Assignment 5: Data Visualization

Danielle Butler

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
#Import basic libraries
library(tidyverse);library(lubridate);library(here)
```

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

```r
library(ggridges)

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

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

```r
here()
```

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

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

Niwot.Ridge.Processed <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)

#2 Make sure R is reading dates as date format; if not change the format to date.
class(PeterPaul.Chem$sampledate)
```

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

```r
class(Niwot.Ridge.Processed$collectDate)
```

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

```r
PeterPaul.Chem$sampledate <- ymd(PeterPaul.Chem$sampledate)
Niwot.Ridge.Processed$collectDate <- ymd(Niwot.Ridge.Processed$collectDate)
```

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

```r
#3
library(ggplot2)

my_theme <- theme_classic() +
  theme(
    plot.background = element_rect(fill = "lightgray"),
    plot.title = element_text(face = "bold", size = 16),
    legend.position = "top",  # Legend at the bottom
  )

# Set the custom theme as default
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.

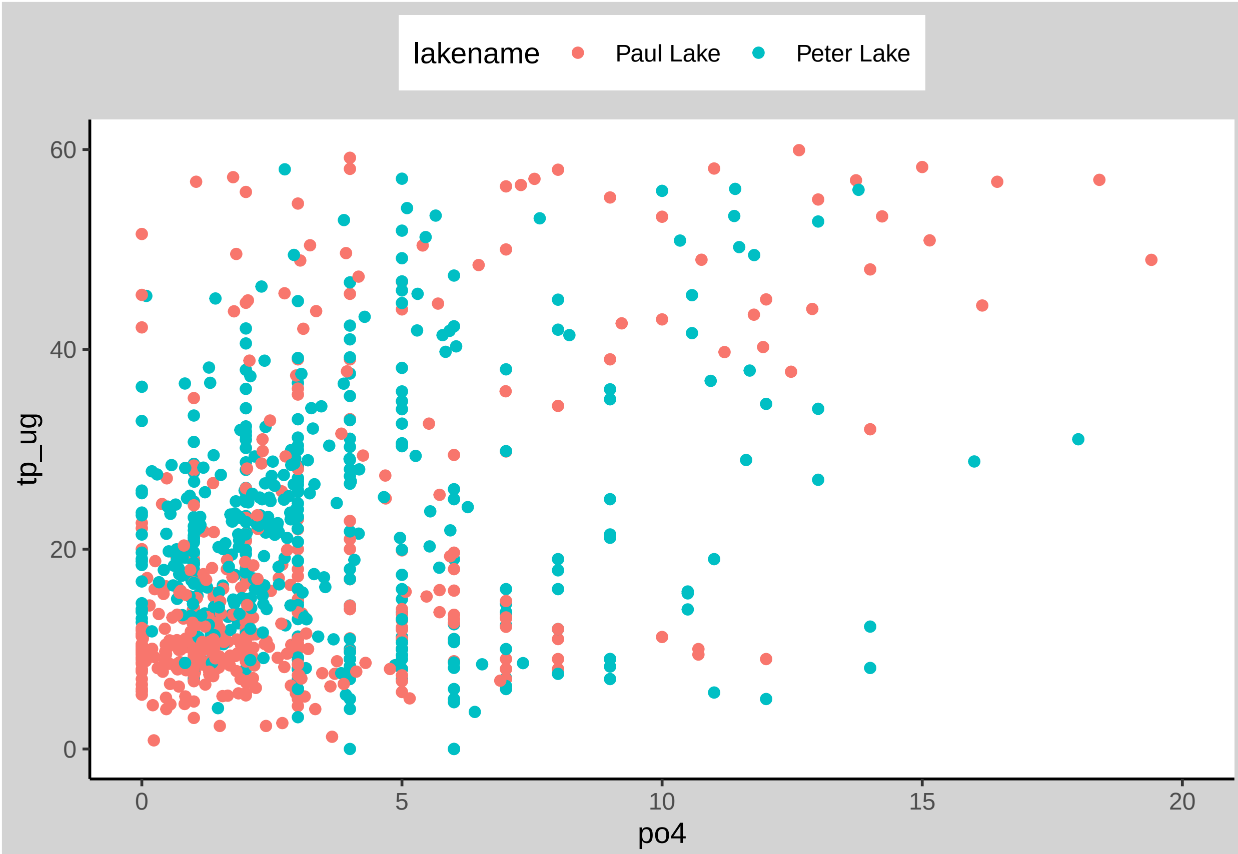
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4
x_limits <- c(0, 20)
y_limits <- c(0, 60)

O4plot <- ggplot(PeterPaul.Chem) +
  geom_point(aes(x=po4, y=tp_ug,color=lakename),method="lm") +
  xlim(x_limits) +
  ylim(y_limits)
```

```
## Warning in geom_point(aes(x = po4, y = tp_ug, color = lakename), method =
## "lm"): Ignoring unknown parameters: ‘method‘
```

```r
print(O4plot)
```

```
## Warning: Removed 22028 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

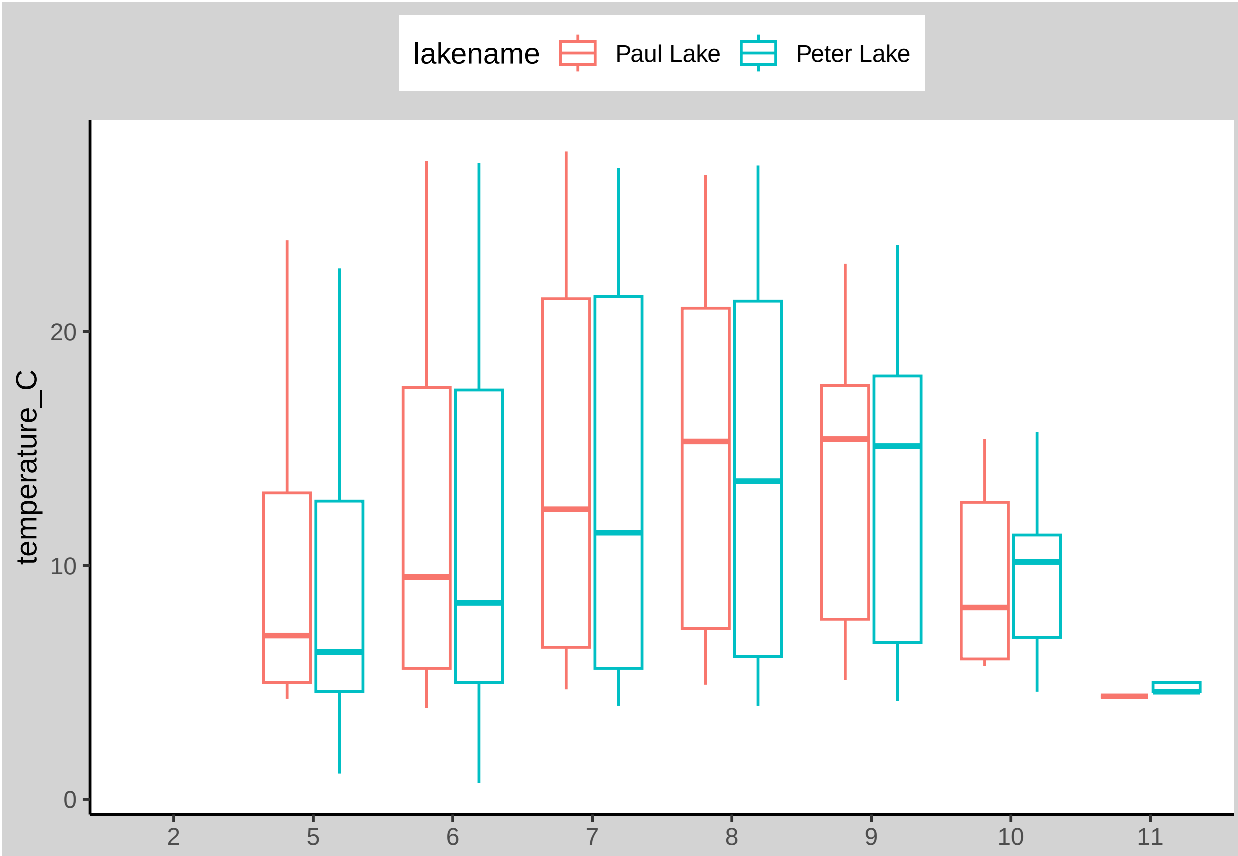


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

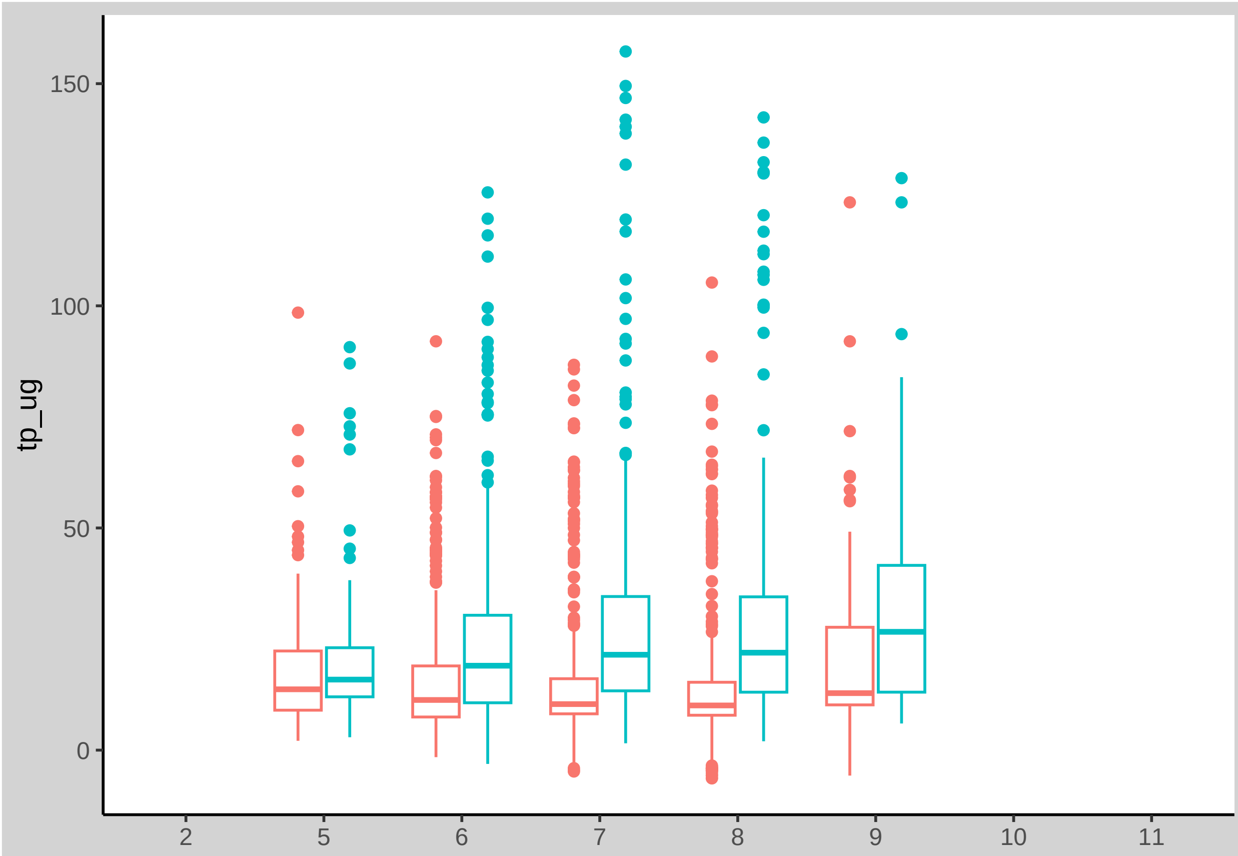
```
#5
#a temperature
temp_plot <- ggplot(PeterPaul.Chem) +
  geom_boxplot(aes(x = factor(month), y = temperature_C, color = lakename)) +
  theme(axis.title.x = element_blank())
print(temp_plot)
```

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



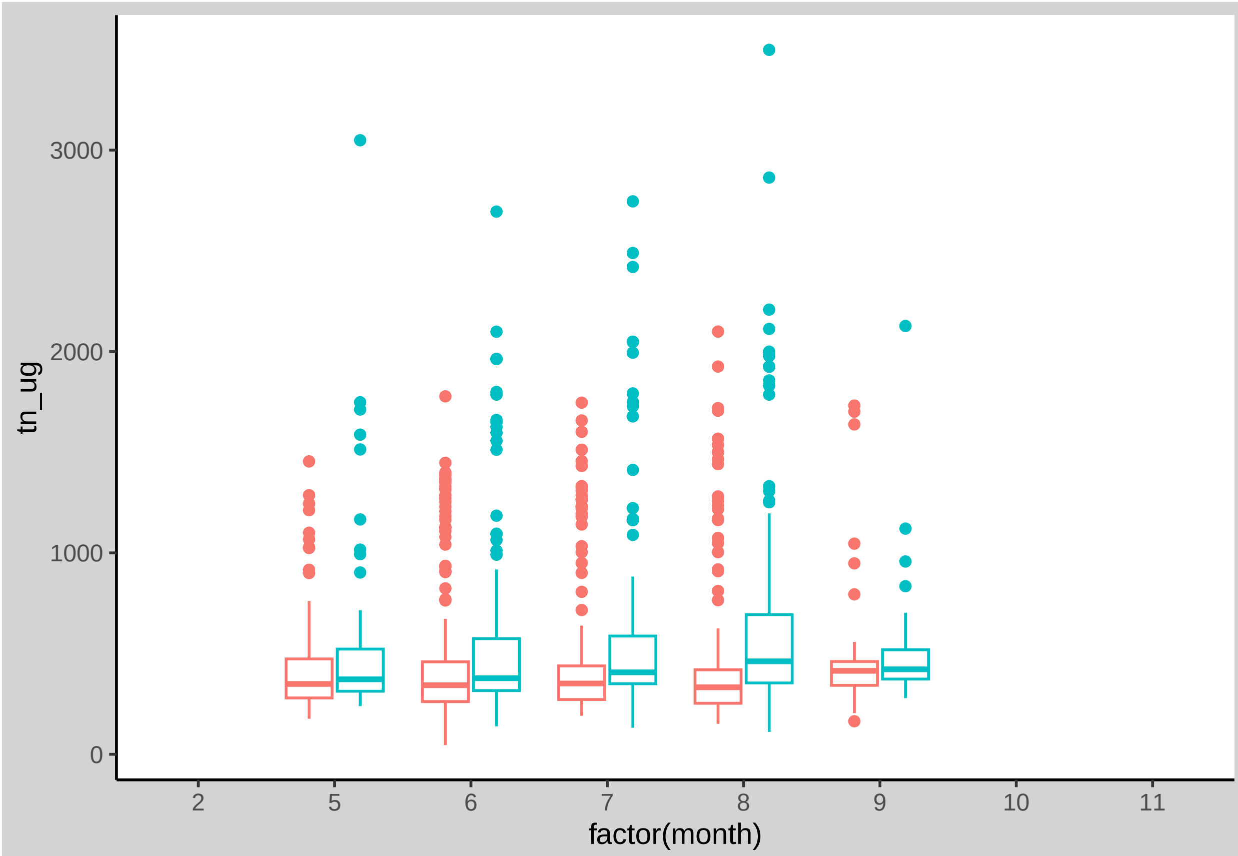

```r
#b temperature
tp_plot <- ggplot(PeterPaul.Chem) +
  geom_boxplot(aes(x = factor(month), y = tp_ug, color = lakename)) +
  theme(legend.position = "none",
    axis.title.x = element_blank())

print(tp_plot)
```

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




```
#c temperature
tn_plot <- ggplot(PeterPaul.Chem) +
  geom_boxplot(aes(x = factor(month), y = tn_ug, color = lakename)) +
 theme(legend.position = "none")
print(tn_plot)
```

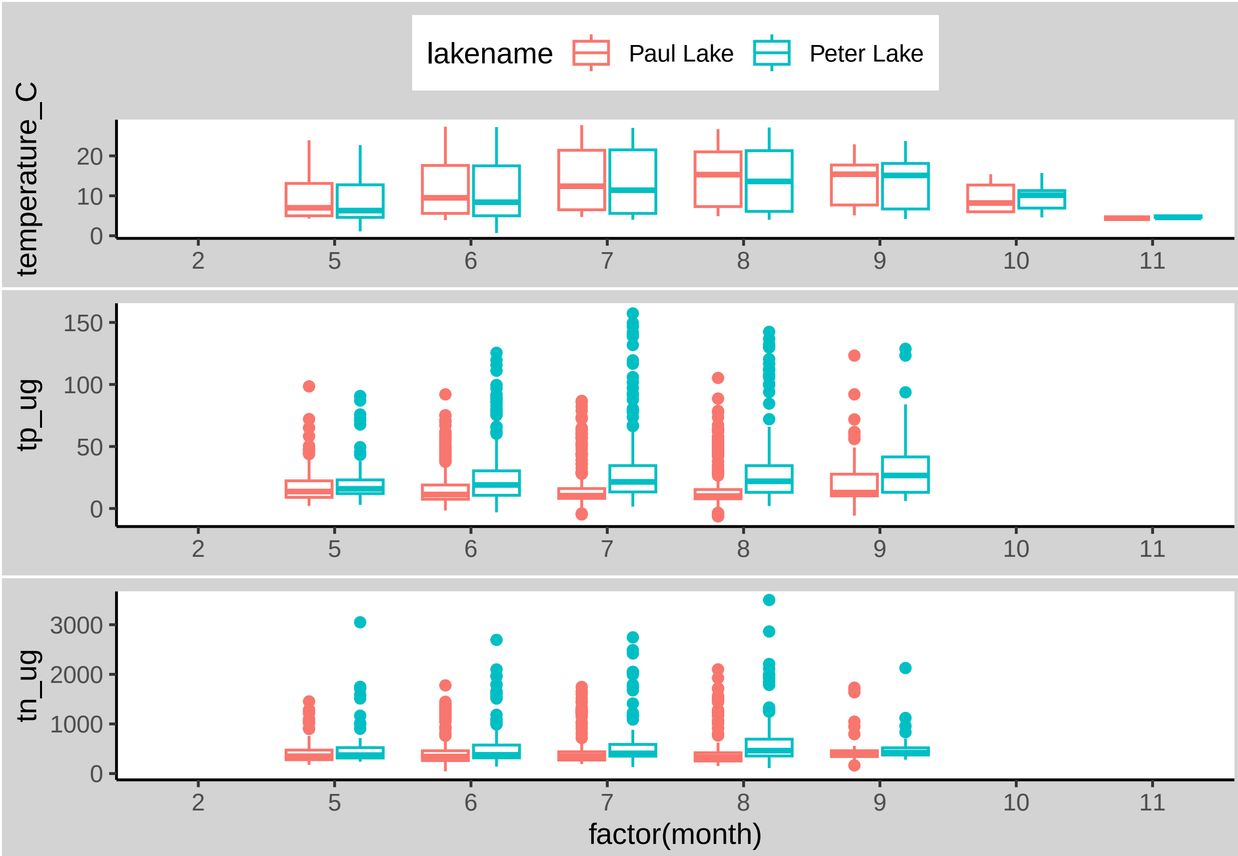
```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```r
#final
final_plot <- plot_grid(
  temp_plot, tp_plot, tn_plot,
  ncol = 1, 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()`).
```

```r
print(final_plot)
```



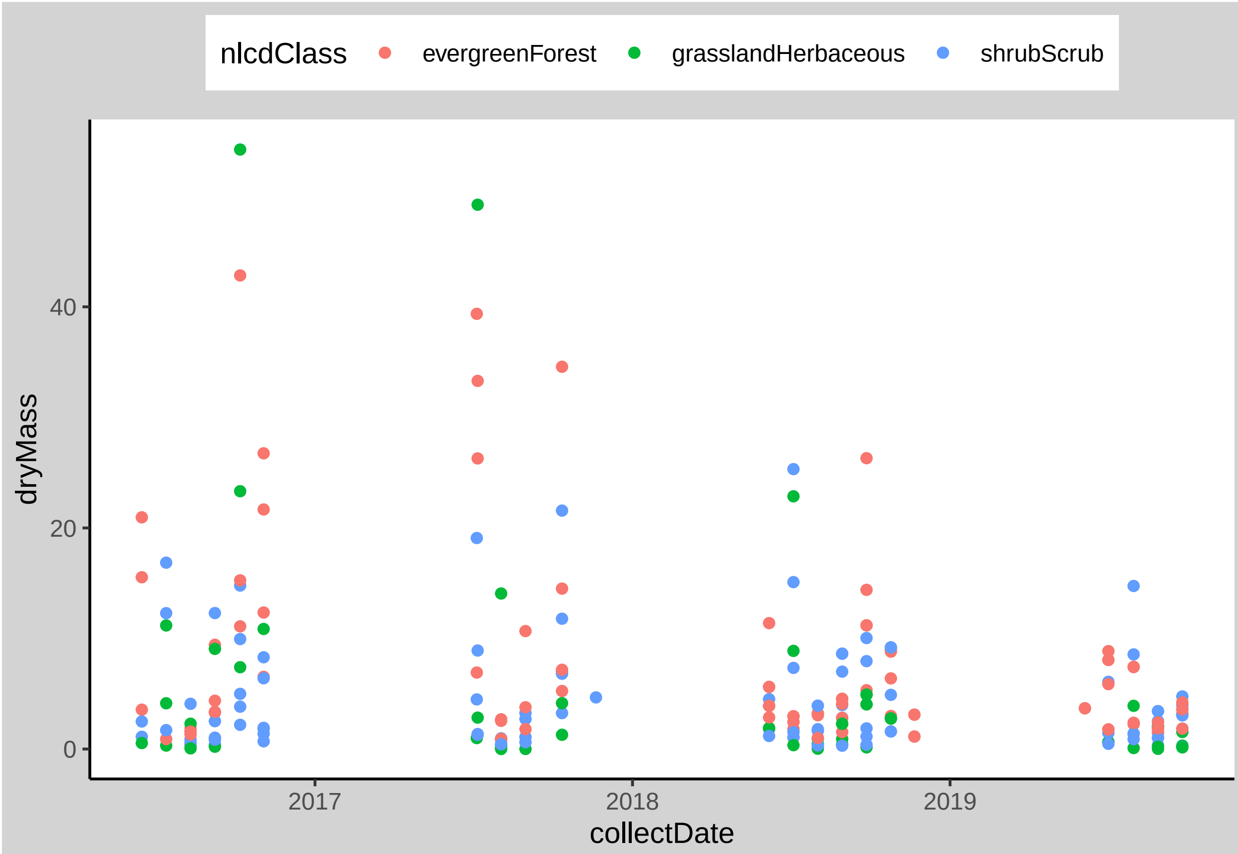
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Total Phosphorus is higher in Peter Lake than Paul Lake and increases in the summer months. Temperature is similar in both lakes, but rises in the summer months, including September with residual heat, most likely. And finally, Nitrogen is also higher in Peter Lake than Paul Lake.

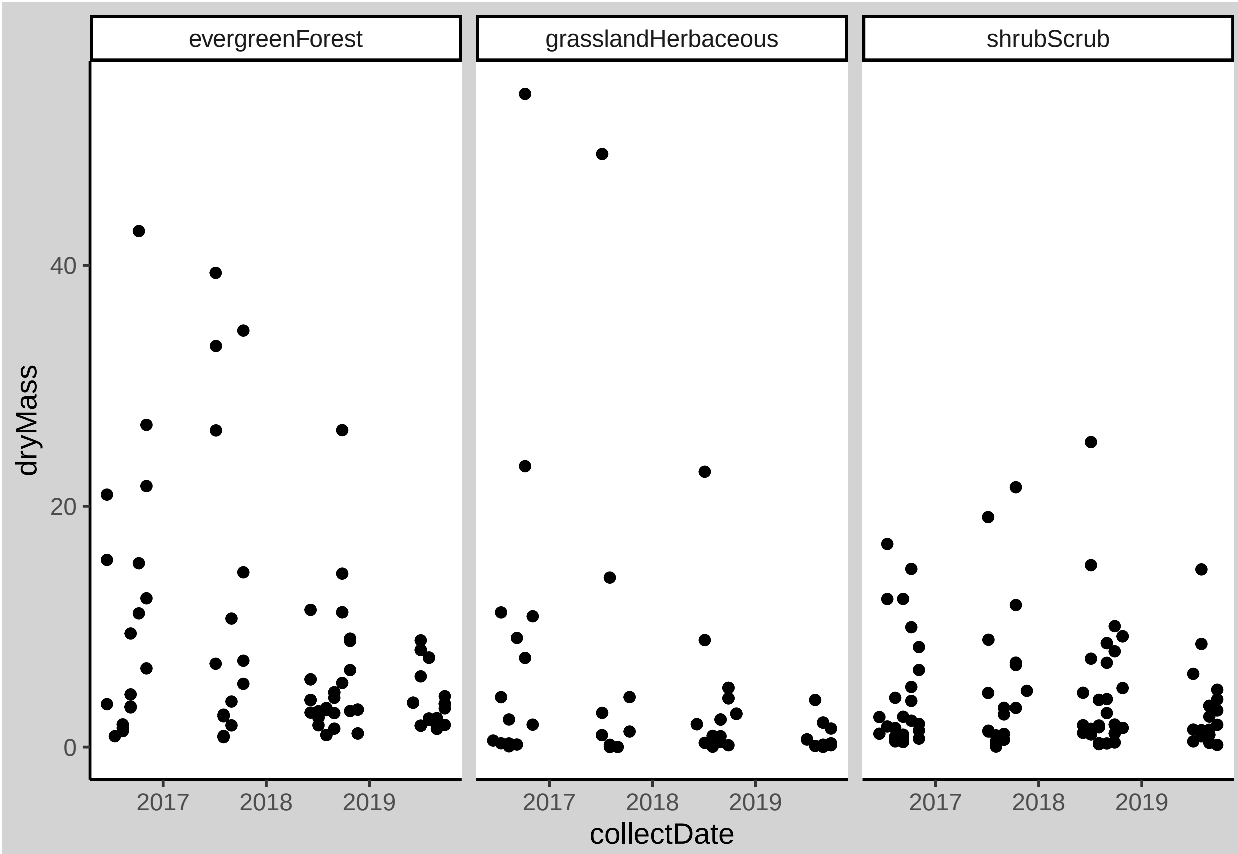
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

Niwot.Ridge.Processed.Needles <-
  ggplot(subset(Niwot.Ridge.Processed, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass,color=nlcdClass)) +
  geom_point()

  print(Niwot.Ridge.Processed.Needles)
```




```r
#7
Niwot.Ridge.Needles.Facets <-
  ggplot(subset(Niwot.Ridge.Processed, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass)) +
  geom_point()+
  facet_wrap(vars(nlcdClass),ncol=3)

  print(Niwot.Ridge.Needles.Facets)
```



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

Answer: I think 7 is more effective because it is easier to see for each class the trends of the dry mass by date. But you can still see across the facets if you want to compare them as well.