Assignment 5: Data Visualization

Rachel Williams

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);library(lubridate);library(here);library(cowplot)
```

```
## -- 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/EDE_Fall2024
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

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

```
PeterPaul_processed <- read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Pr
NiwotRidge_litter <- read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2
class(PeterPaul_processed$sampledate)
```

```
## [1] "character"
```

```
PeterPaul_processed$sampledate <- ymd(PeterPaul_processed$sampledate)
class(PeterPaul_processed$sampledate)
```

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

```
class(NiwotRidge_litter$collectDate)
```

```
## [1] "character"
```

```
NiwotRidge_litter$collectDate <- ymd(NiwotRidge_litter$collectDate)
class(NiwotRidge_litter$collectDate)
```

```
## [1] "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(axis.text = element_text(color = "black"),
        plot.background = element_rect(color="gray"), plot.title=element_text(size=16, face="bold", hju
```

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()`).

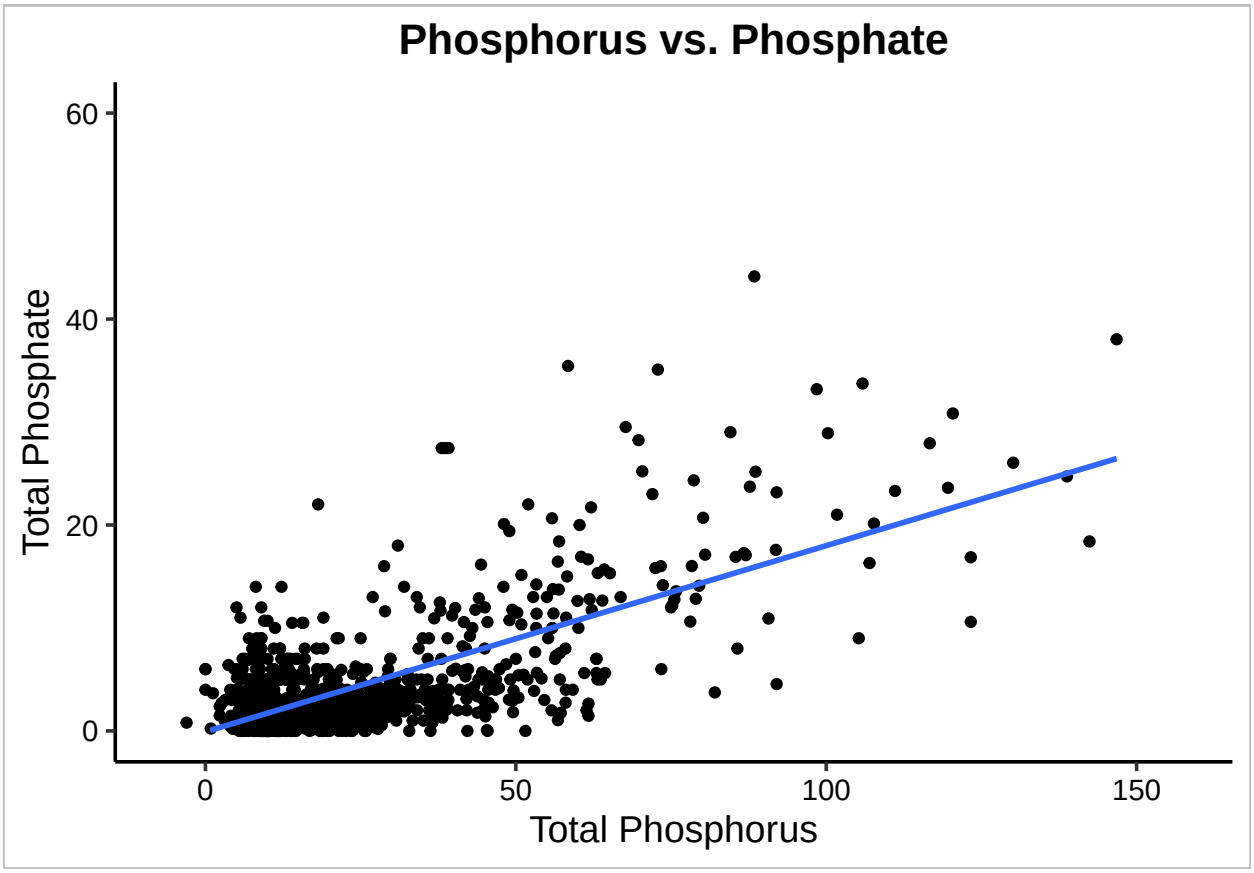
```
#4
PhosGraph <- PeterPaul_processed %>%
ggplot(
    aes(
      x=tp_ug,
      y=po4),
    ) + geom_point() + ggtitle("Phosphorus vs. Phosphate")  + ylim(0, 60) + xlab("Total Phosphorus") + y
  mytheme
PhosGraph
```

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

```
## Warning: Removed 21947 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_smooth()`).
```

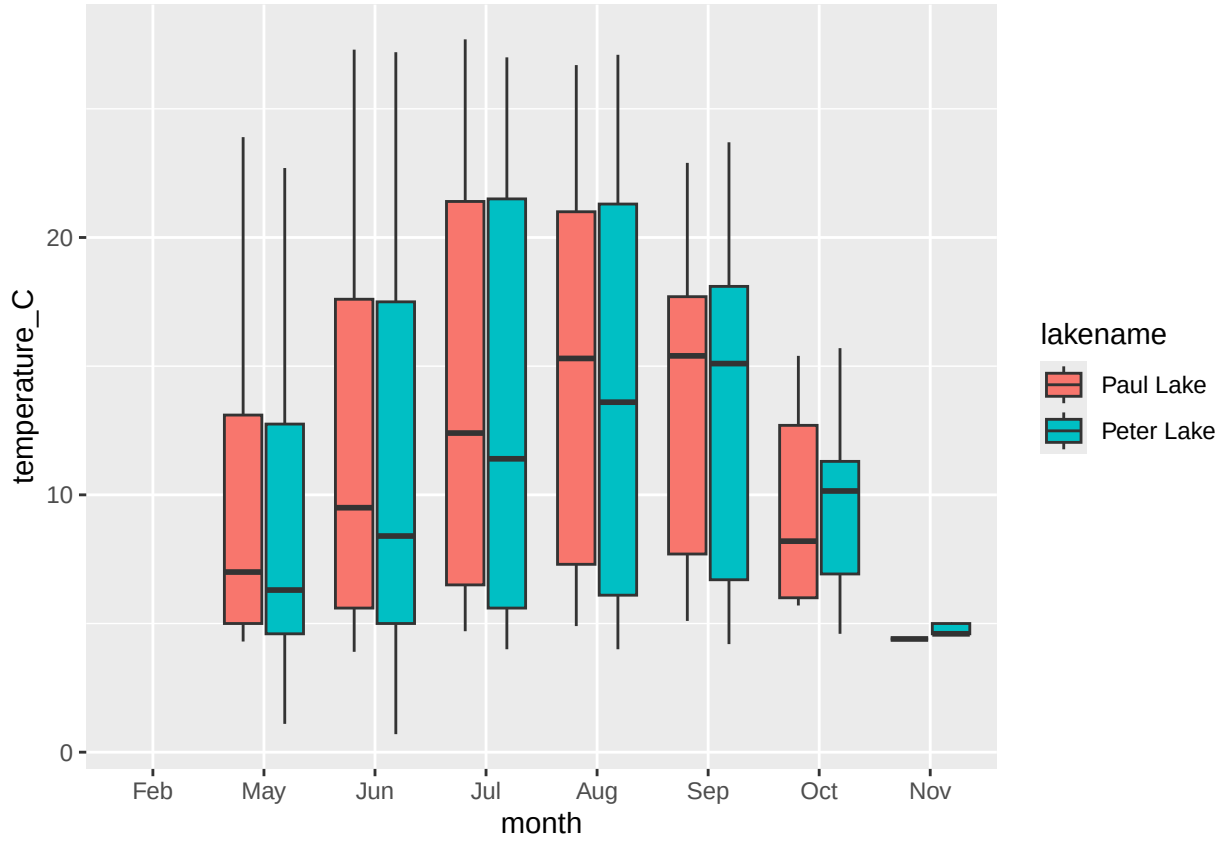


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

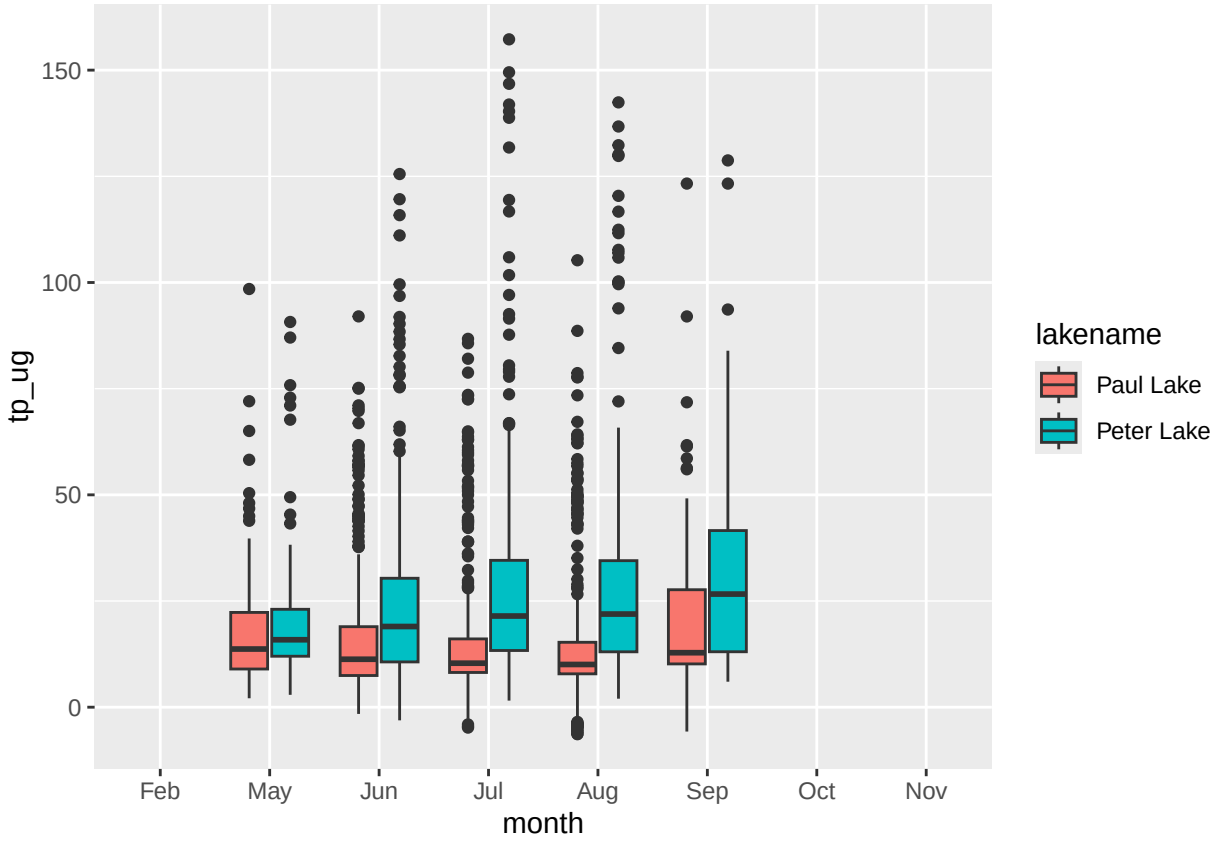
```
#5
PeterPaul_processed$month <-factor(PeterPaul_processed$month,
       levels= 1:12,
       labels = month.abb)

tempbox <- PeterPaul_processed %>%
  ggplot(aes(x=month, y=temperature_C, fill=lakename)) +
  geom_boxplot()
tempbox
```

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



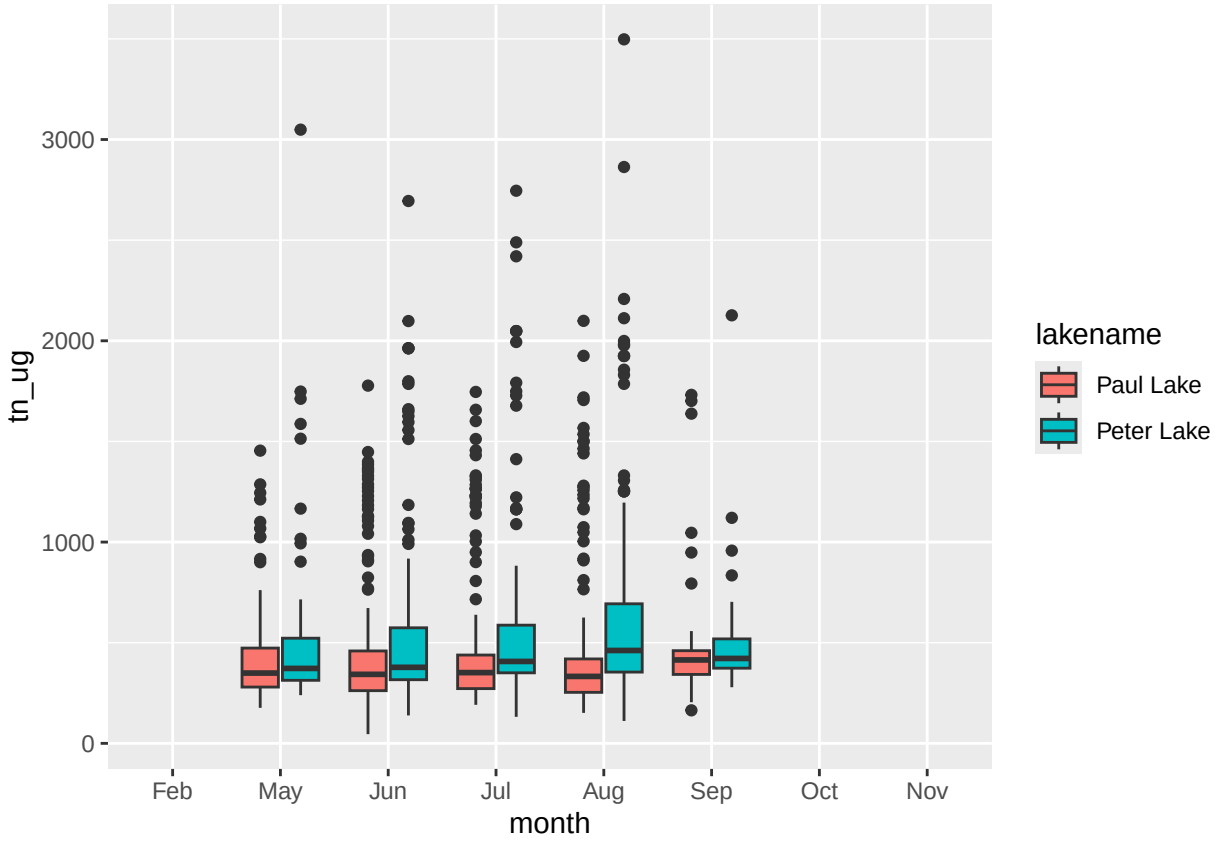

```
TPbox <- PeterPaul_processed %>%
  ggplot(aes(x= month, y= tp_ug, fill=lakename)) +
  geom_boxplot()
TPbox
```

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




```
TNbox <- PeterPaul_processed %>%
  ggplot(aes(x=month, y=tn_ug, fill=lakename)) +
  geom_boxplot()
TNbox
```

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



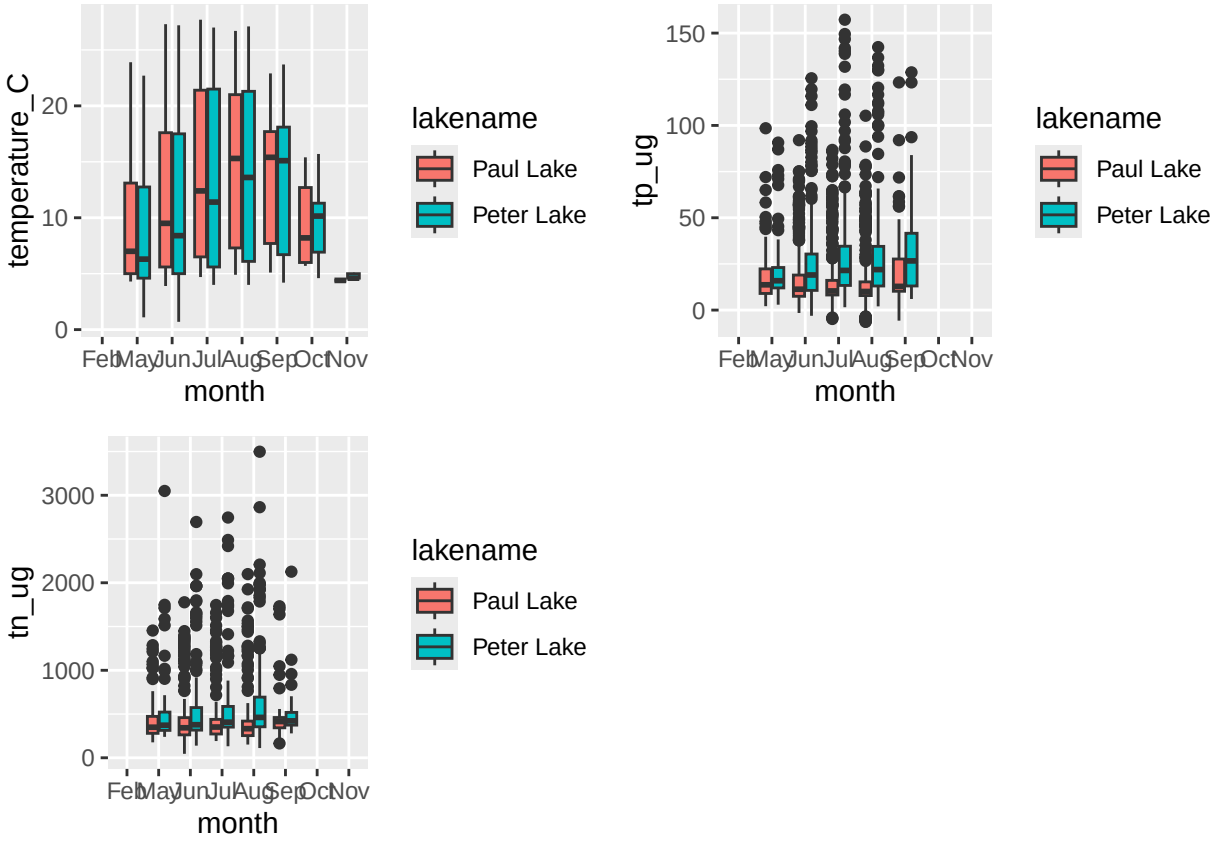

```
combined_plot <- plot_grid(tempbox, TPbox, TNbox, ncol=2)
```

```
## 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()‘).
```

```
combined_plot
```



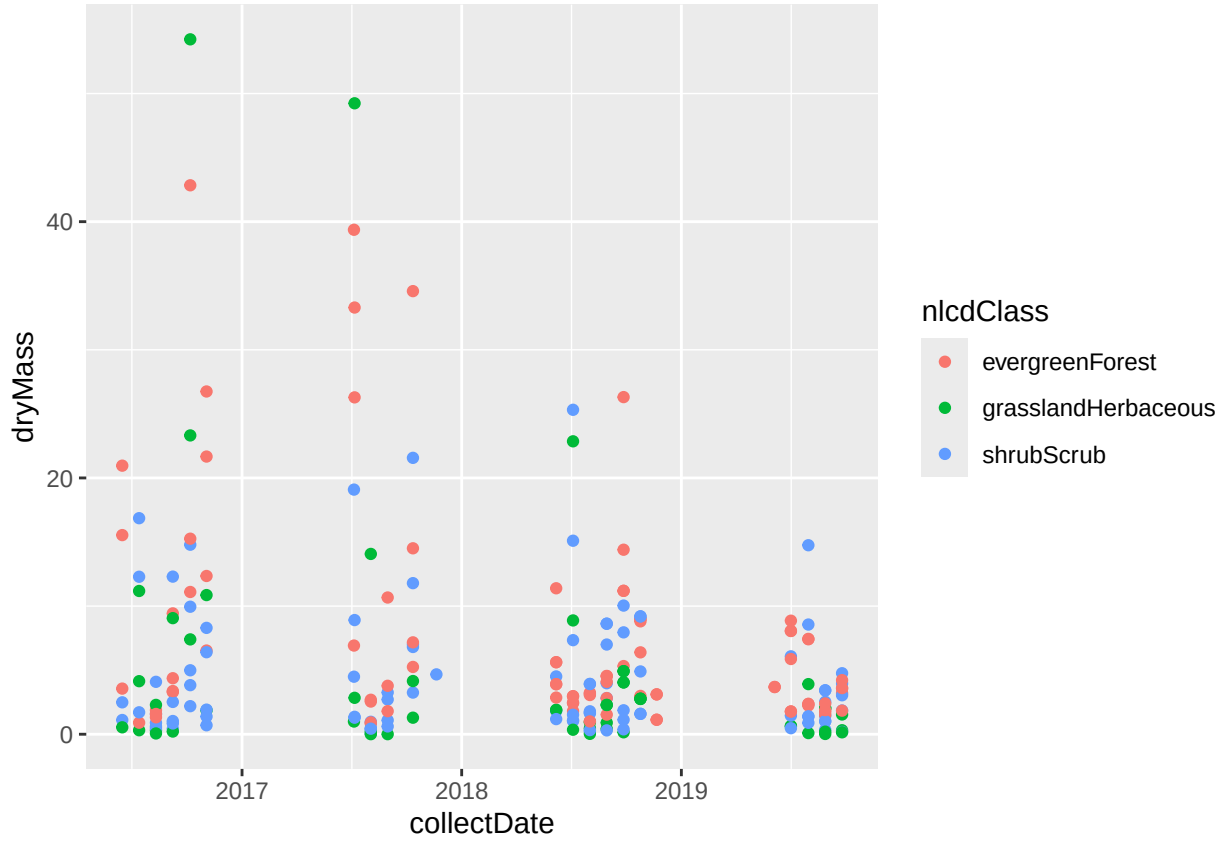
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Typically the variables of interest increase during the summer with a peak in July ot August. Peter Lake seems to have greater amounts of TP and TN than Paul lake. Both lakes have similar temperatures to each other each month.

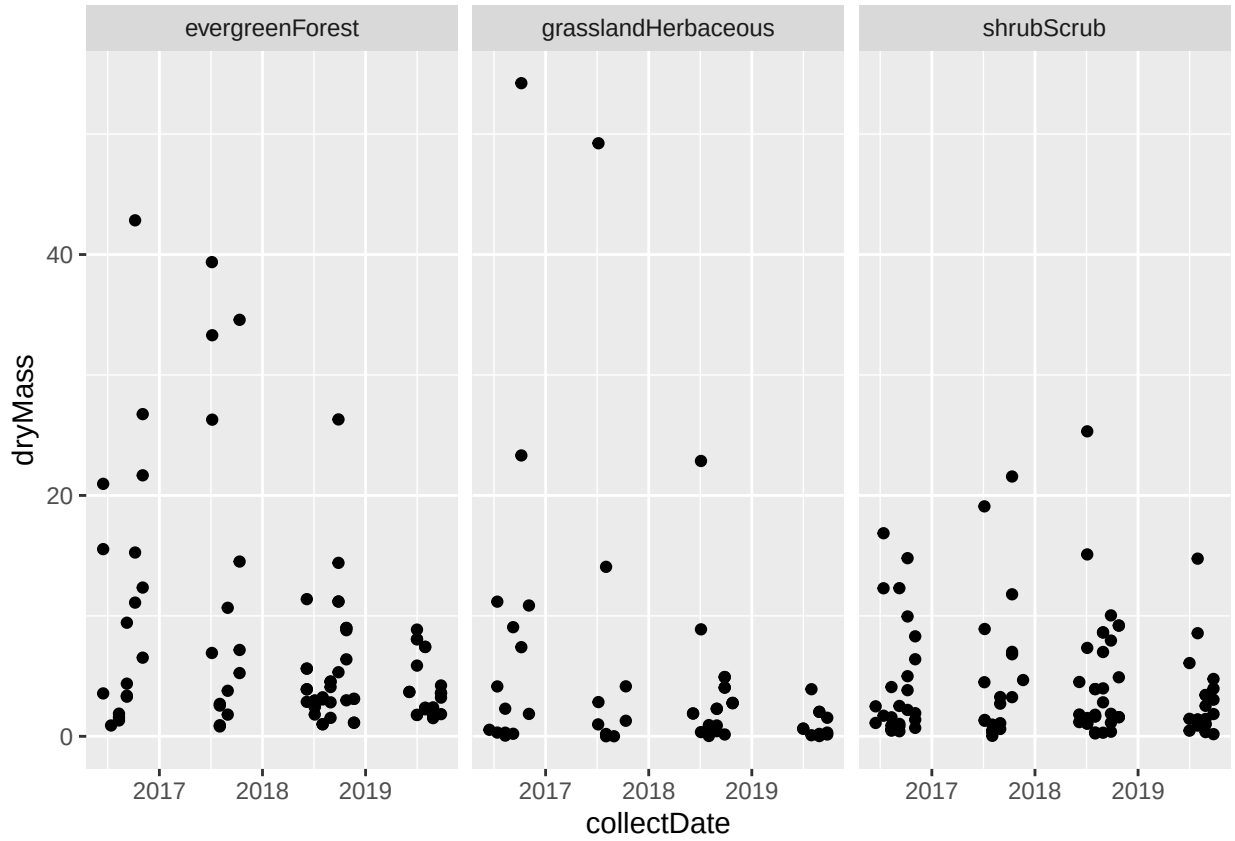
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

needles <- NiwotRidge_litter[NiwotRidge_litter$functionalGroup == "Needles",]

Needlemass <- needles %>%
ggplot(aes(x=collectDate, y=dryMass, color = nlcdClass)) + geom_point()
Needlemass
```




```
#7

Needlemass2 <- needles %>%
  ggplot(aes(x=collectDate, y=dryMass)) + geom_point() +
  facet_wrap(facets= vars(nlcdClass),ncol = 3, nrow = 1)
Needlemass2
```



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

Answer: I think plot 7 is more effective because plot 6 has overlapping points that make it hard to read. In plot 7 you can look at each location seperately and compare them if needed. If plot 7 had differentiated colors between all three areas it would also be an effective way to visualize the difference without covering other points.