Assignment 5: Data Visualization

Justin Maynard

Fall 2023

```
#Clear environment
#rm(list = ls())
```

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     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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 /Users/justinmaynard/Fall_2023_EDE
```

```
library(cowplot)
```

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

```
here()
```

```
## [1] "/Users/justinmaynard/Fall_2023_EDE"
```

```
NTL_LTER_chemistry <- read.csv(file = "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_P
NEON_NIWO_litter <- read.csv(file = "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(NTL_LTER_chemistry$sampledate)
```

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

```
class(NEON_NIWO_litter$collectDate)
```

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

```
NTL_LTER_chemistry$sampledate <- ymd(NTL_LTER_chemistry$sampledate )
NEON_NIWO_litter$collectDate <- ymd(NEON_NIWO_litter$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
my_theme <-
  theme(
      plot.background = element_rect(fill = "#f5f0d9", color = NA),
      panel.background = element_rect(fill = "#f5f0d9", color = NA),
      legend.background = element_rect(fill = "#f5f0d9", color = NA),
      panel.grid.major =  element_line(color = "#d0d0d0", linewidth = .2),
      legend.position="top")
```

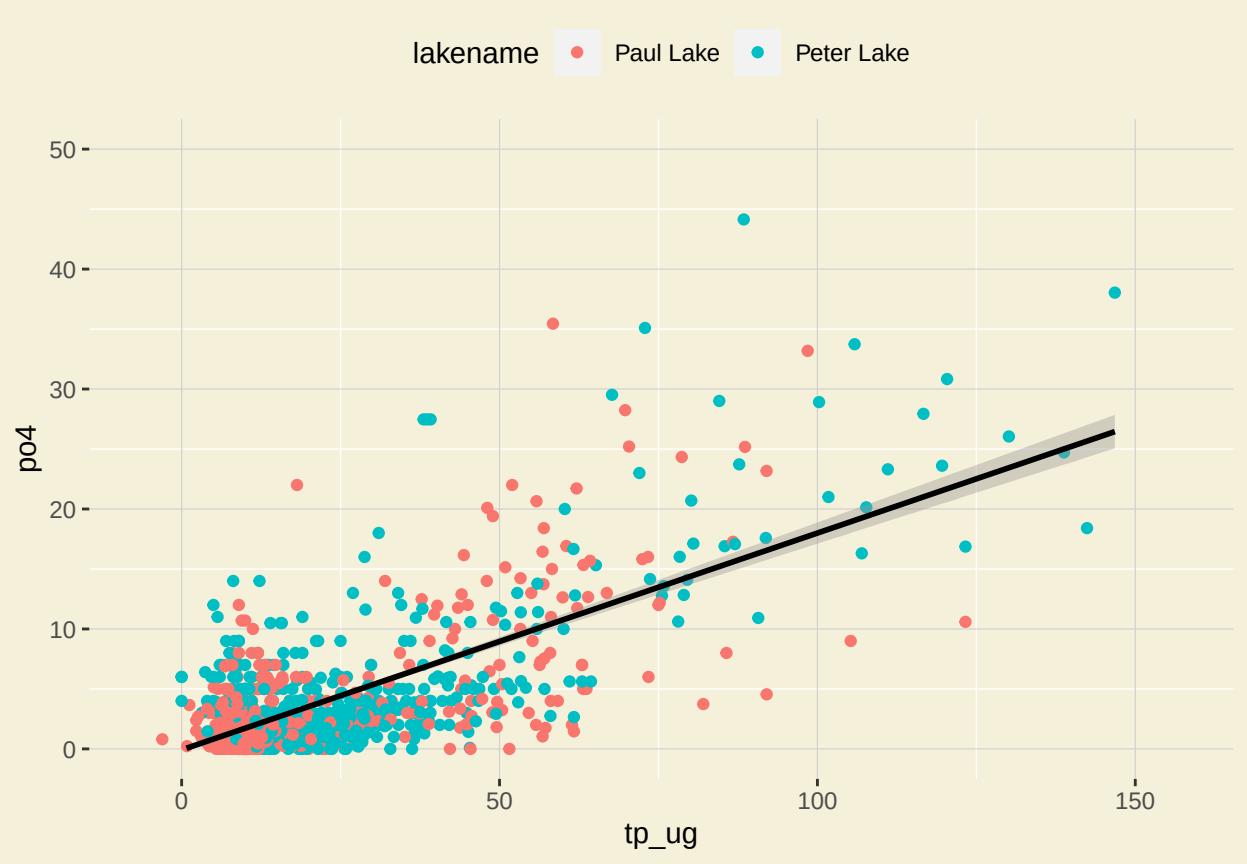
Create graphs

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 a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4
q4 <- ggplot(NTL_LTER_chemistry,
             aes(
               x = tp_ug,
               y = po4,
               color = lakename
             )) +
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  ylim(c(0,50)) + my_theme
q4
```

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

## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21947 rows containing missing values (`geom_point()`).

## Warning: Removed 2 rows containing missing values (`geom_smooth()`).
```

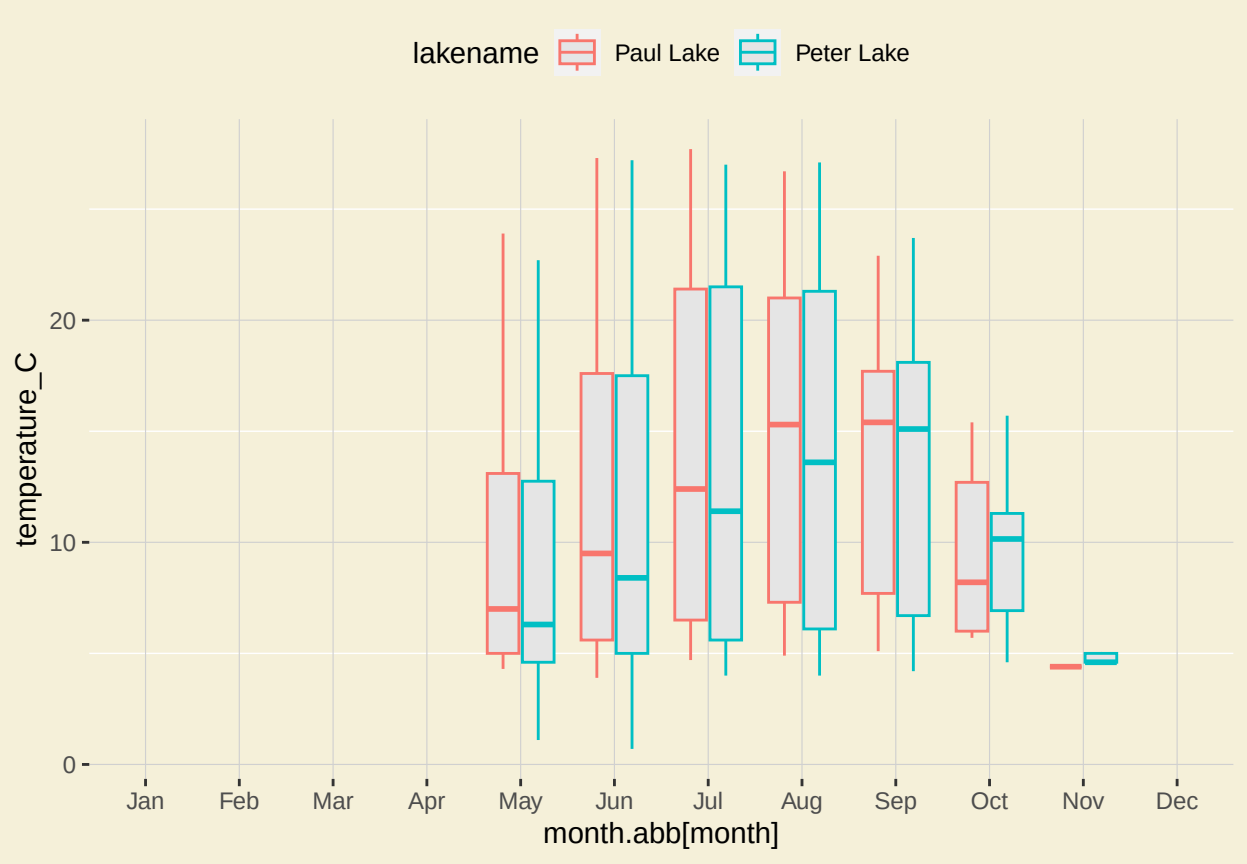


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

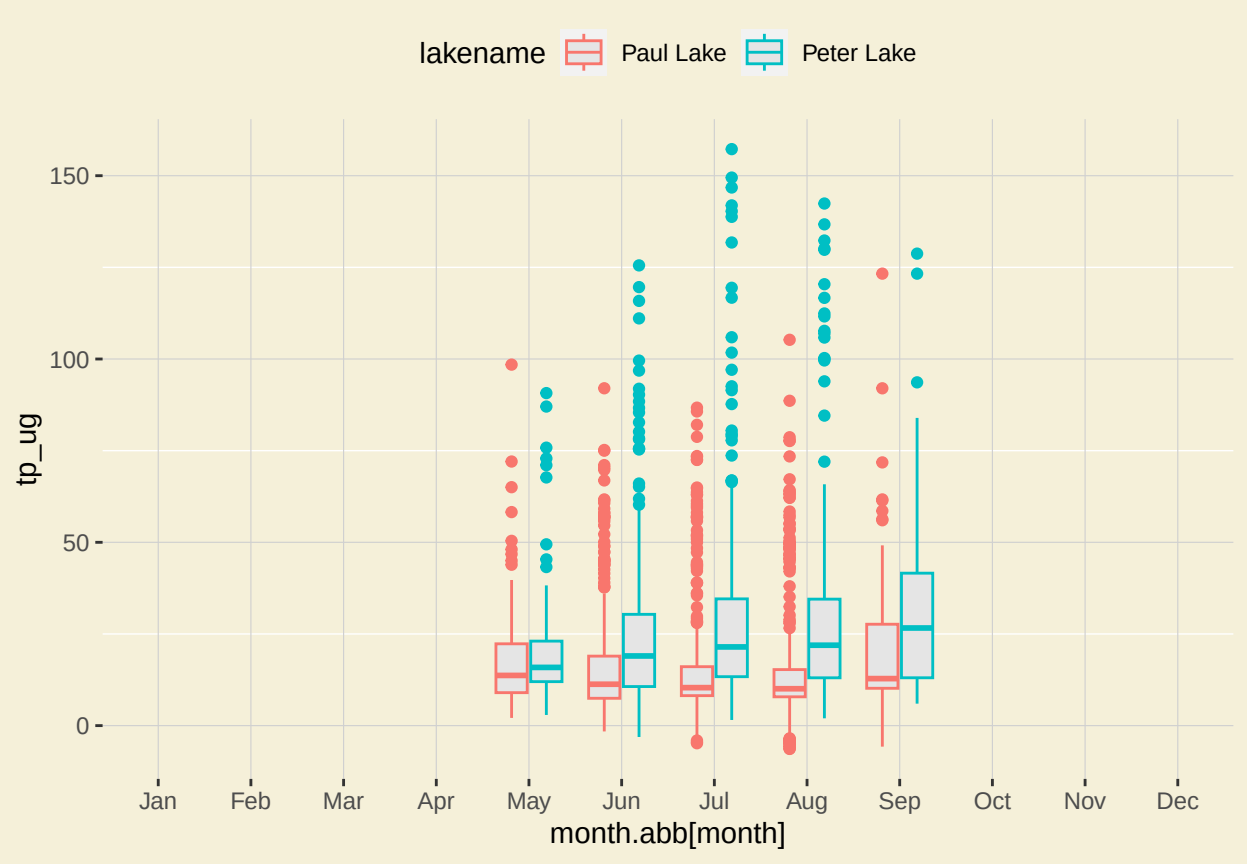
```
#5
library(cowplot)

q5a <- ggplot(NTL_LTER_chemistry,
              aes(
                x = month.abb[month],
                y = temperature_C,
                color = lakename
              )) +
  geom_boxplot(fill = "grey90") +
  my_theme +
  scale_x_discrete(limits = month.abb)
q5a
```

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



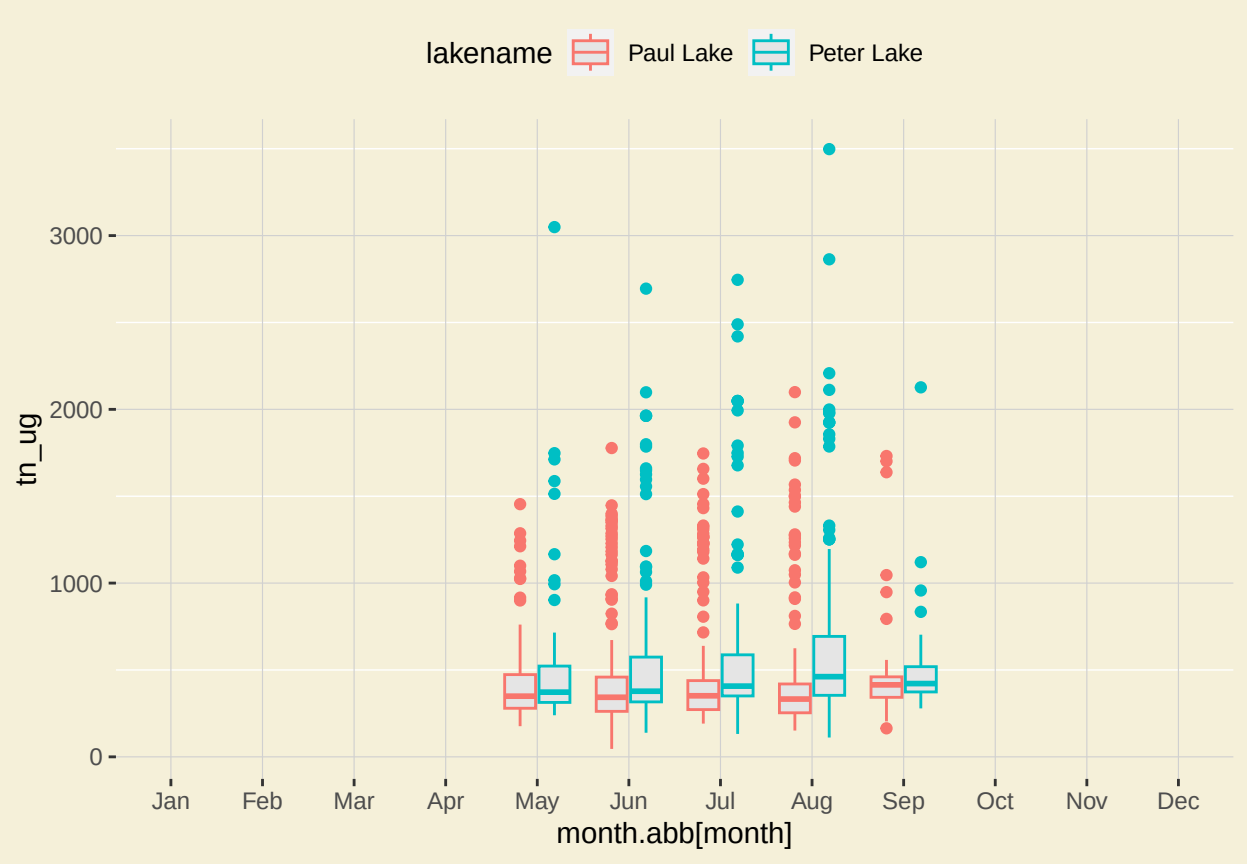

```
q5b <- ggplot(NTL_LTER_chemistry,
              aes(
                x = month.abb[month],
                y = tp_ug,
                color = lakename
              )) +
  geom_boxplot(fill = "grey90") + my_theme +
  scale_x_discrete(limits = month.abb)
q5b
```

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




```
q5c <- ggplot(NTL_LTER_chemistry,
              aes(
                x = month.abb[month],
                y = tn_ug,
                color = lakename
              )) +
  geom_boxplot(fill = "grey90") + my_theme +
  scale_x_discrete(limits = month.abb)
q5c
```

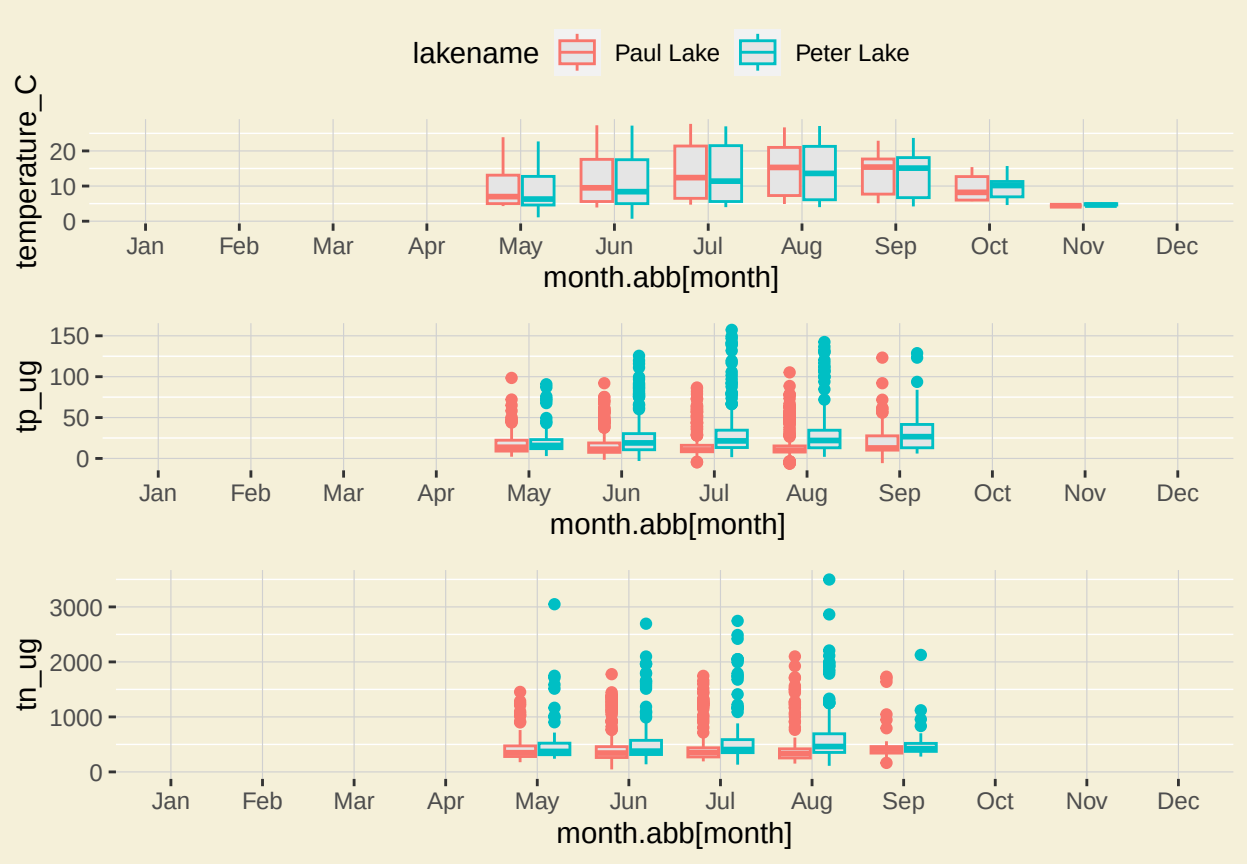
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```




```
q5_final <- plot_grid(q5a + theme(legend.position="top"),
          q5b + theme(legend.position = "none"),
          q5c + theme(legend.position = "none"),
          nrow = 3,
          align = 'h',
          rel_heights = c(1.25, 1)) + my_theme
```

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

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

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

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

```
q5_final
```



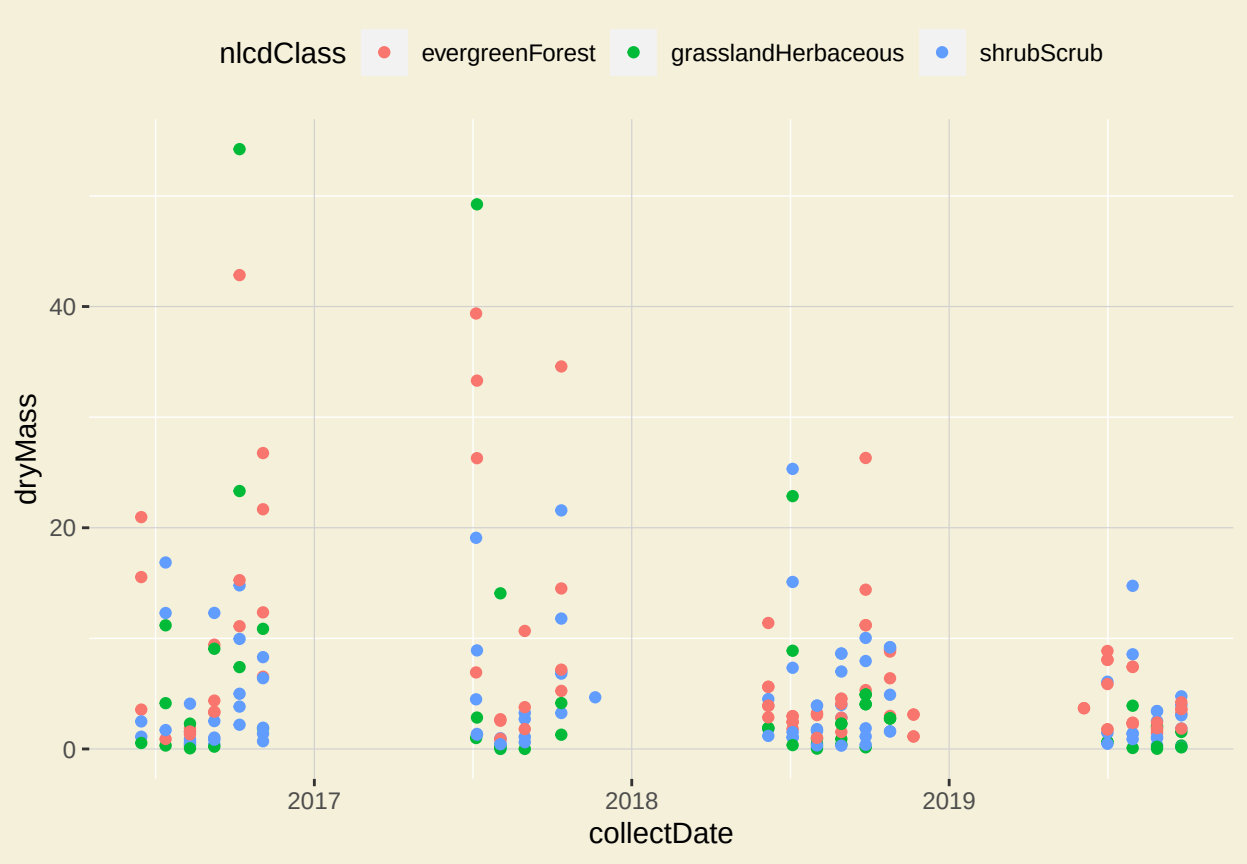
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperature increases during the summer months, while the tp_ug slighly increases for Peter Lake in the late summer. The tn_ug also slightly increases for Peter lake in the late summer.

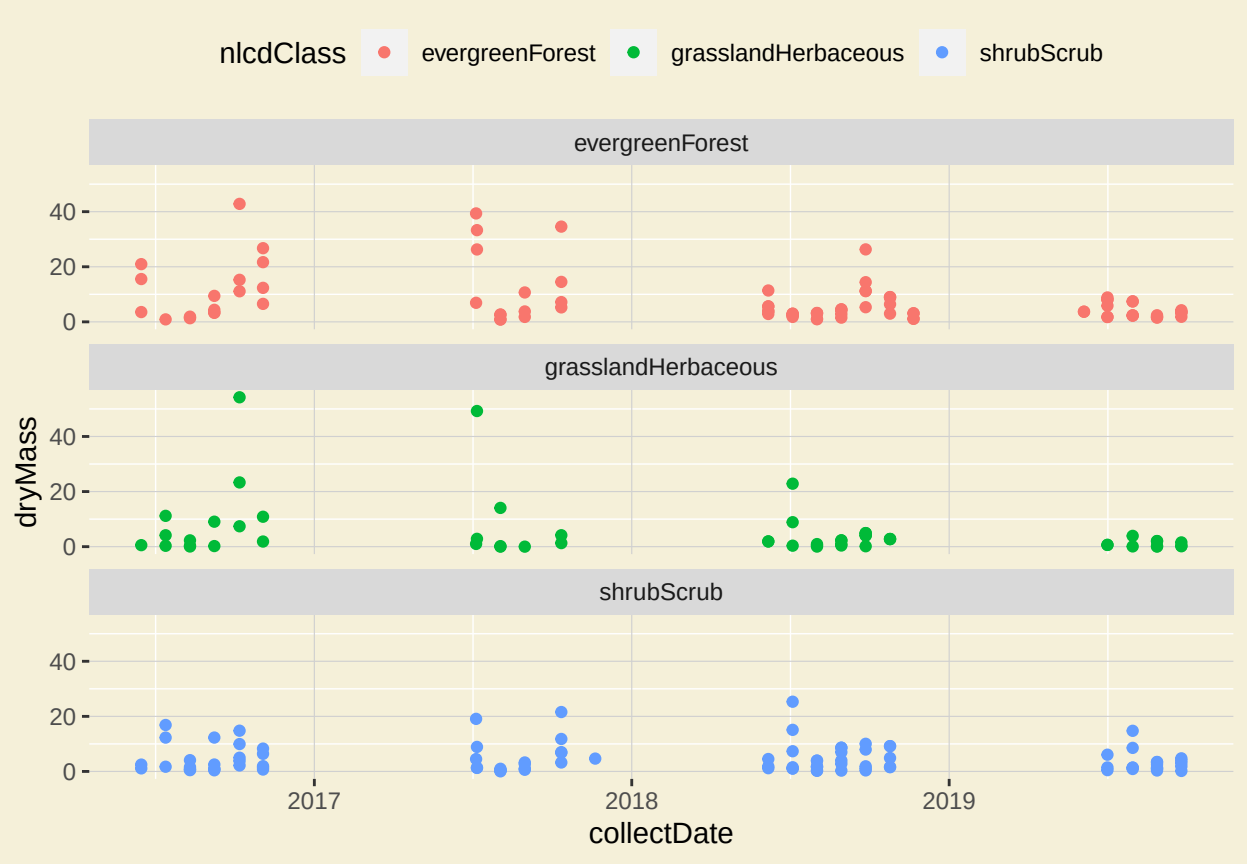
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
q6 <- ggplot(subset(NEON_NIWO_litter, functionalGroup == "Needles"),
       aes(
         x = collectDate,
         y = dryMass,
         color = nlcdClass
       )) +
  geom_point() + my_theme
q6
```




```
#7
q7 <- ggplot(subset(NEON_NIWO_litter, functionalGroup == "Needles"),
       aes(
         x = collectDate,
         y = dryMass,
         color = nlcdClass
       )) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3) + my_theme
q7
```



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

Answer: Plot 7 is more effective as it sepeates out the nlcd class type, allowing one to visualize the change in dry mass for each class by date. Plot 6 is too busy to try and determine change in dry mass by class by year.