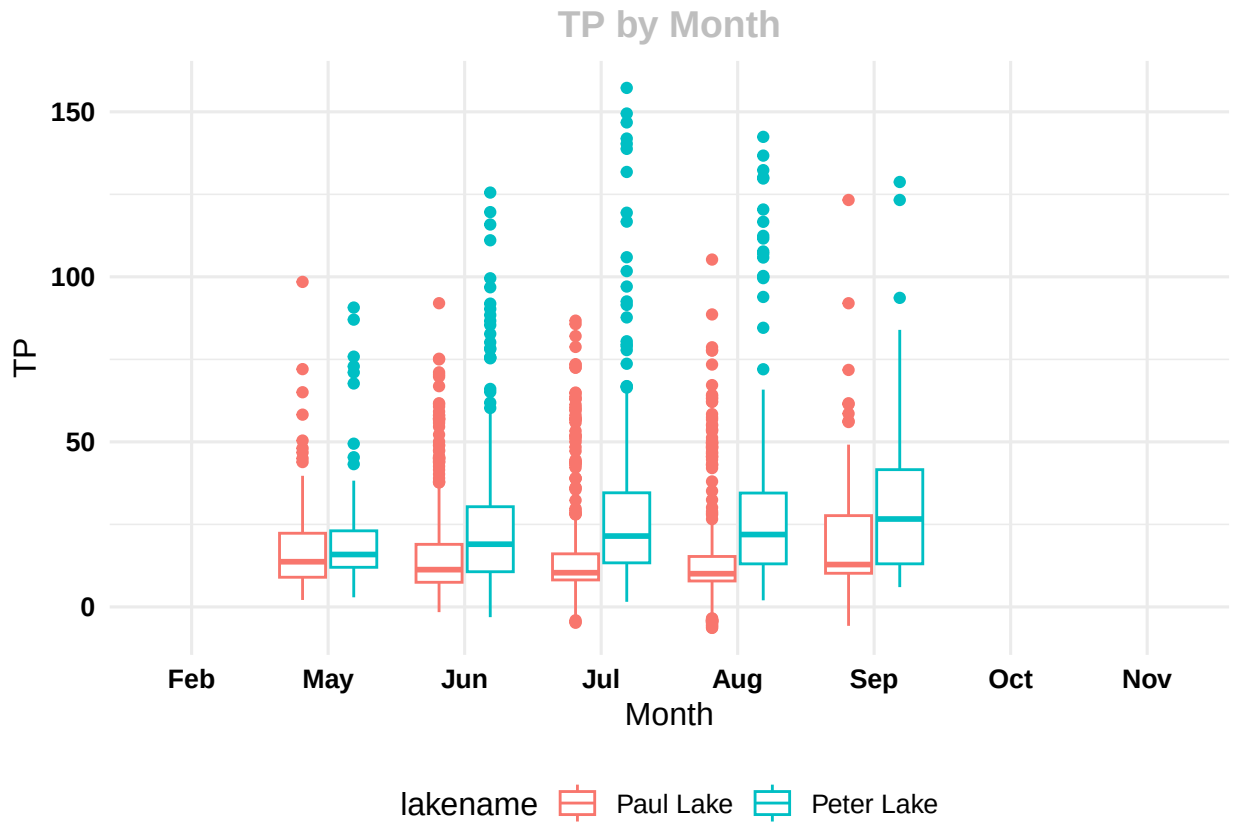
```
#5
#Temperature Boxplot
#Convert month to a factor with 12 levels, labelled with month names



temperature_plot <- PeterPaul.processed %>%
  ggplot(aes(x = factor(month, levels = 1:12, labels = month.abb),
             y = temperature_C,
             color = lakename)) +  # Close aes before adding layers
  geom_boxplot() +
  labs(title = "Temperature by Month",
       x = "Month",
       y = "Temperature (°C)") +
  my_theme

print(temperature_plot)
```

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



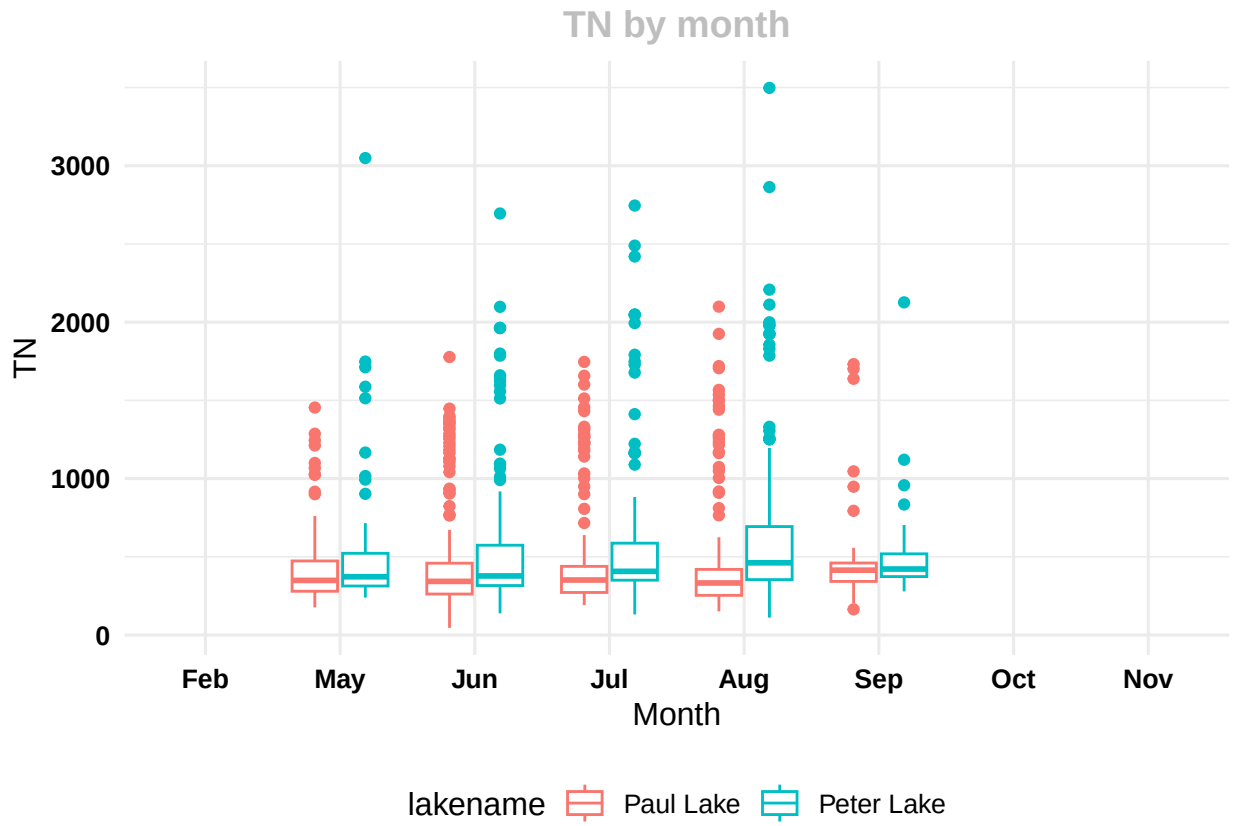

```
#TP Plot
TP_plot <- PeterPaul.processed %>%
  ggplot(aes( x = factor(month, levels = 1:12, labels = month.abb),
              y = tp_ug, color = lakename))+
  geom_boxplot()+
  labs(title = "TP by Month",
       x= "Month",
       y= "TP")+
  my_theme
print(TP_plot)
```

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




```
#TN Plot
TN_plot <- PeterPaul.processed %>%
  ggplot(aes(x= factor(month, levels = 1:12, labels = month.abb),
                       y= tn_ug,
                       color = lakename)) +
           geom_boxplot()+
           labs(title = "TN by month", x = "Month",
                y = "TN")+
           my_theme
print(TN_plot)
```

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




```
#install.packages("cowplot")
#install.packages("ggplot2")
#install.packages("dplyr")
library(ggplot2)
library(cowplot)
library(dplyr)


my_theme2 <- theme_minimal(base_size = 10) +
  theme(
   axis.title.x = element_blank(),
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),  # Remove x-axis title for upper plots
    legend.position = "none"
  )

my_theme3 <- theme_minimal(base_size = 10) +
  theme(
    legend.position = "right"
  )

# Apply the modified custom theme to the individual plots
temperature_plot <- temperature_plot + my_theme2
TP_plot <- TP_plot + my_theme2
TN_plot <- TN_plot + my_theme3
```

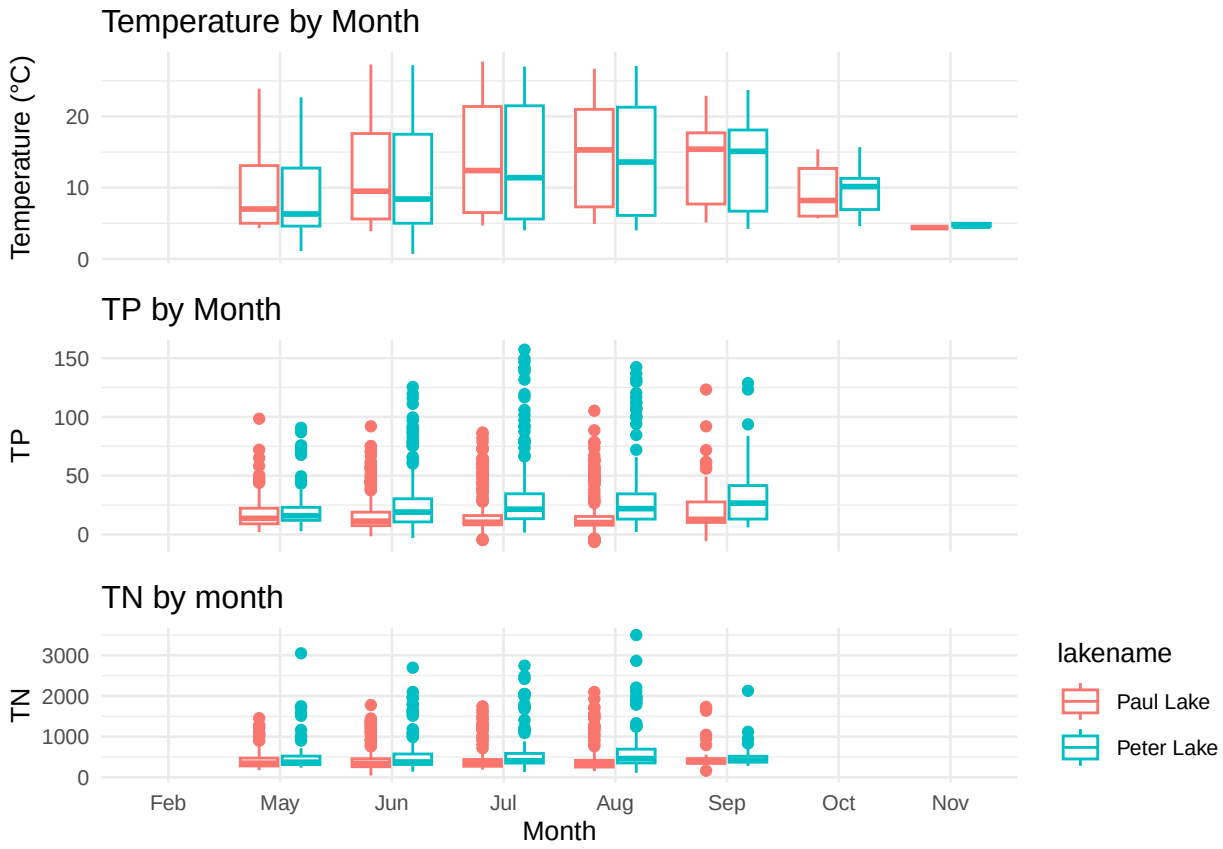
```r
# Combine the plots
combined.plots <- plot_grid(
  temperature_plot, TP_plot, TN_plot,
  ncol = 1,          # Arrange plots in 1 column
  align = "v",        # Align vertically
   rel_heights = c(1, 1, 1) # Equal heights for all plots
)
```

```
## 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()‘).
```

```r
print(combined.plots)
```



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

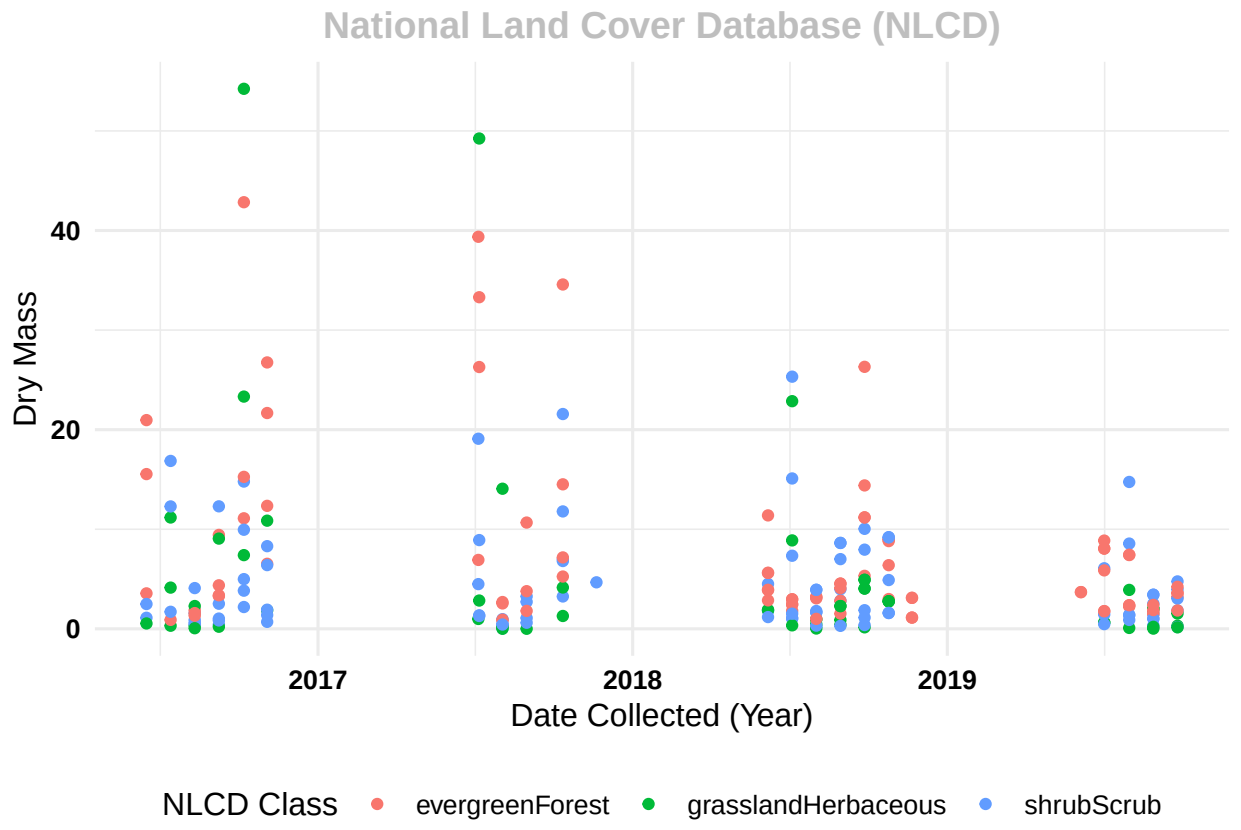
Answer: There is a clear seasonal trend where temperatures rise from May to August and then begin to decline in September and both lakes have similar trends in that sense. With regards to TP, there is a variation across months and higher concentrations seem to be appear in summer

months - where Peter Lake shows more this trend. Peter Lake also has more variations and displays more outliers. When analyzing TN, there is a variability across months but no consitent season patters. Peter Lake has overall higher rates of TN and bigger spikes as wel. It also looks like, TP variation follows temperature by month (seasons) more clearly than TN does, especially when looking at Peter Lake. Overall, Peter Lake seems to have more nutrient fluctuations than Paul Lake.

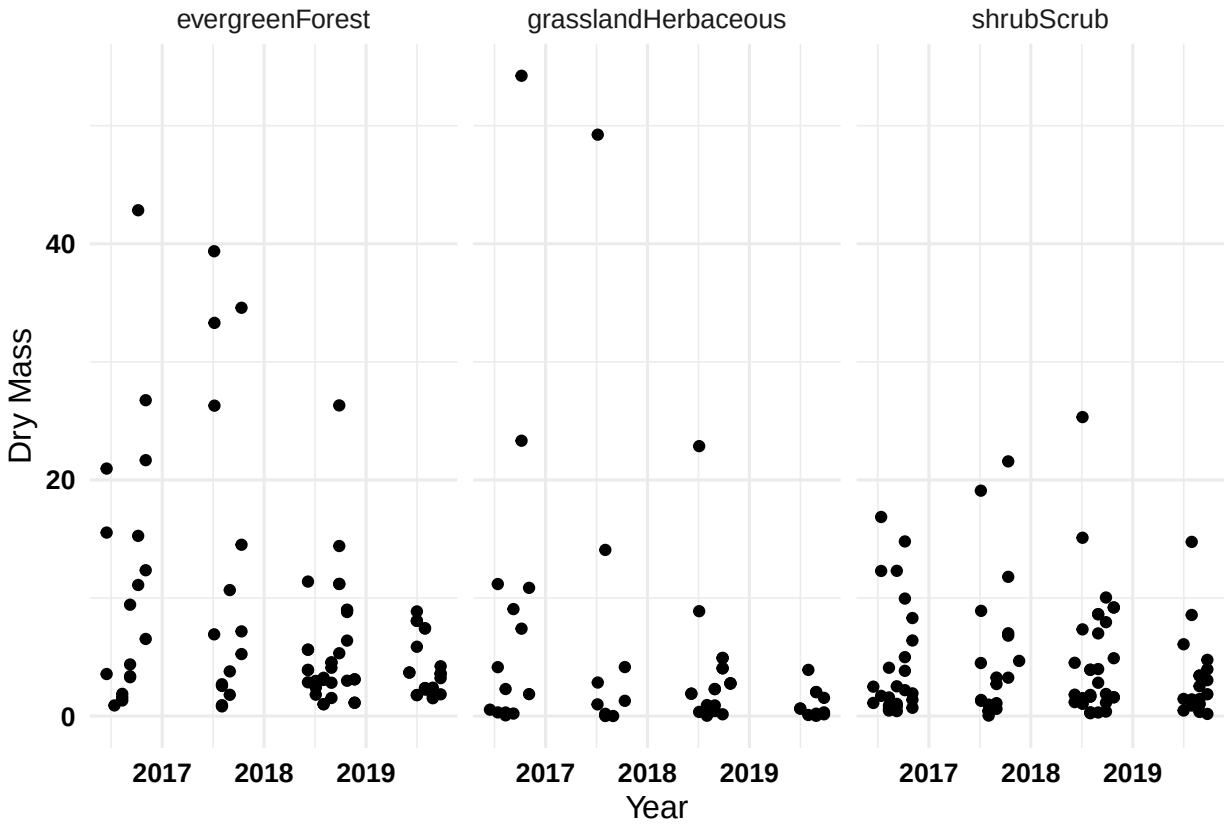
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
Litter.data <- Litter.processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y= dryMass, color = nlcdClass)) +
    geom_point() +
    labs(title = "National Land Cover Database (NLCD)",
         x = "Date Collected (Year)",
         y = "Dry Mass",
         color = "NLCD Class") +
  my_theme
 print(Litter.data)
```



```r
#7

Litter.data.facets <- Litter.processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y= dryMass)) +
    geom_point() +
   facet_wrap(
facets = vars(nlcdClass),
nrow=1,ncol=3) +
    labs(tile = "National Land Cover Database (NLCD)",
         x = "Year",
         y = "Dry Mass",
         Legend = "NLCD Class") +
  my_theme

 print(Litter.data.facets)
```



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

Answer: In this case, 7 is better for analysis. By Faceting you are allowed to compare trends across groups without the visual noise that color can introduce which happened in this case with three different groups. Number 7 allows viewers to focus on each class independently while still being able to see differences and similarities across data. However, it will depend on your goal. If your gooal was to look at trends trends across time regardless of the NLCD class, the color separation might be better.