Assignment 5: Data Visualization

Sarah Sussman

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).

```
# Load packages
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/EDA_Spring2025
```

```
library(cowplot)
```

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

```
# Verify home directory
getwd()
```

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

```
# Read in .csv's
PeterPaul.chem.nutrients <-
  read.csv(("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
NiwotRidge.litter <-
  read.csv(("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)
```

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
#2
NiwotRidge.litter$collectDate <- ymd(NiwotRidge.litter$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

```
#3
mytheme <- theme_bw(base_size = 16) +
  theme(legend.position = "top",
        axis.text = "darkgreen")
```

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()`).

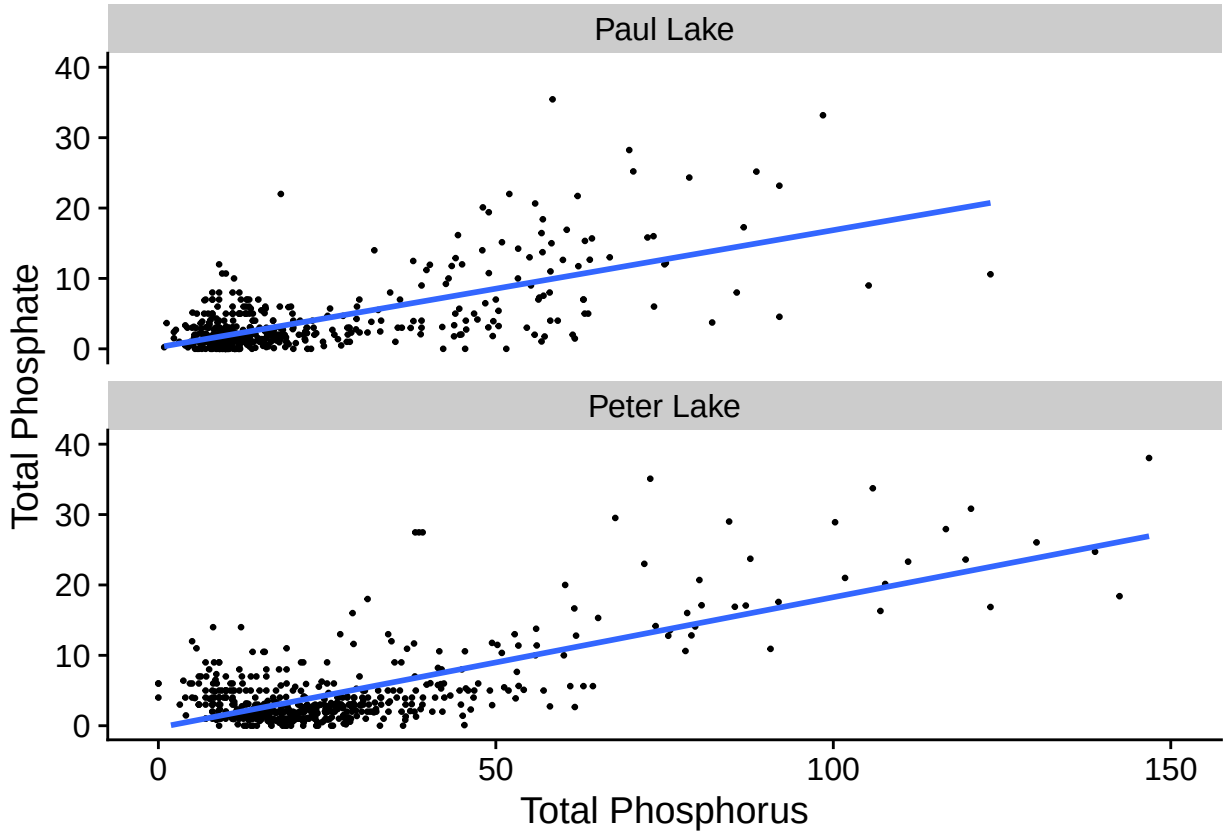
```
#4
PeterPaul.chem.nutrients %>%
  ggplot(aes(x=tp_ug,
             y=po4)) +
  geom_point(size=0.5) +
  theme_cowplot() +
  facet_wrap(facet=vars(lakename), nrow=2,ncol=1) +
  geom_smooth(method=lm, se=FALSE) +
  xlim(0,150) +
  ylim(0,40) +
  labs(x="Total Phosphorus", y="Total Phosphate")
```

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

```
## Warning: Removed 21949 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21949 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 1 row 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.

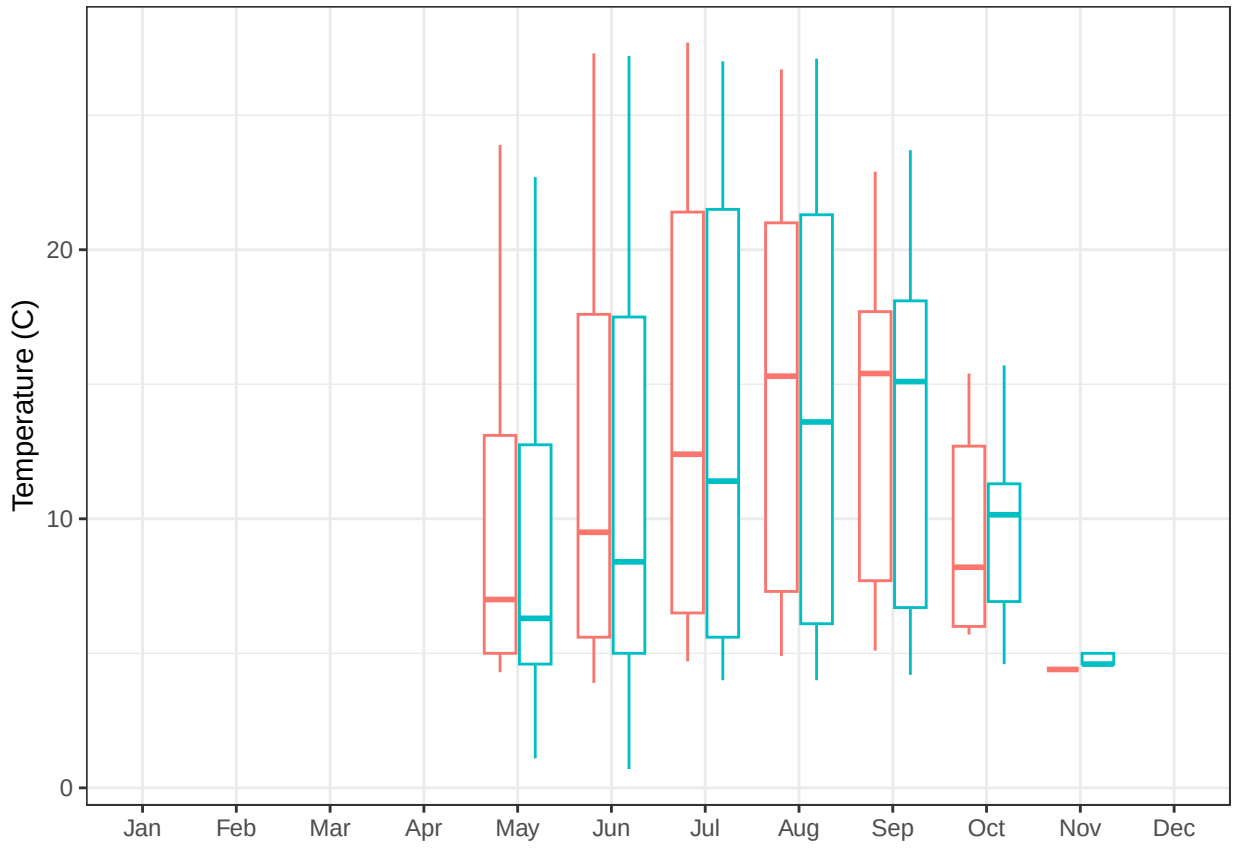
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
# Create temperature plot
temp_plot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x=factor(month,
                            levels=1:12,
                            labels=month.abb),
                            color=lakename,
                   y=temperature_C)) +
scale_x_discrete(drop=F, name = "Month") +
  theme_bw() +
  theme(axis.title.x = element_blank(), legend.position = "none") +
  labs(y = "Temperature (C)")

# Print temperature plot
temp_plot
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
```

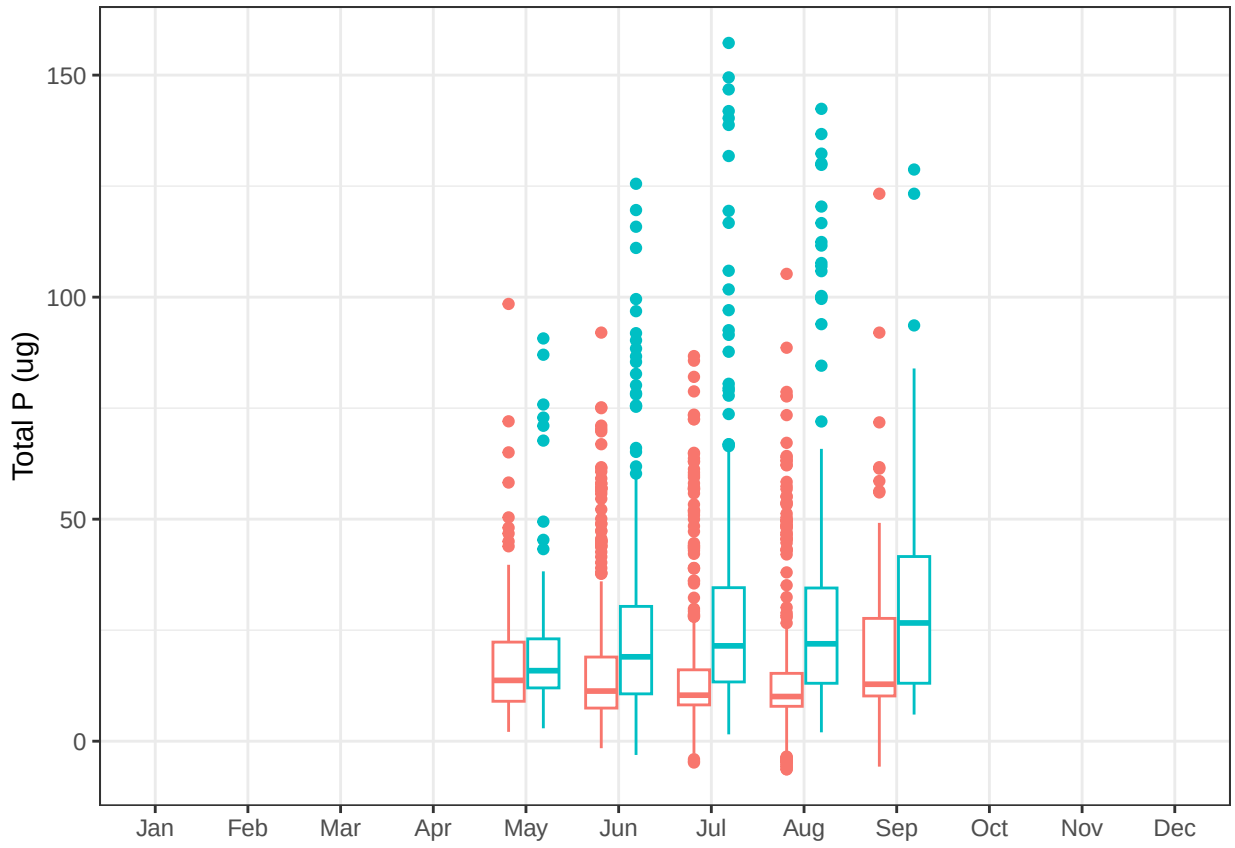
```
## (‘stat_boxplot()‘).
```




```
# Create TP plot
TP_plot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x=factor(month,
                            levels=1:12,
                            labels=month.abb),
                   color=lakename,
                   y=tp_ug)) +
  scale_x_discrete(drop=F, name = "Month") +
  theme_bw() +
  theme(axis.title.x = element_blank(), legend.position = "none") +
  labs(y="Total P (ug)")

# Print TP plot
TP_plot
```

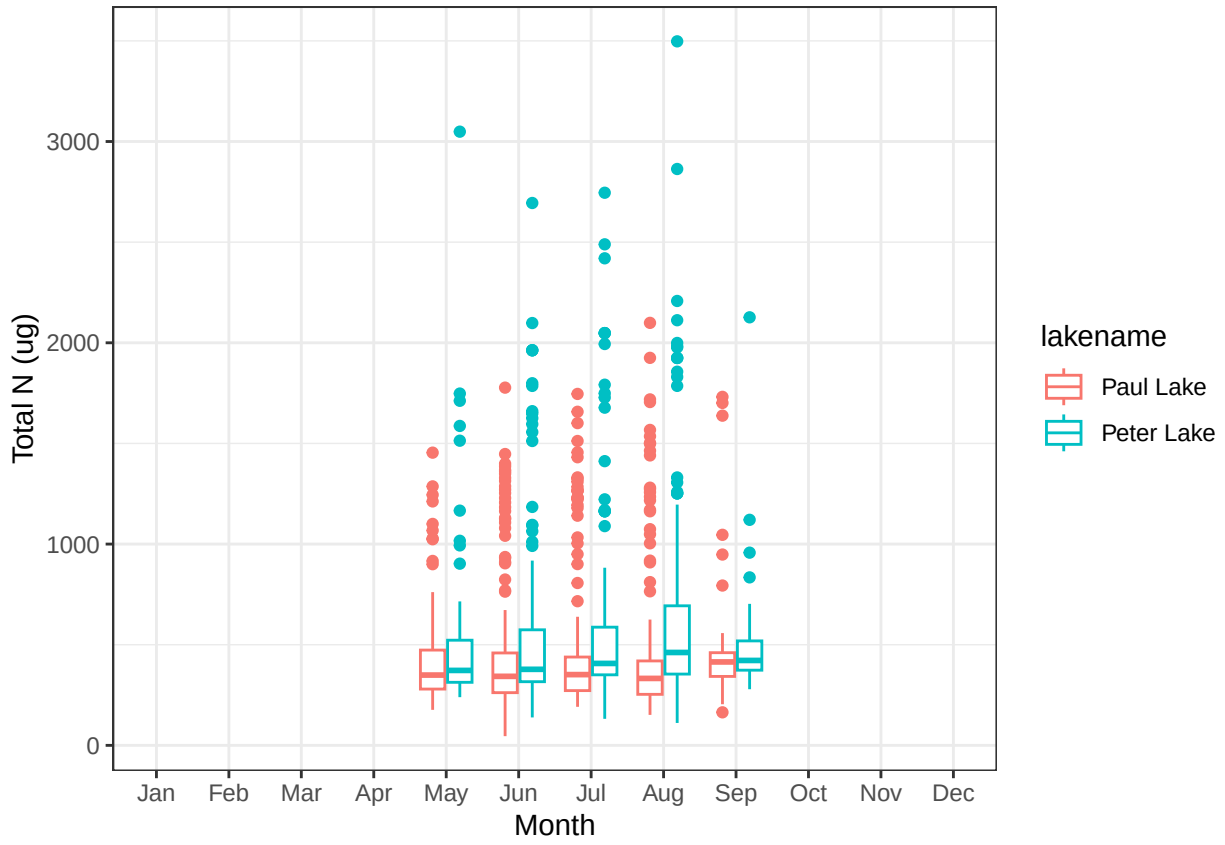
```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
# Create TN plot
TN_plot <- ggplot(PeterPaul.chem.nutrients) +
  geom_boxplot(aes(x=factor(month,
                            levels=1:12,
                            labels=month.abb),
                   color=lakename,
                   y=tn_ug)) +
  scale_x_discrete(drop=F, name = "Month") +
  theme_bw() +
  labs(y="Total N (ug)")

# Print TN plot
TN_plot
```

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

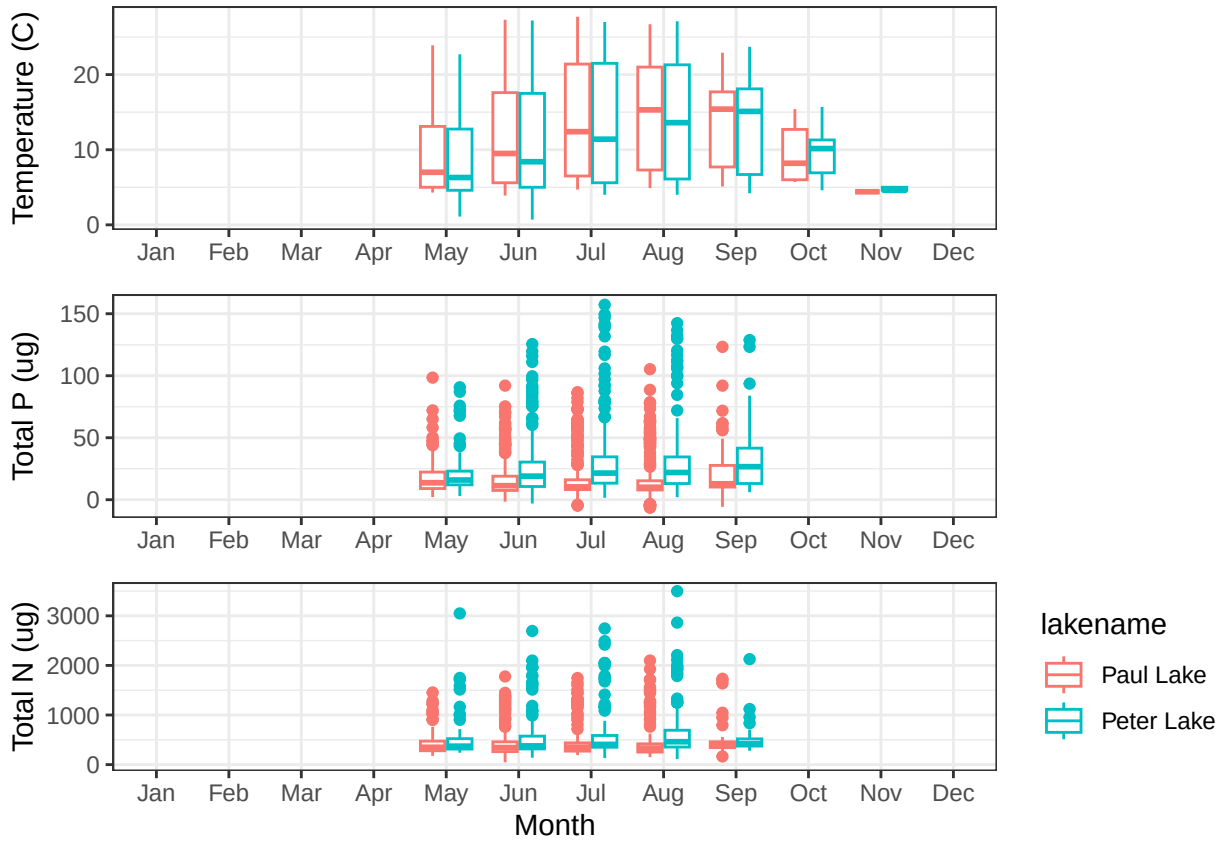



```
# Cowplot
plot_grid(temp_plot,TP_plot,TN_plot, ncol = 1, align = "v", rel_heights = c(2, 2, 2))
```

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



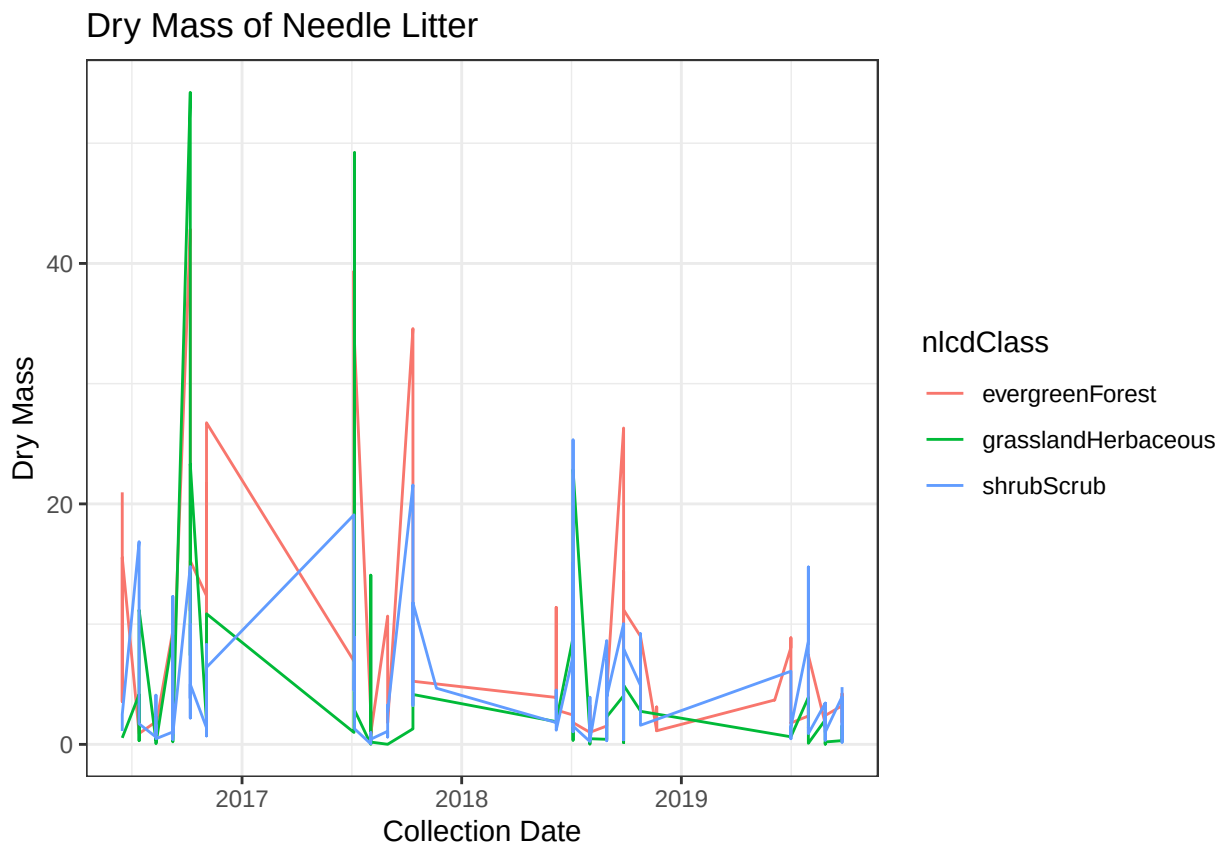
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: For temperature, it seems like both the lakes have similar temperature ranges. Temperatures for both lakes have longer ranges in the summer (June, July, August) compared to the months of May, September, October, and November (November is especially small). For total phosphorus, Peter Lake appears to have a higher total phosphorus number than Paul lake, especially in July and September. For total nitrogen, Peter Laake also apppears to have a higher numbers, but it's not as much of an increase compared to phosphorus.

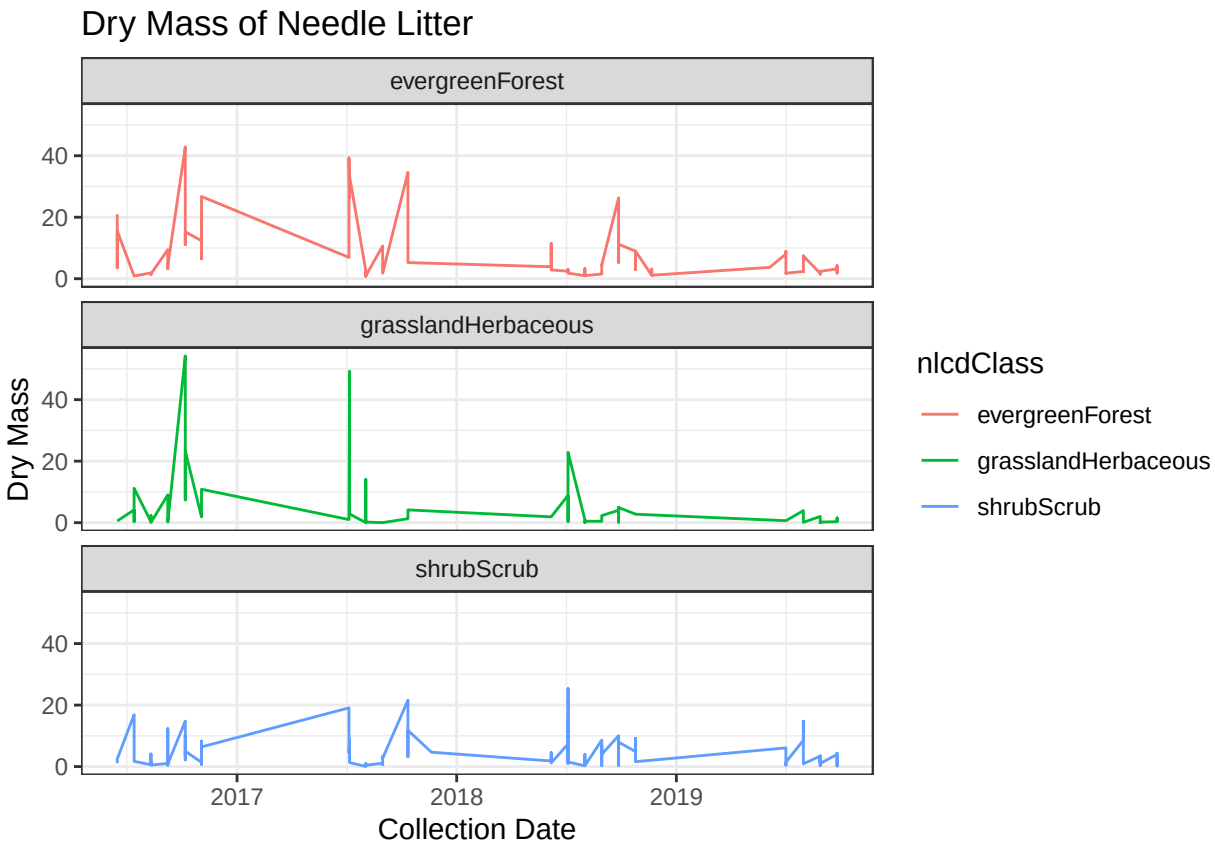
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NiwotRidge.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate,
             y=dryMass,
             color=nlcdClass)) +
  geom_line() +
  theme_bw() +
  labs(x = "Collection Date",
       y = "Dry Mass",
       title = "Dry Mass of Needle Litter")
```




```
#7
NiwotRidge.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate,
             y=dryMass,
             color=nlcdClass)) +
  geom_line() +
  theme_bw() +
  labs(x = "Collection Date",
       y = "Dry Mass",
       title = "Dry Mass of Needle Litter") +
  facet_wrap(facet=vars(nlcdClass),nrow=3,ncol=1)
```



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

Answer: I think that plot 7 is more effective than plot 6. When we are plotting the dry mass of the needle litter for each land class type on the same plot it looks disorganized and messy. When they are one three seperate plots, it is easier to look at the individual data from each land claass and it is easier to make comparisons between them.