

# Assignment 5: Data Visualization

Gary Alvarez

Spring 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      1.1.3      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
## here() starts at /home/guest/R/EDA_Spring2024
```

```
#Library for creating ridge plots
library(ggribes)
```

```
#Enhanced color ramps
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
library(colormap)
install.packages("cowplot")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(cowplot)
```

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

```
here() ##The working directory is "/home/guest/R/EDA_Spring2024"
```

```
## [1] "/home/guest/R/EDA_Spring2024"
```

```
Litter_Processed <- read.csv(
  file=here(
    "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)

PeterPaul_Processed <- read.csv(
  file=here(
    "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE
)
#2

Litter_Processed$collectDate <- ymd(Litter_Processed$collectDate)
PeterPaul_Processed$sampldate <- ymd(PeterPaul_Processed$sampldate)
```

## 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
my_theme <- #theme_base() +
  theme(
    line = element_line(
      color='black',
      linewidth =1
    ),
    legend.background = element_rect(
      color='grey',
      fill = 'white'
    ),
    plot.background = element_rect(
      color='lightblue'
    ),
    plot.title = element_text(
      color = "darkgreen",
      size = 12,
      hjust = 0.5),
    legend.title = element_text(
      color='darkgreen'
    )
  )
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.

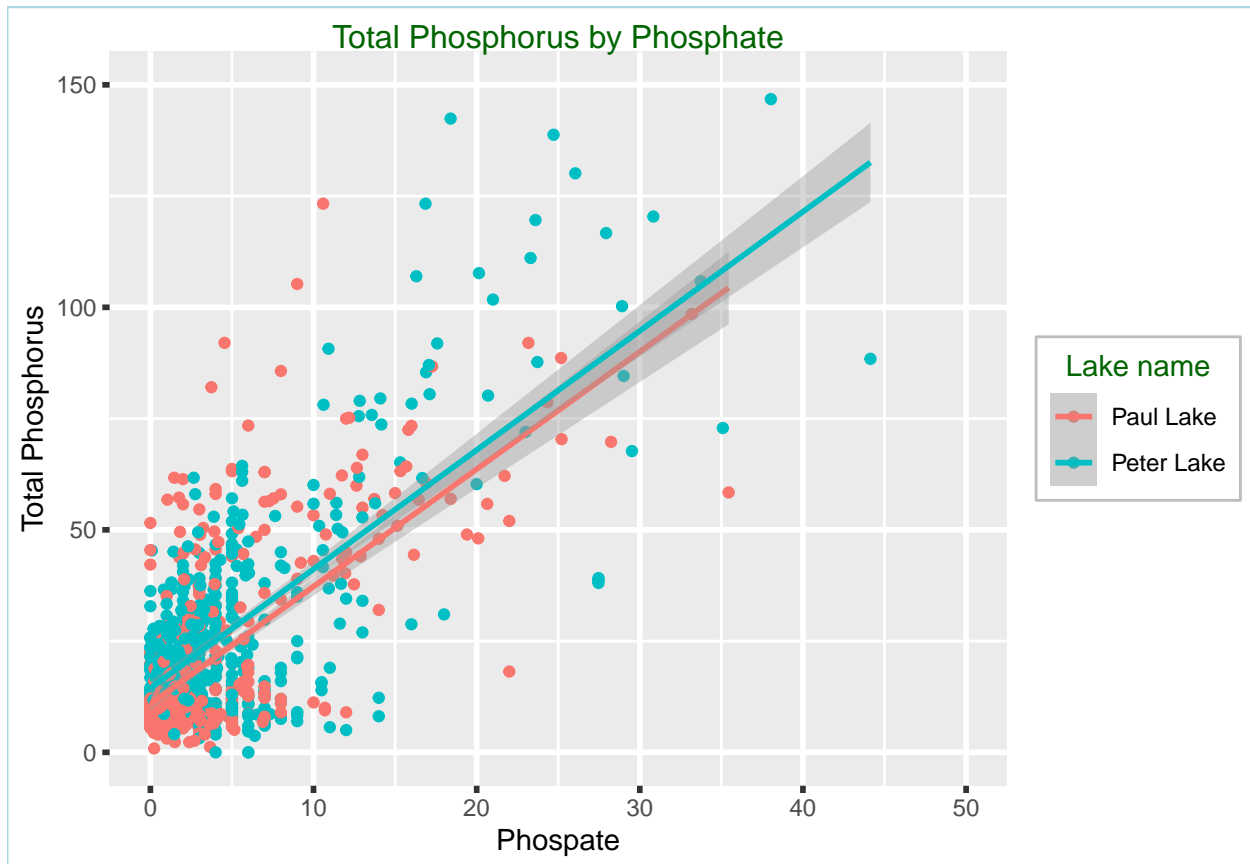
4. [NTL-LTER] Plot total phosphorus (**tp\_ug**) by phosphate (**po4**), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the **lm** method. Adjust your axes to hide extreme values (hint: change the limits using **xlim()** and/or **ylim()**).

```
#4
phosphorusgraph <- ggplot(
  subset(
    PeterPaul_Processed, lakename == "Peter Lake" | lakename == "Paul Lake"),
  aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm") + # Add regression line
  ggtitle("Total Phosphorus by Phosphate") +
  labs(x = "Phosphate", y = "Total Phosphorus", color = "Lake name") +
  xlim(0, 50) + # Set limits for x-axis
  ylim(0, 150) # Set limits for y-axis
print(phosphorusgraph)
```

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

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

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

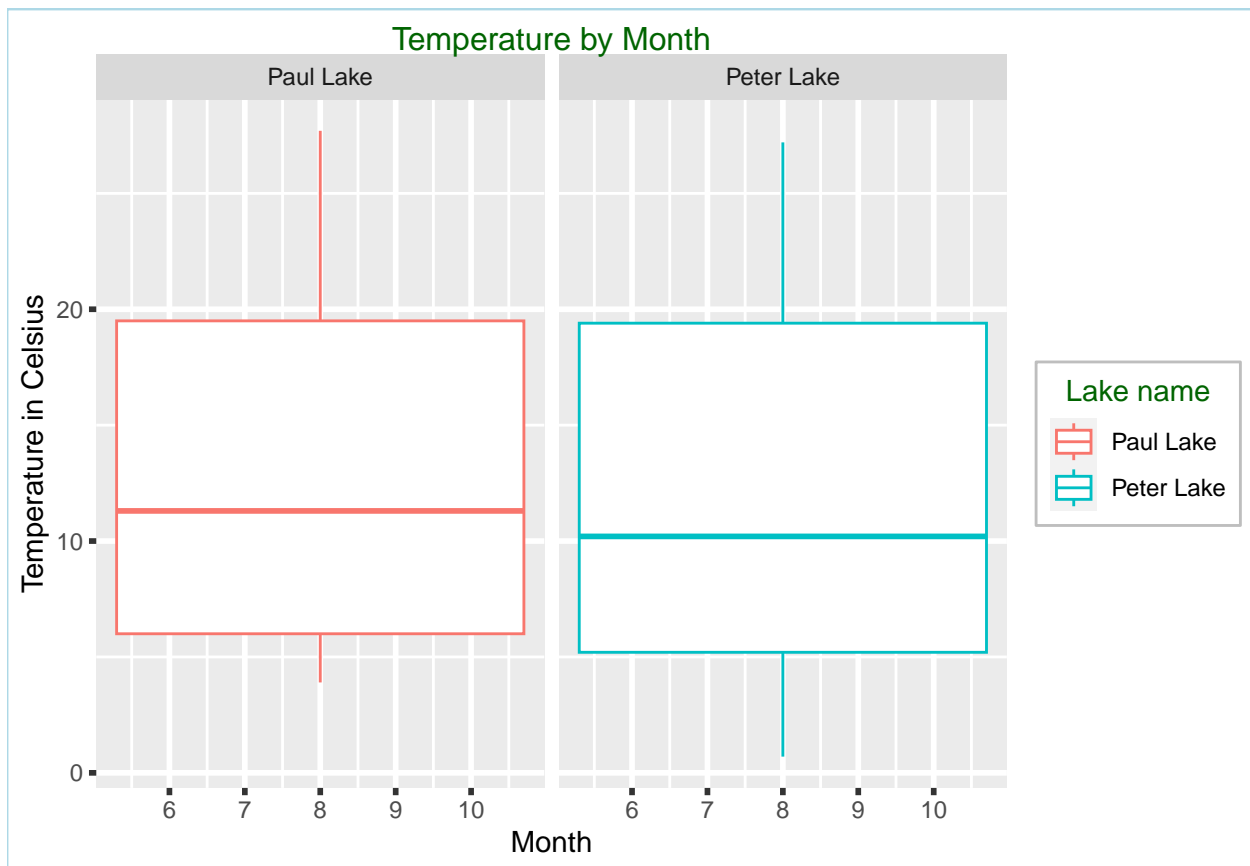


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

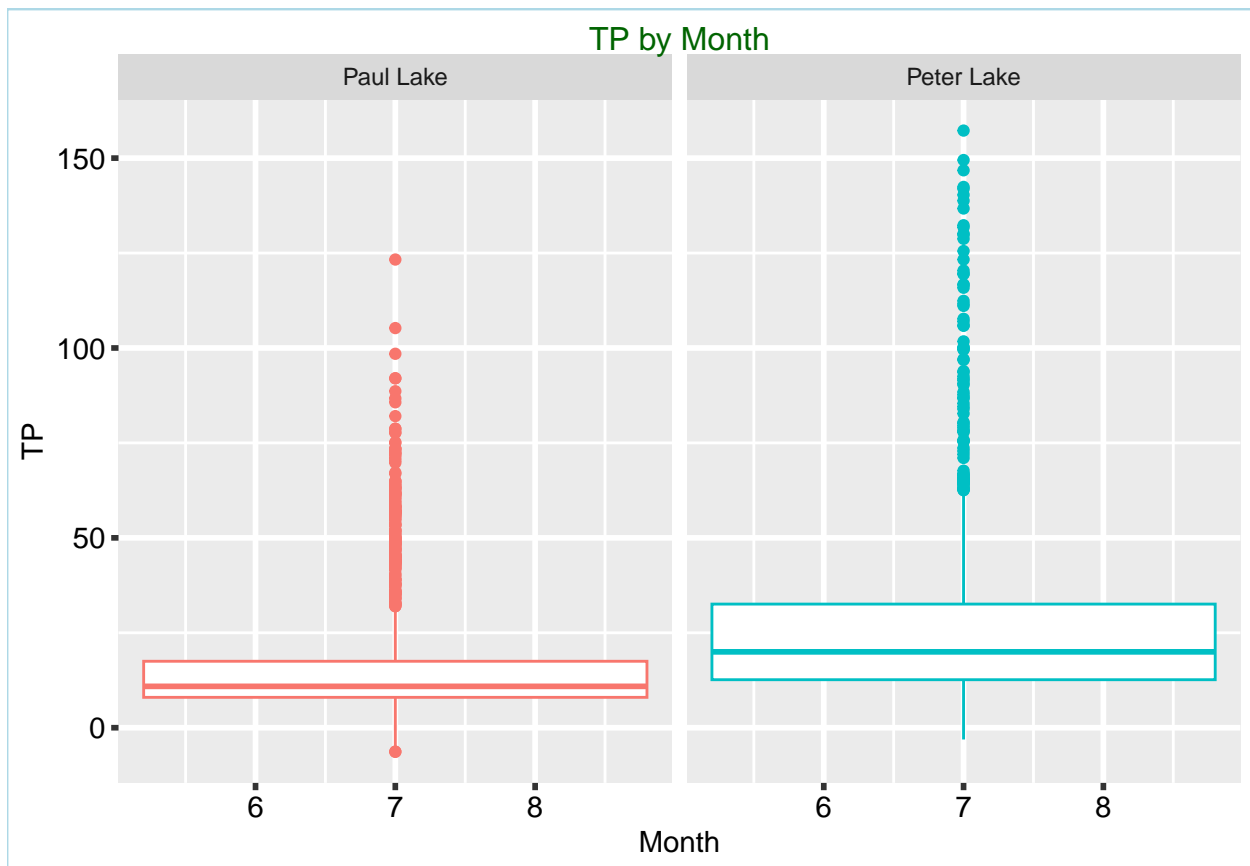
```
#5
plot1_temp <- ggplot(
  PeterPaul_Processed, aes(
    x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  ggtitle("Temperature by Month") +
  labs(x = "Month", y = "Temperature in Celsius", color = "Lake name") +
  facet_wrap(vars(lakename))
print(plot1_temp)
```

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



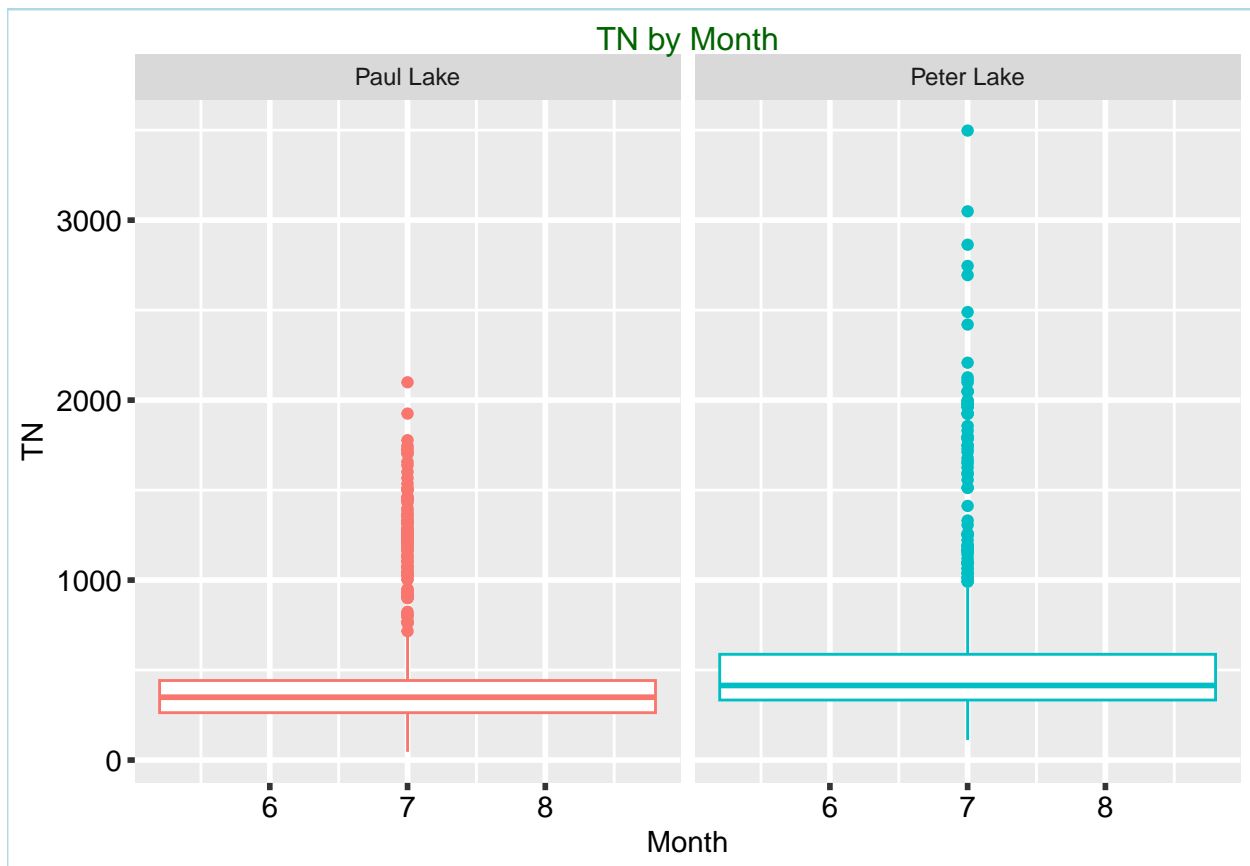
```
plot2_tp <- ggplot(PeterPaul_Processed, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  ggtitle("TP by Month") +
  labs(x = "Month", y = "TP", color = "Lake name") +
  facet_wrap(vars(lakename)) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "none")
print(plot2_tp)
```

## Warning: Removed 20729 rows containing non-finite values ('stat\_boxplot()').



```
plot3_tn <- ggplot(PeterPaul_Processed, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  ggtitle("TN by Month") +
  labs(x = "Month", y = "TN", color = "Lake name") +
  facet_wrap(vars(lakename)) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "none")
print(plot3_tn)
```

## Warning: Removed 21583 rows containing non-finite values ('stat\_boxplot()').



```
# Arrange plots using cowplot
plot_grid(plot1_temp, plot2_tp, plot3_tn, nrow = 3, align = 'hv', rel_heights = c(2, 2, 2))
```

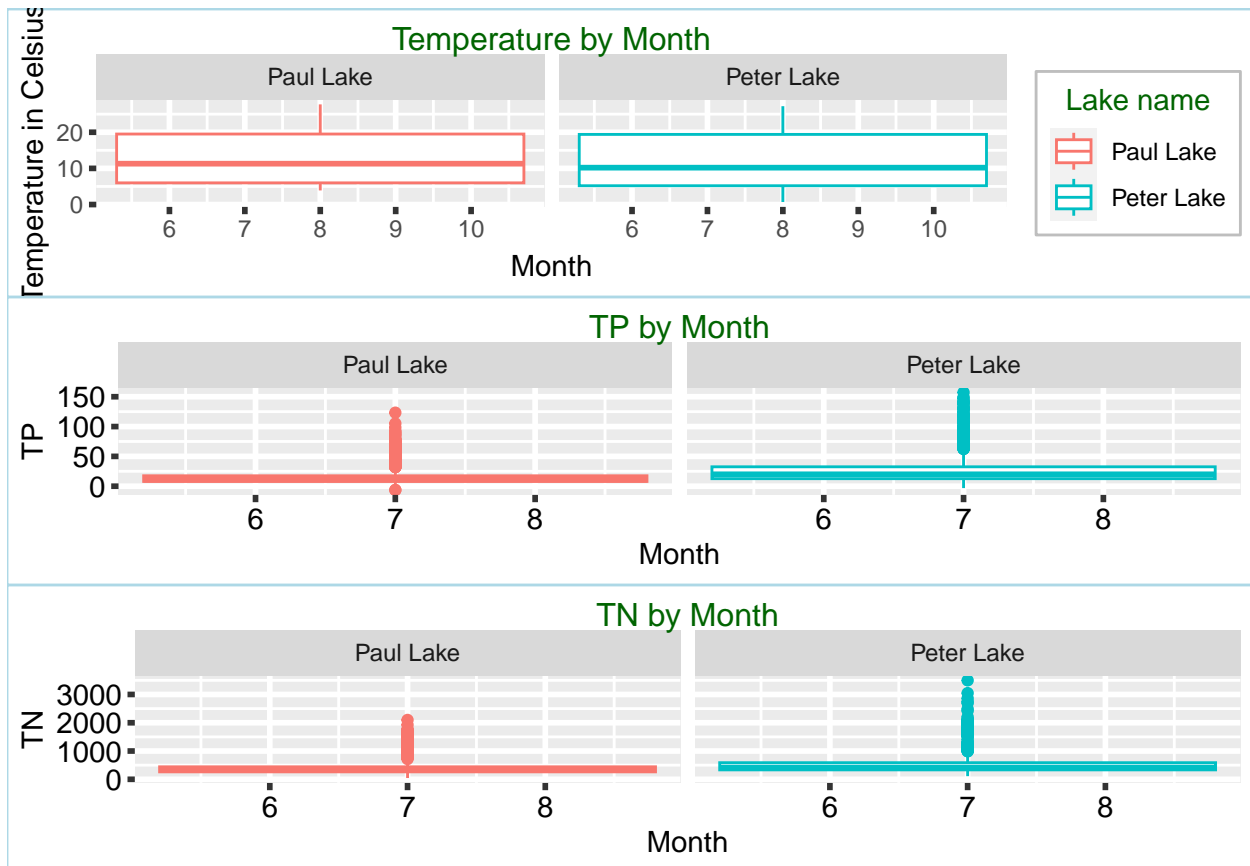
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```

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

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

```
## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
```

```
## Placing graphs unaligned.
```



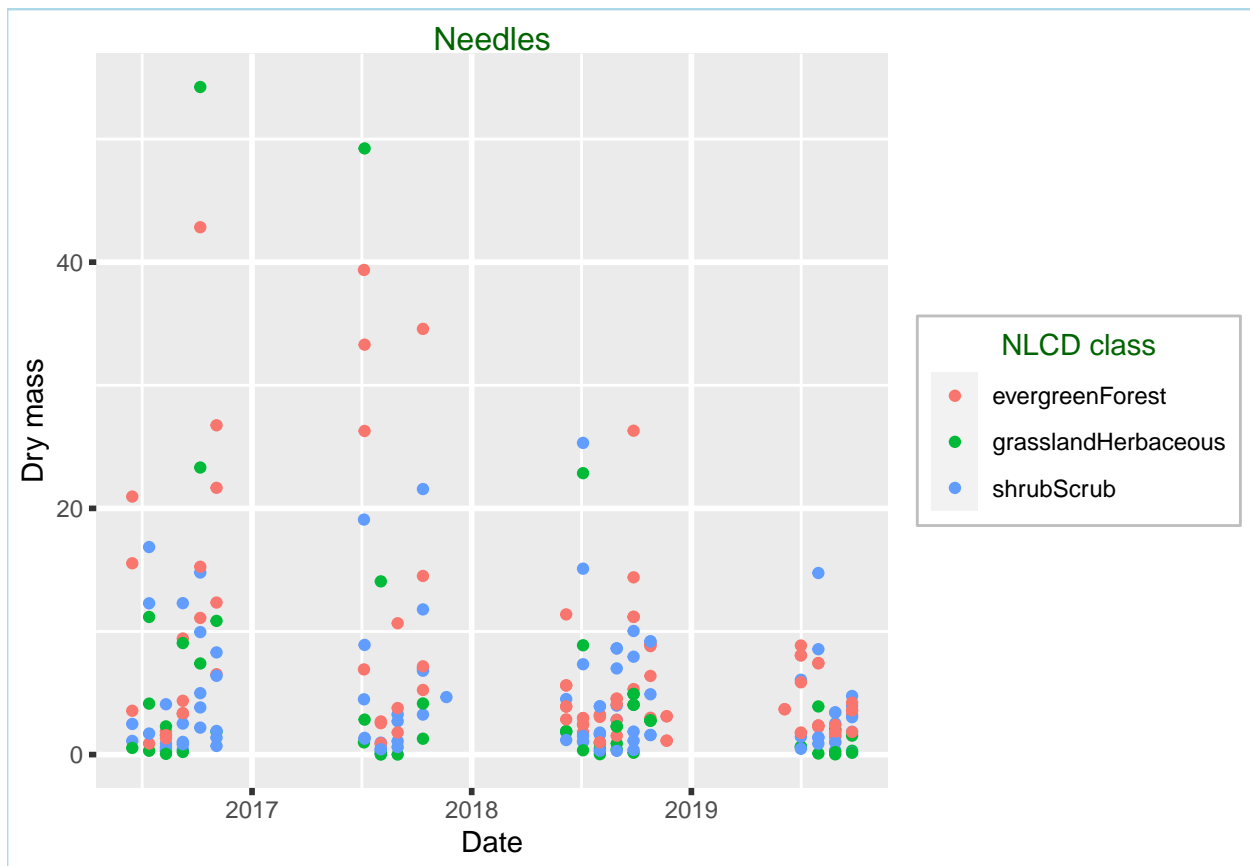
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: In the Summer, as the temperature increases, the TN and TP increase as well. The Peter Lake has an overall higher concentration of TN and TP, compared to the Paul Lake.

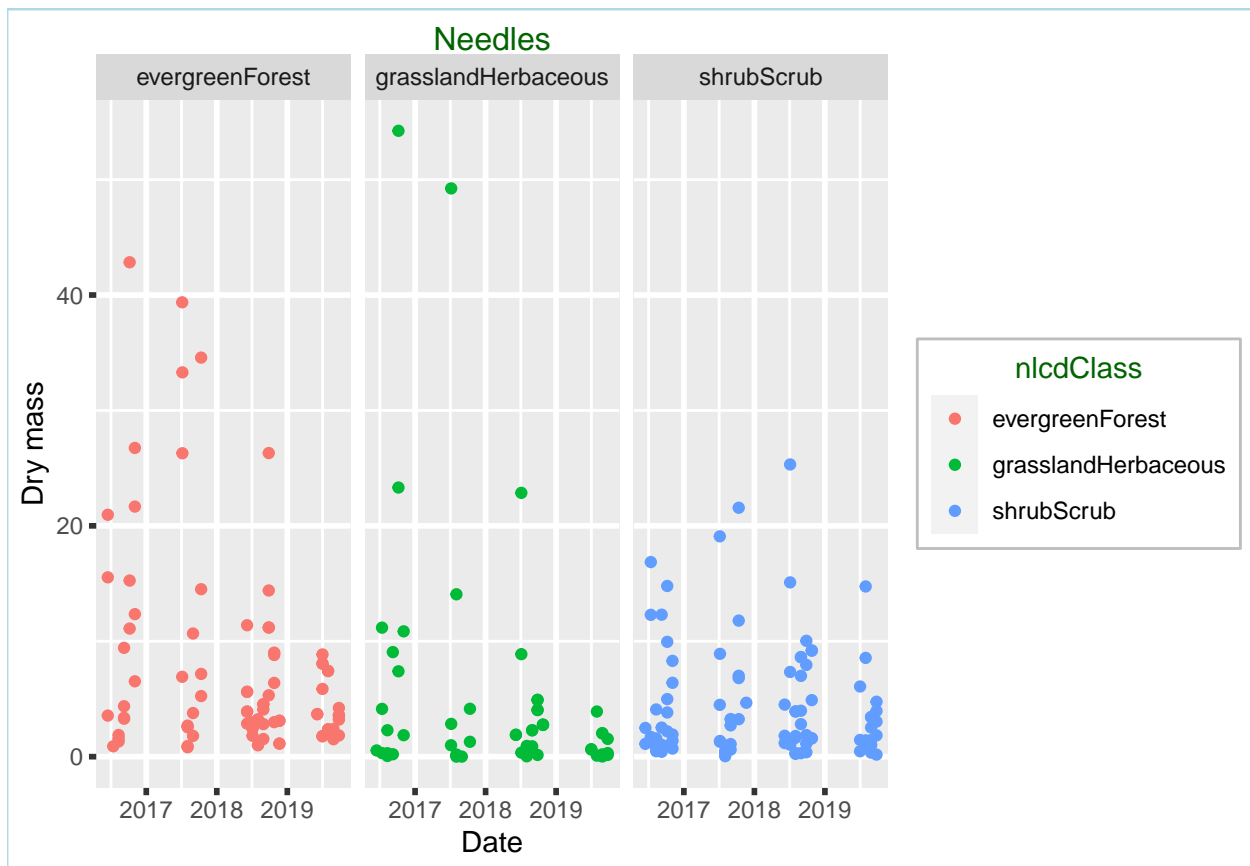
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the “Needles” functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
needles_plot <- ggplot(
  subset(Litter_Processed,
    functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  ggtitle("Needles") +
  labs(x = "Date", y = "Dry mass", color = "NLCD class")
print(needles_plot)
```





```
#7
needles_separated_plot <- ggplot(
  subset(Litter_Processed,
    functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  ggtitle("Needles") +
  labs(x = "Date", y = "Dry mass") +
  facet_wrap(vars(nlcdClass))
print(needles_separated_plot)
```



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

Answer: Plot 7 is more effective, because it shows the NLCD classes as different graphs, and the points are not packed in the same place.