Assignment 5: Data Visualization

Alejandra Rios

Fall 2023

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 set up session

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
```

```r
library(lubridate)
library(here)
```

```
## here() starts at C:/Users/Owner/OneDrive - Duke University/Documents/EDE_Fall2023
```

```r
library(ggplot2)
library(cowplot)
```

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

```r
chem_nutrients <- read_csv("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
```

```
## Rows: 23008 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): lakename
## dbl  (13): year4, daynum, month, depth, temperature_C, dissolvedOxygen, irra...
## date  (1): sampledate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
litter <- read_csv("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv")
```

```
## Rows: 1692 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): plotID, trapID, functionalGroup, qaDryMass, nlcdClass, plotType, g...
## dbl  (5): dryMass, subplotID, decimalLatitude, decimalLongitude, elevation
## date (1): collectDate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#2 reading dates as date format

chem_nutrients$sampledate <- as.Date(chem_nutrients$sampledate, format = "%Y/%m/%d")

litter$collectDate <- as.Date(litter$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 build theme
#change axis labels
#change legend position
my_theme <- theme(axis.text = element_text(color = "red"),
                  legend.position = "top")

#set as default theme
theme_set(my_theme)
```

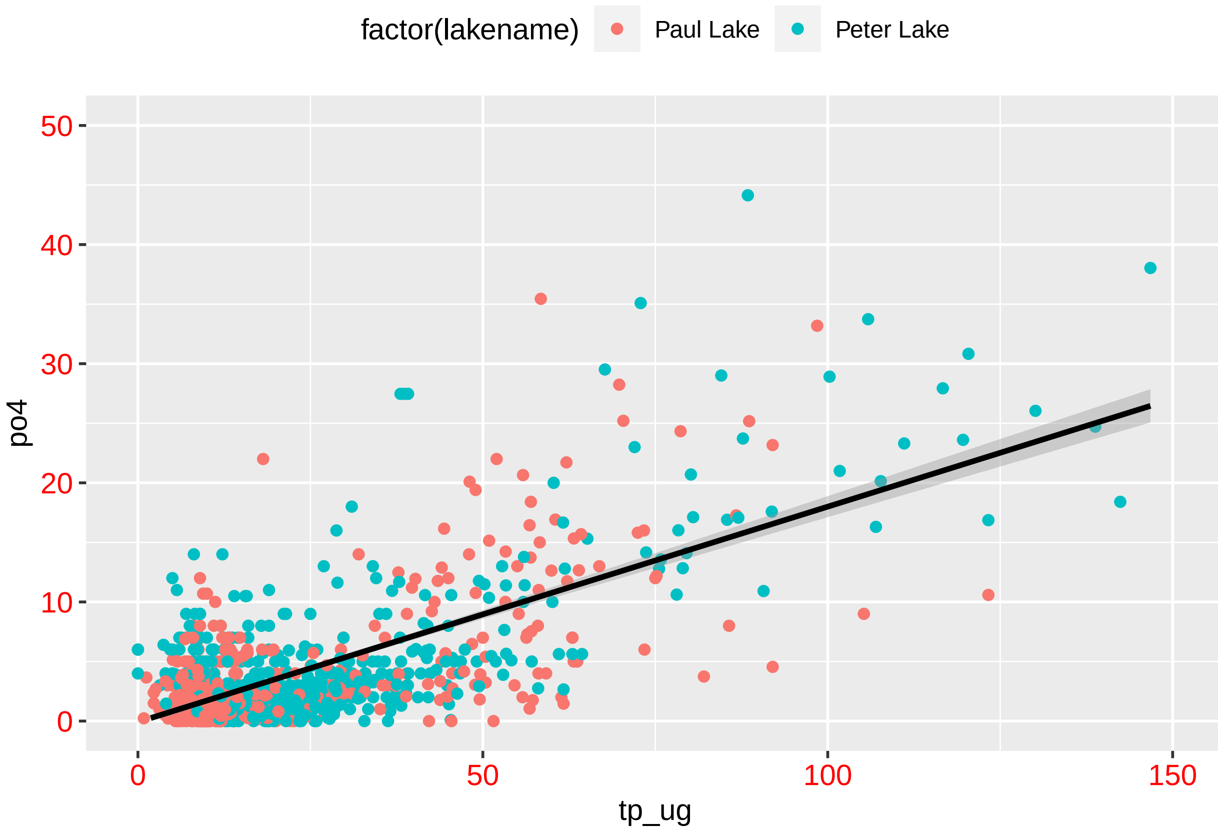
Create graphs

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 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()`).

```
#4 total phosphorous v. phosphate
ggplot(chem_nutrients, aes(x = tp_ug, y = po4, color = factor(lakename)))+ #assign x and y, assign aest
  geom_point()+ #scatterplot
  geom_smooth(method = "lm", formula = y ~ x, color = "black")+
#line of best fit
  xlim(0, 150) + #adjust x scale
  ylim(0, 50) #adjust y scale
```

```
## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21948 rows containing missing values (`geom_point()`).

## Warning: Removed 1 rows containing missing values (`geom_smooth()`).
```

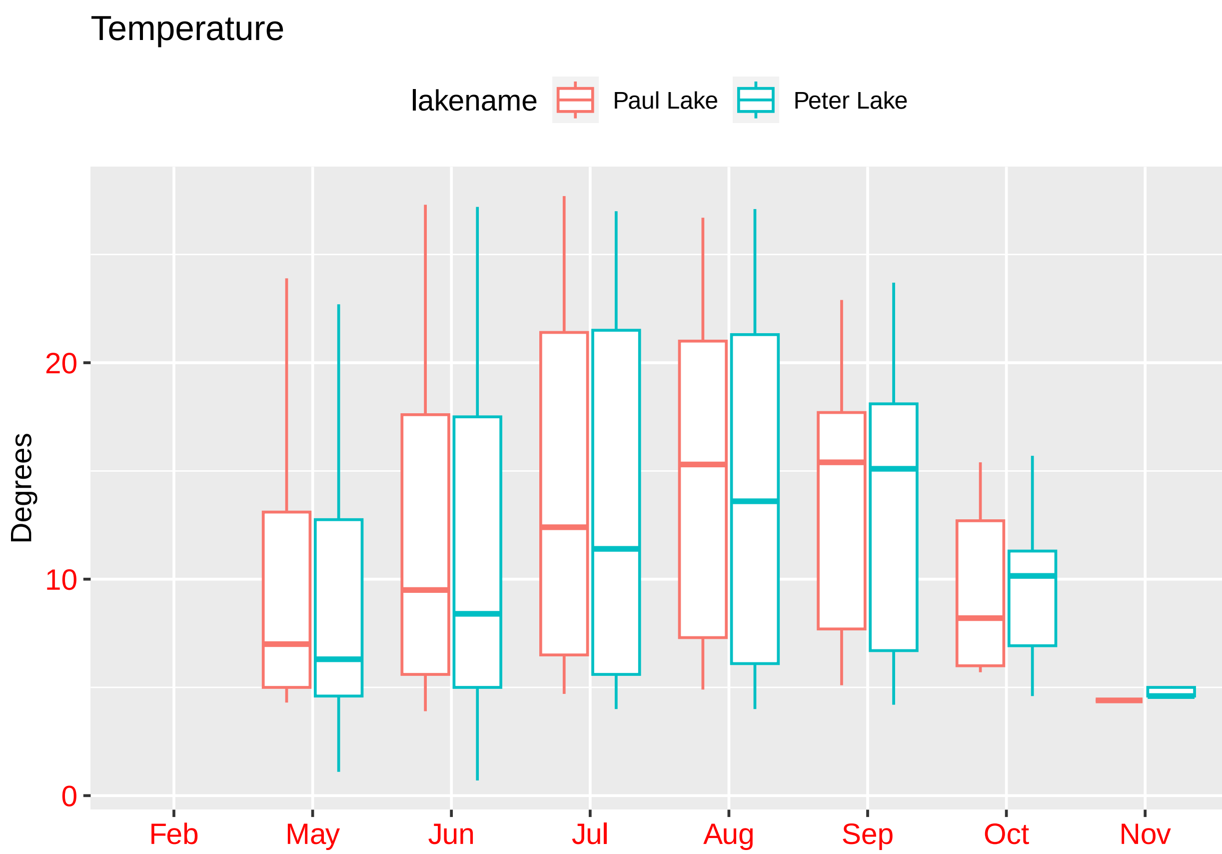


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

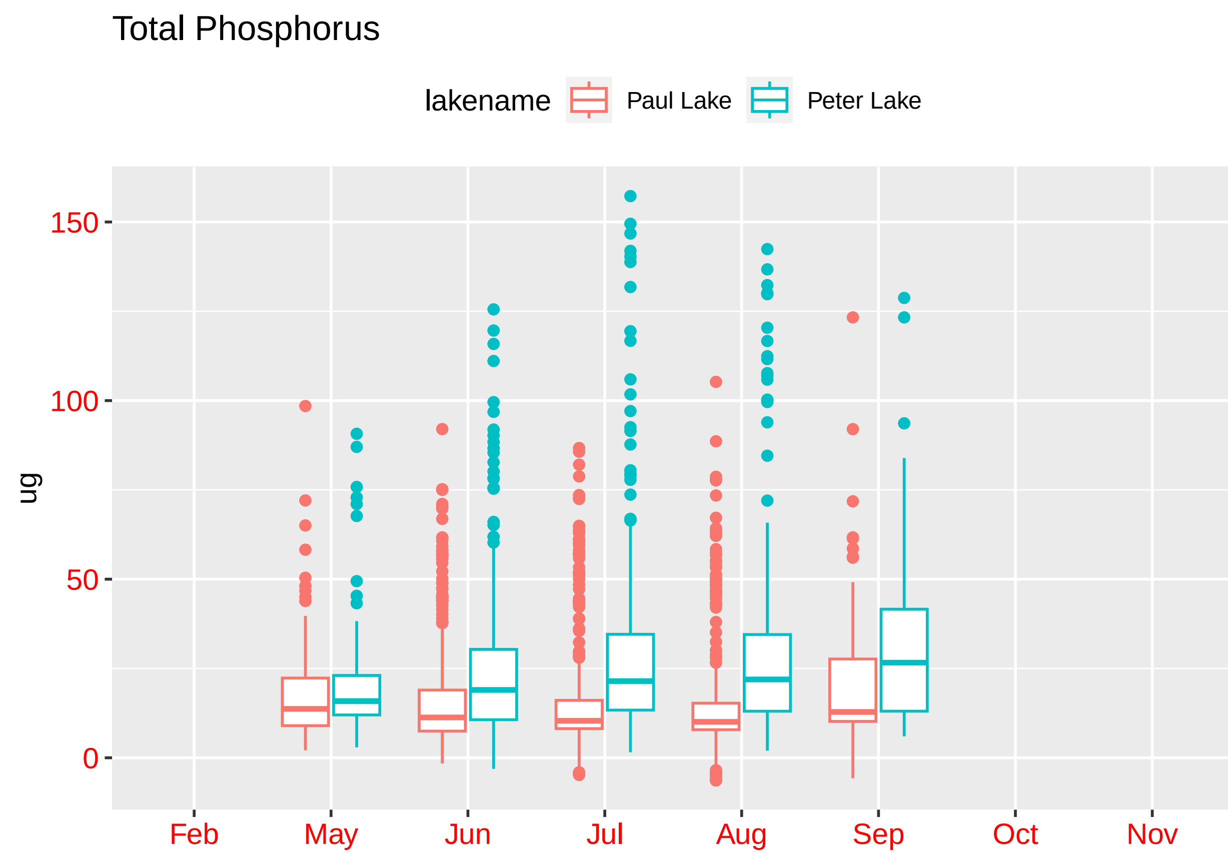
```
#5 3 boxplots
chem_nutrients$month <- factor(chem_nutrients$month,
       levels = 1:12,
       labels = month.abb)

#temperature boxplot
plot_temp <- ggplot(chem_nutrients, aes(x = factor(month), y = temperature_C, color = lakename))+
  geom_boxplot()+
  labs(
    x = NULL,
    y = "Degrees",
    title = "Temperature")
print(plot_temp)
```

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



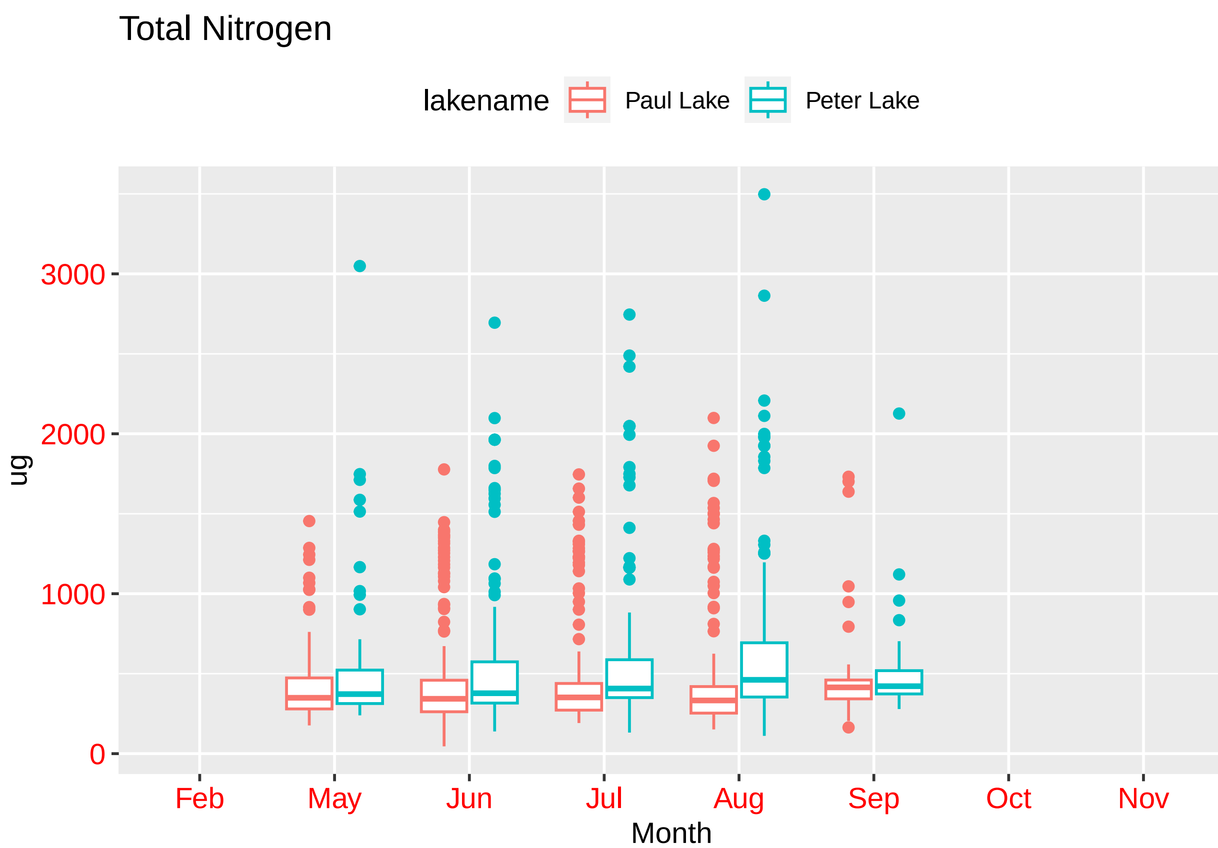

```
#TP boxplot
plot_tp <- ggplot(chem_nutrients, aes(x = factor(month), y = tp_ug, color = lakename))+
  geom_boxplot()+
  labs(
    x = NULL,
    y = "ug",
    title = "Total Phosphorus")
print(plot_tp)
```

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




```r
#TN boxplot
plot_tn <- ggplot(chem_nutrients, aes(x = factor(month), y = tn_ug, color = lakename))+
  geom_boxplot()+
  labs(
    x = "Month",
    y = "ug",
    title = "Total Nitrogen")
print(plot_tn)
```

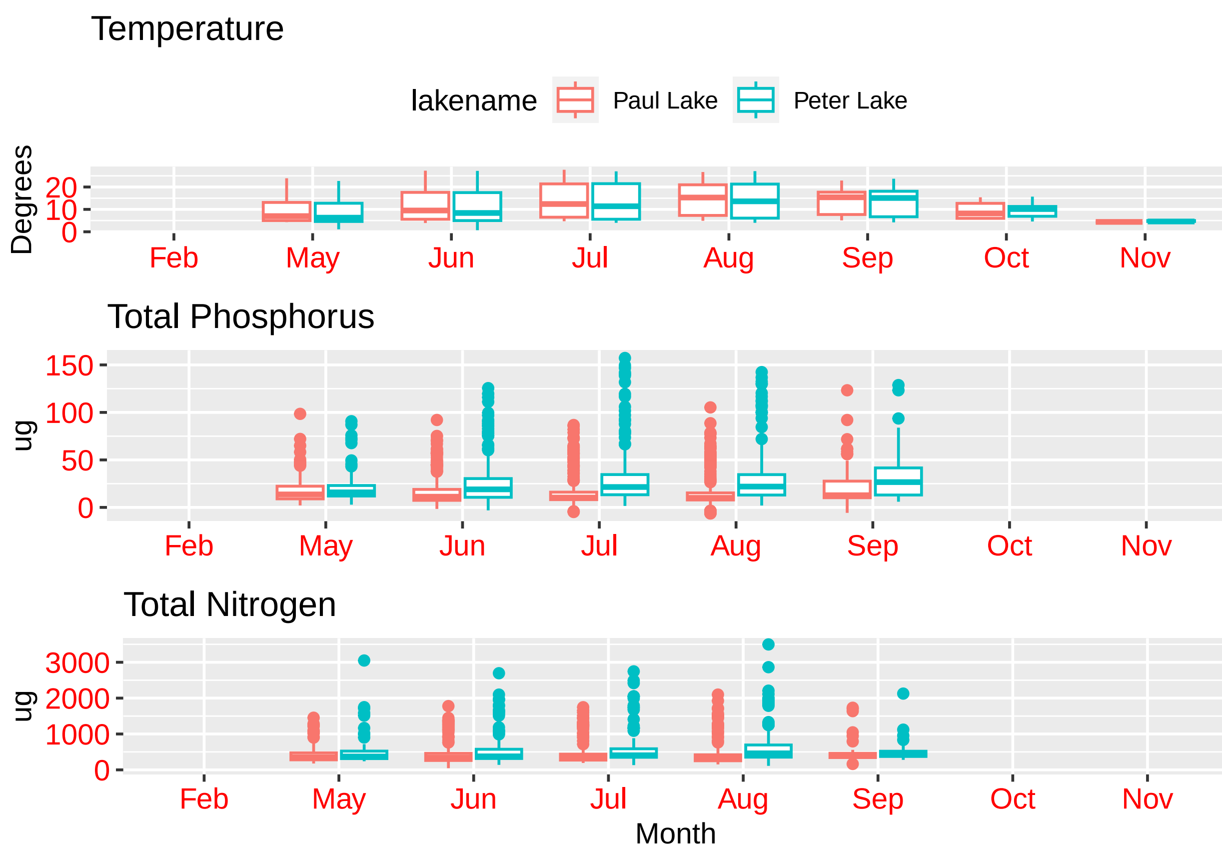
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```




```r
#combine boxplots
combined_plots <- plot_grid(plot_temp + theme(legend.position = "top"),
                            plot_tp + theme(legend.position = "none"),
                            plot_tn + theme(legend.position = "none"), nrow = 3, ncol = 1)
```

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

```r
print(combined_plots)
```

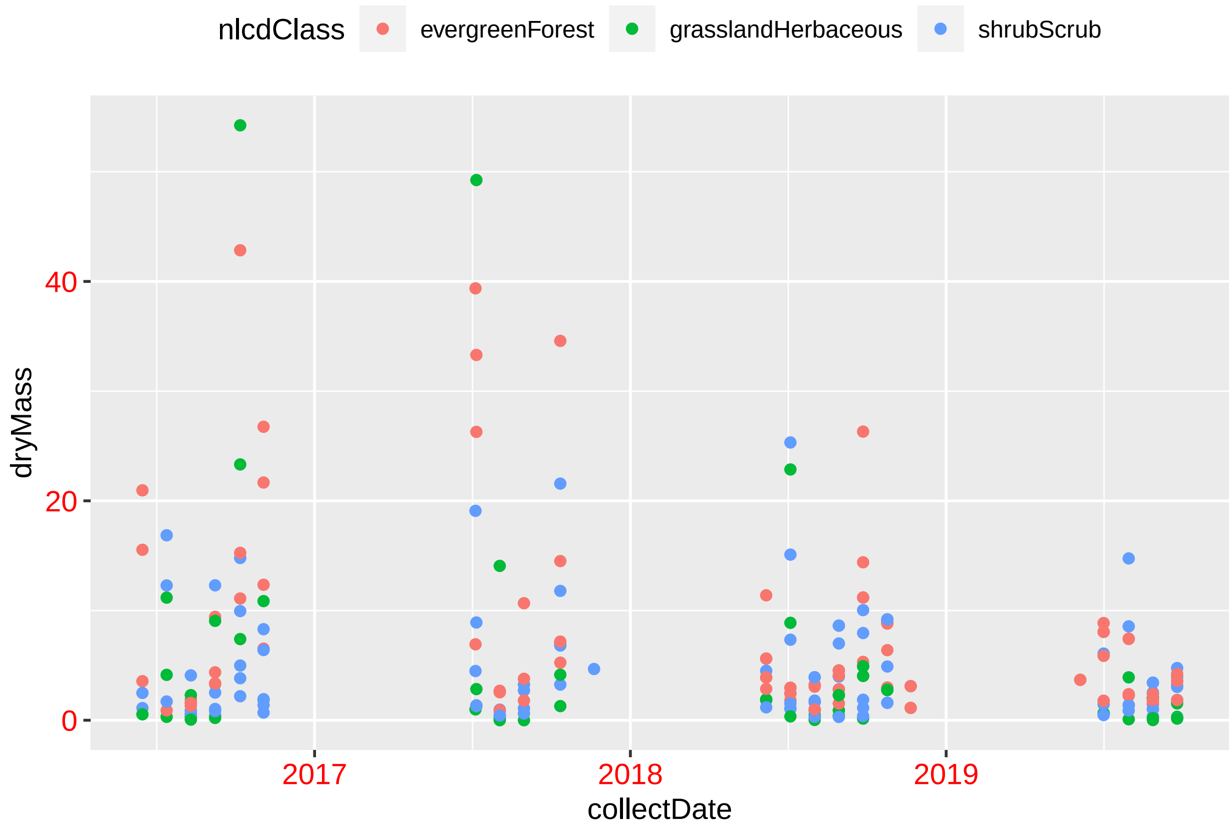


Question: What do you observe about the variables of interest over seasons and between lakes?

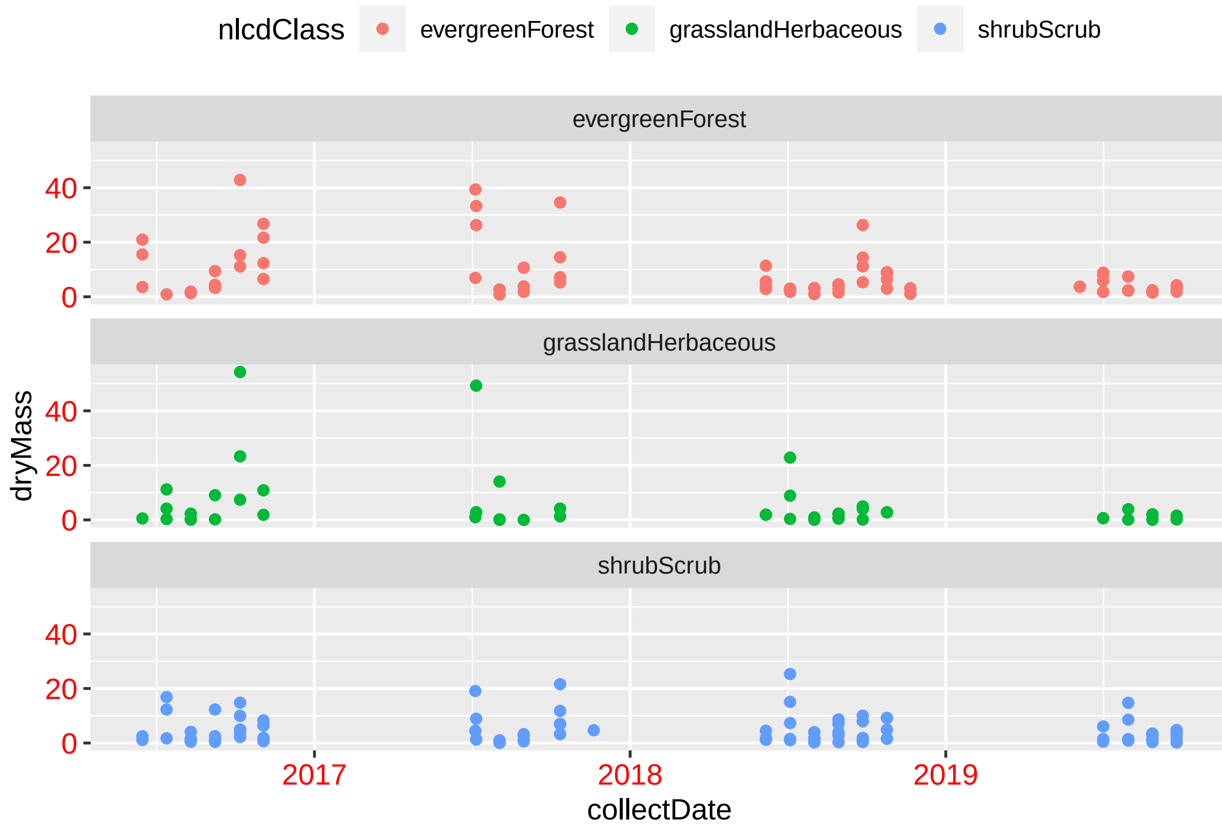
> Answer: Peter Lake has higher temperature, tp, and tn than Paul Lake. Tn and Tp increase and decrease with temperature.

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 plot histogram
litter$collectDate <- as.Date(litter$collectDate, format = "%Y/%m/%d")

needles_graph<- ggplot(subset(litter, functionalGroup == "Needles"),
              (aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point()
print(needles_graph)
```




```r
#7 plot
facet_plot <- ggplot(subset(litter, functionalGroup == "Needles"),(aes(x = collectDate, y = dryMass, co
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)
print(facet_plot)
```



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

Answer: 7 is easier to understand because the data is separated into 3 graphs but aligned.