Assignment 5: Data Visualization

Prisha

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 Setting up
library(tidyverse); library(lubridate); library(here); library(ggplot2); 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   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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 /Users/prisha/Desktop/EDE/EDE 2025
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

```
## [1] "/Users/prisha/Desktop/EDE/EDE 2025"
```

```
PeterPaul.chem.nutrients <-
  read.csv(here("Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
Litter_data <-
  read.csv(here("Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)

#2 Dates
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Litter_data$collectDate <- ymd(Litter_data$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 Building a theme
mytheme <- theme_minimal(base_size = 13) +
  theme(axis.text = element_text(color = "darkblue"),
        axis.ticks = element_line(colour = "black"),
        plot.title.position = "plot",
        legend.position = "right")
```

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()`).

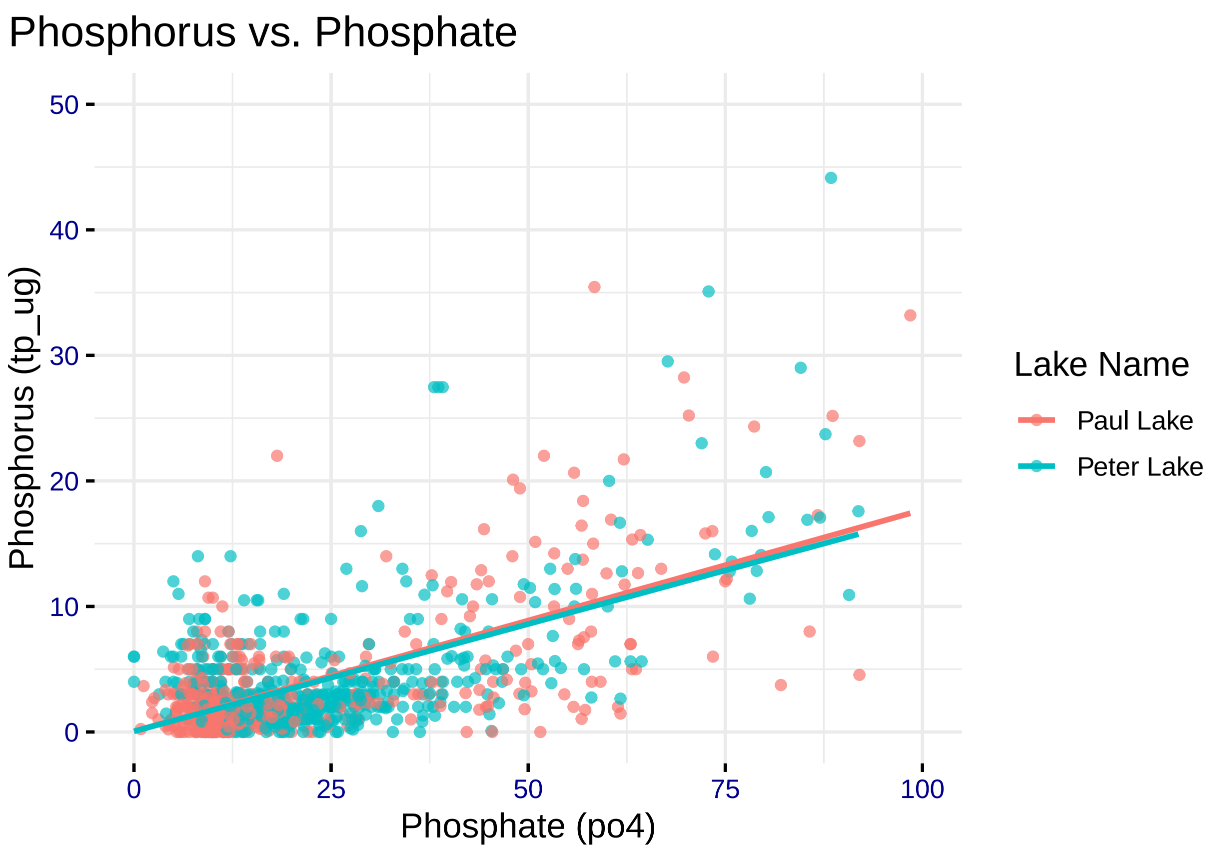
```r
#4 Plot 1
peter_paul_plot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4, colour = lakename)) +
         geom_point(alpha=0.7) +
         geom_smooth(method = "lm", se = FALSE) +
  xlim(0, 100) +
  ylim(0, 50) +
  labs(title = "Phosphorus vs. Phosphate",
       x = "Phosphate (po4)",
       y = "Phosphorus (tp_ug)",
       color = "Lake Name") +
  mytheme

print(peter_paul_plot1)
```

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

## Warning: Removed 21964 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 21964 rows containing missing values or values outside the scale range
## (`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 Plot 2

class(PeterPaul.chem.nutrients$month)
```

```
## [1] "integer"
```

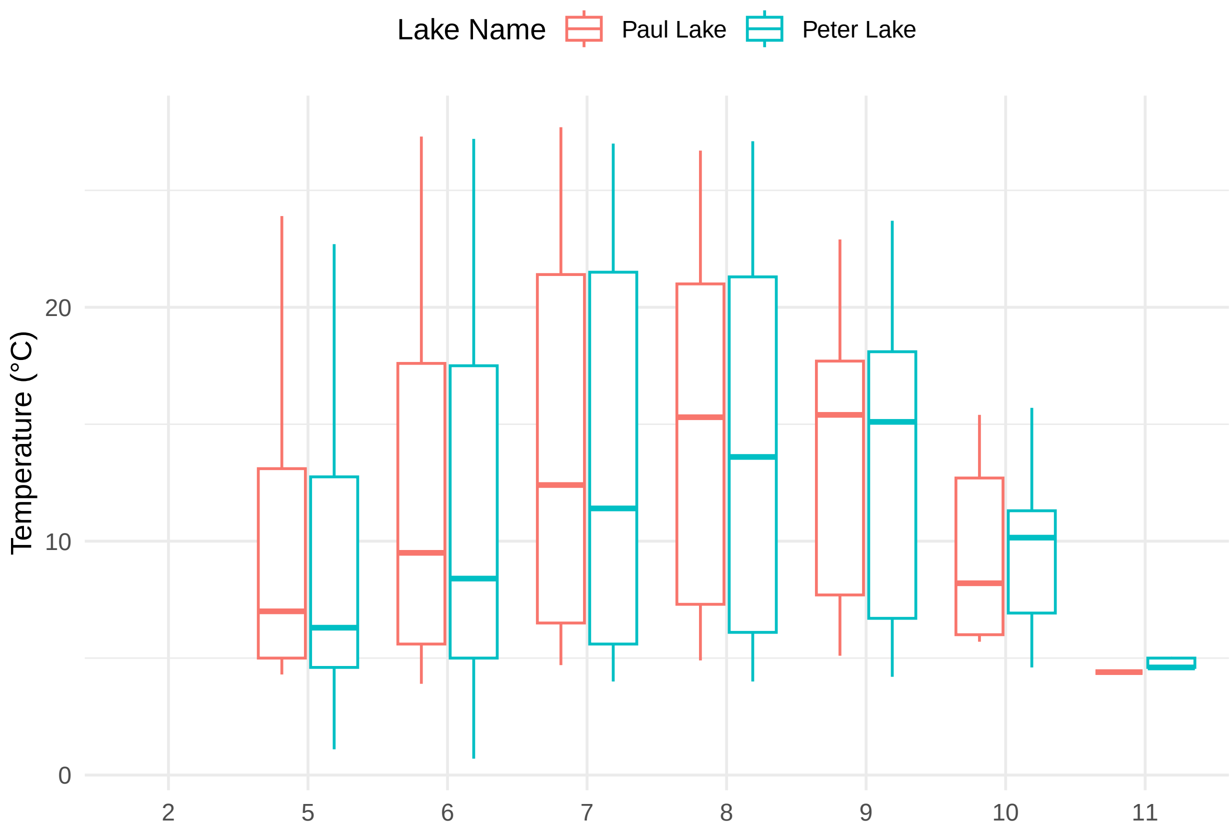
```
PeterPaul.chem.nutrients$month <- as.factor(PeterPaul.chem.nutrients$month)
class(PeterPaul.chem.nutrients$month)
```

```
## [1] "factor"
```

```
Boxplot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C, colour = lakename)) +
  geom_boxplot() +
  labs(y = "Temperature (°C)",
       colour = "Lake Name") +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    legend.position = "top")

print(Boxplot1)
```

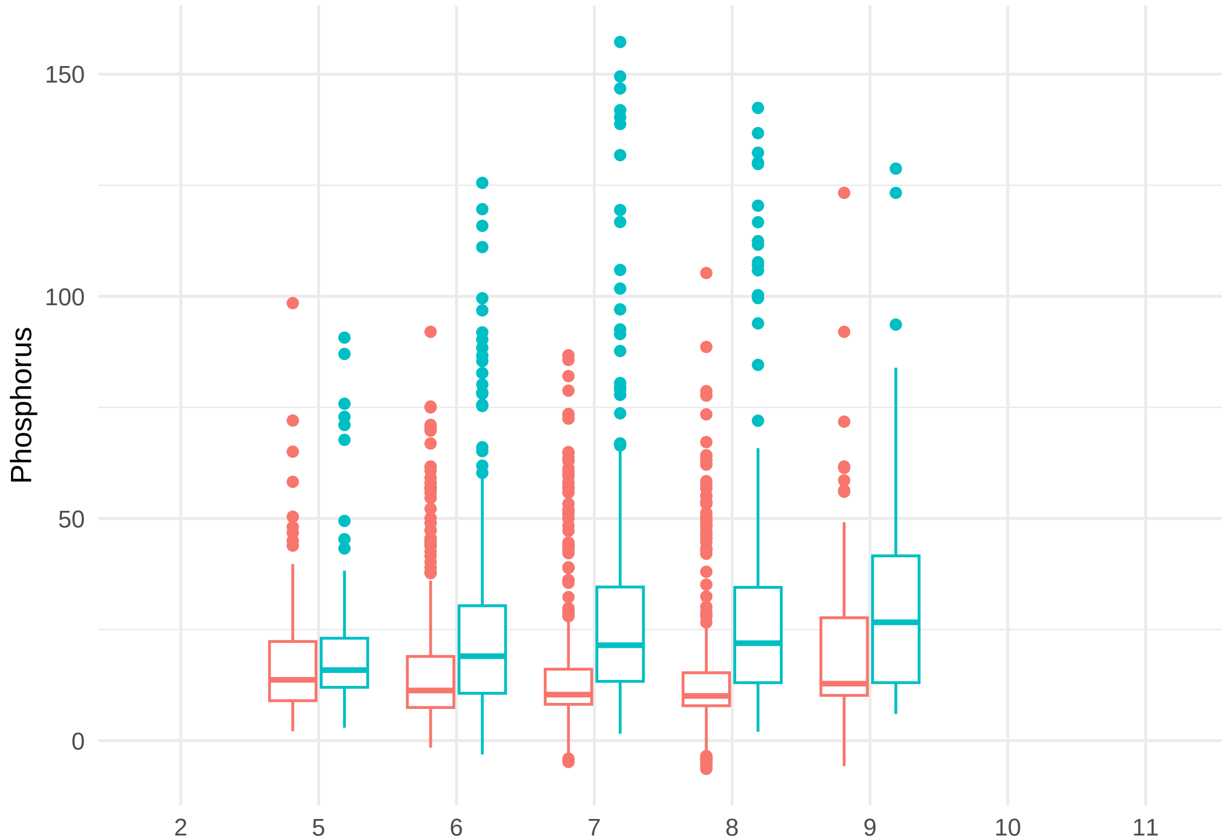
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```r
Boxplot2 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug, colour = lakename)) +
  geom_boxplot() +
  labs(y = "Phosphorus") +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    legend.position = "none")

print(Boxplot2)
```

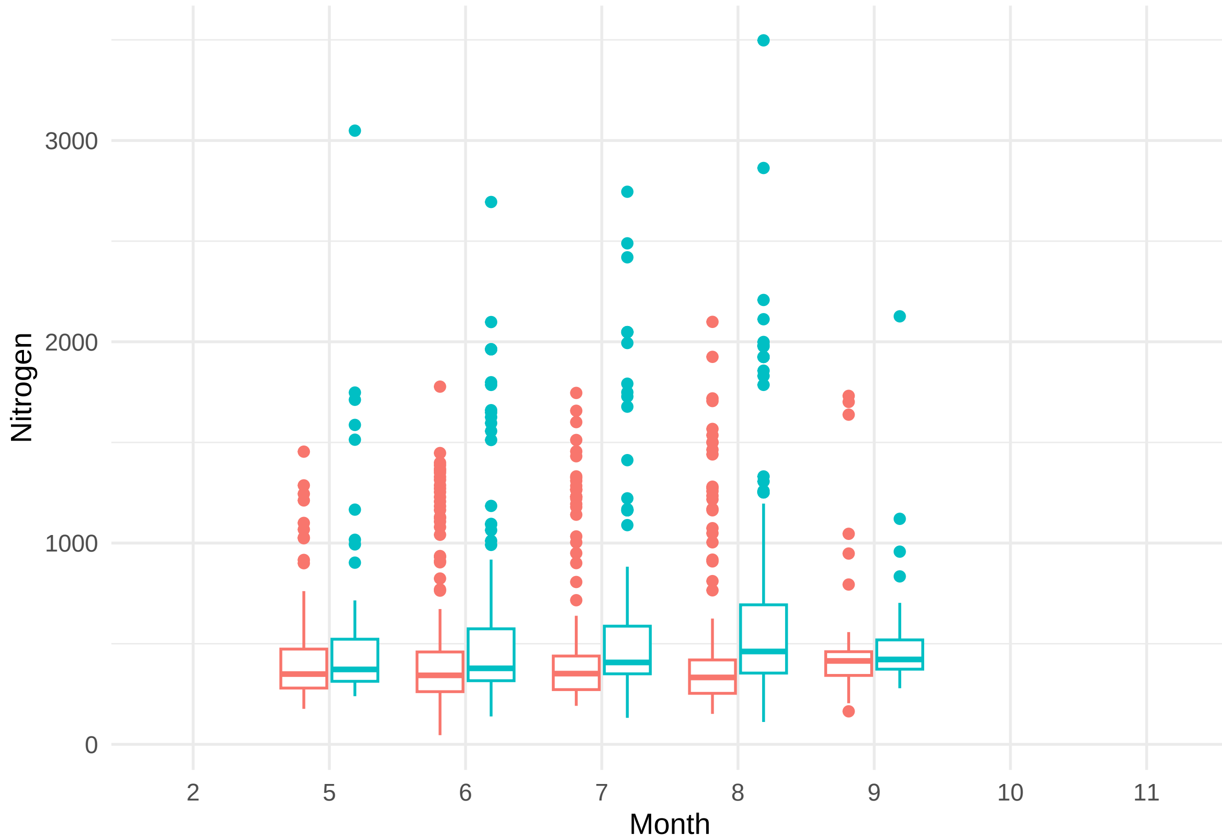
```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
Boxplot3 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug, colour = lakename)) +
  geom_boxplot() +
  labs(y = "Nitrogen", x = "Month") +
  theme_minimal() +
  theme(
    legend.position = "none")

print(Boxplot3)
```

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



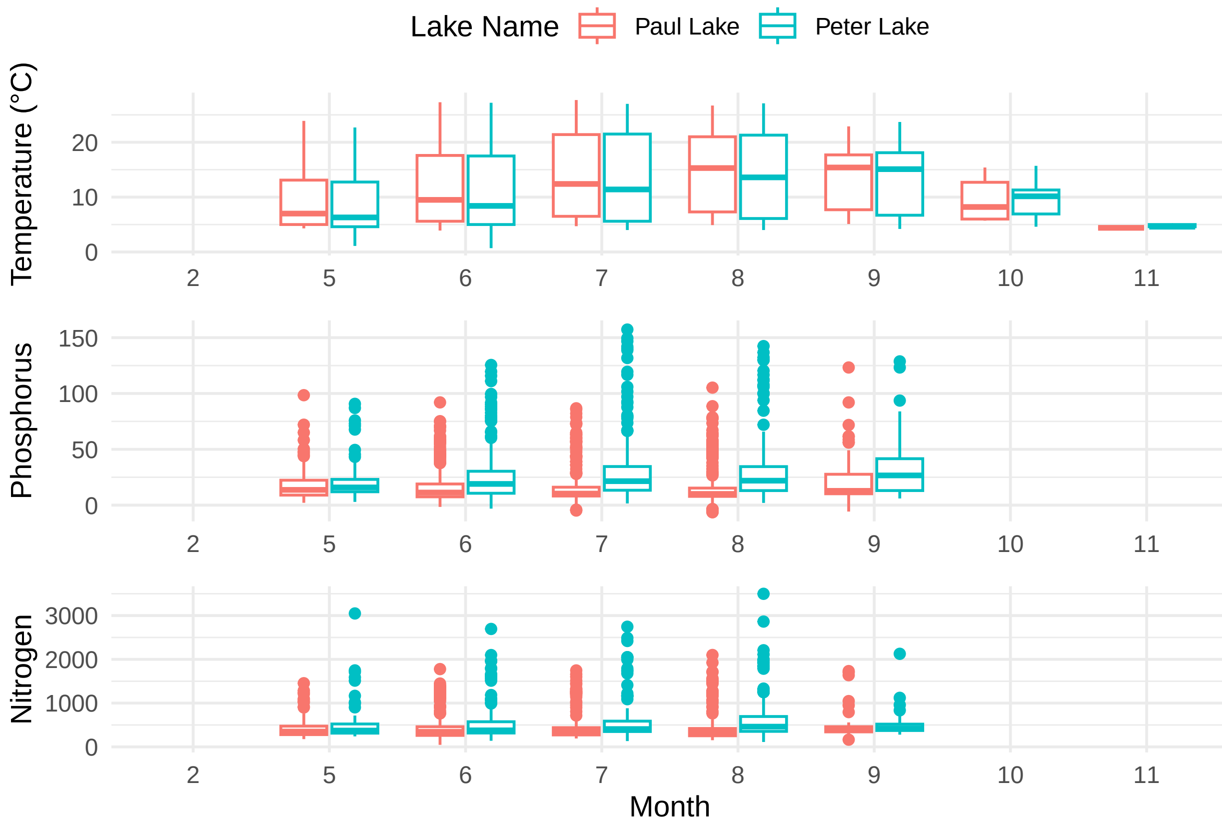

```r
final_plot <-
  plot_grid(Boxplot1, Boxplot2, Boxplot3, nrow = 3, align = 'v', rel_heights = c(1.25, 1,1))
```

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

```r
print(final_plot)
```



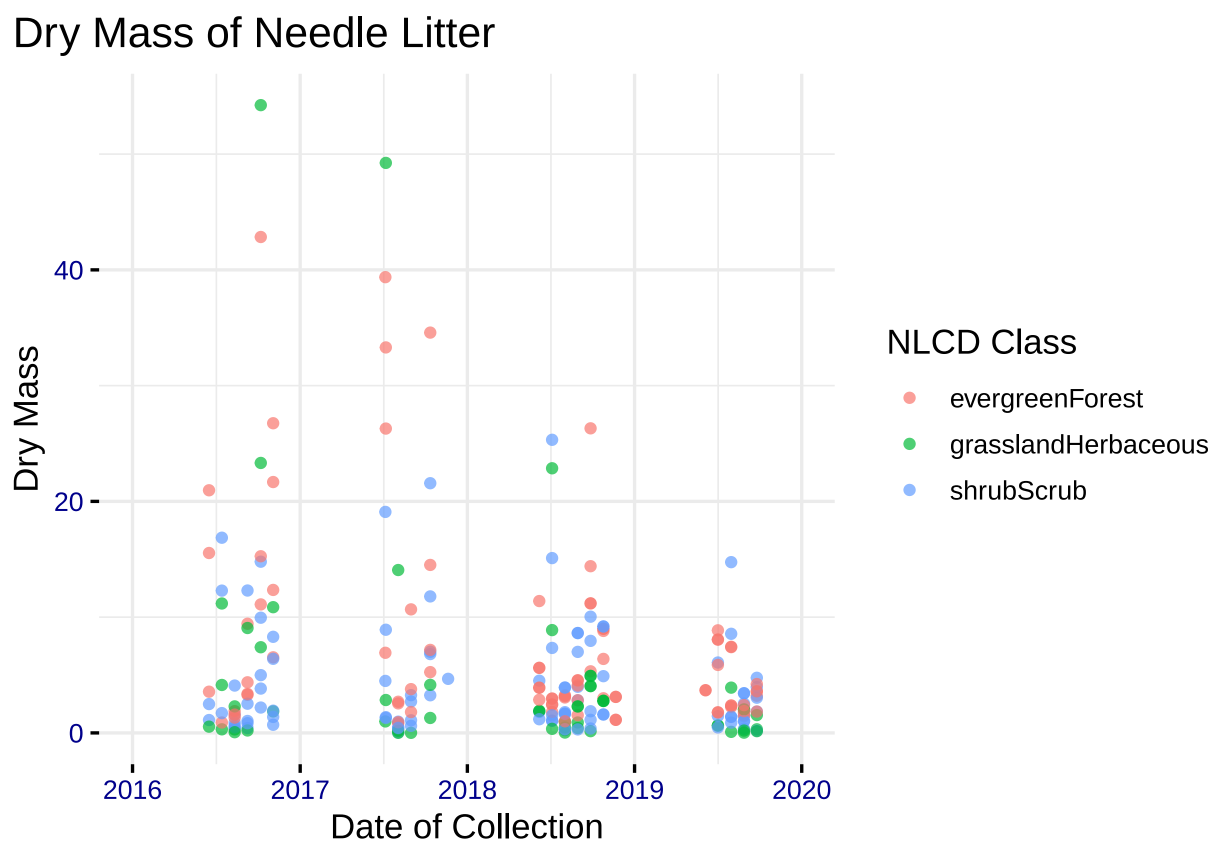
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: We can see that temperature follows a clear seasonal trend, increasing from spring to summer and declining in fall, with similar patterns in both lakes. Paul Lake generally has higher phosphorus and nitrogen levels than Peter Lake, suggesting greater nutrient input or retention. Nutrient levels show high variability and outliers, indicating occasional spikes likely due to runoff or biological activity.

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.

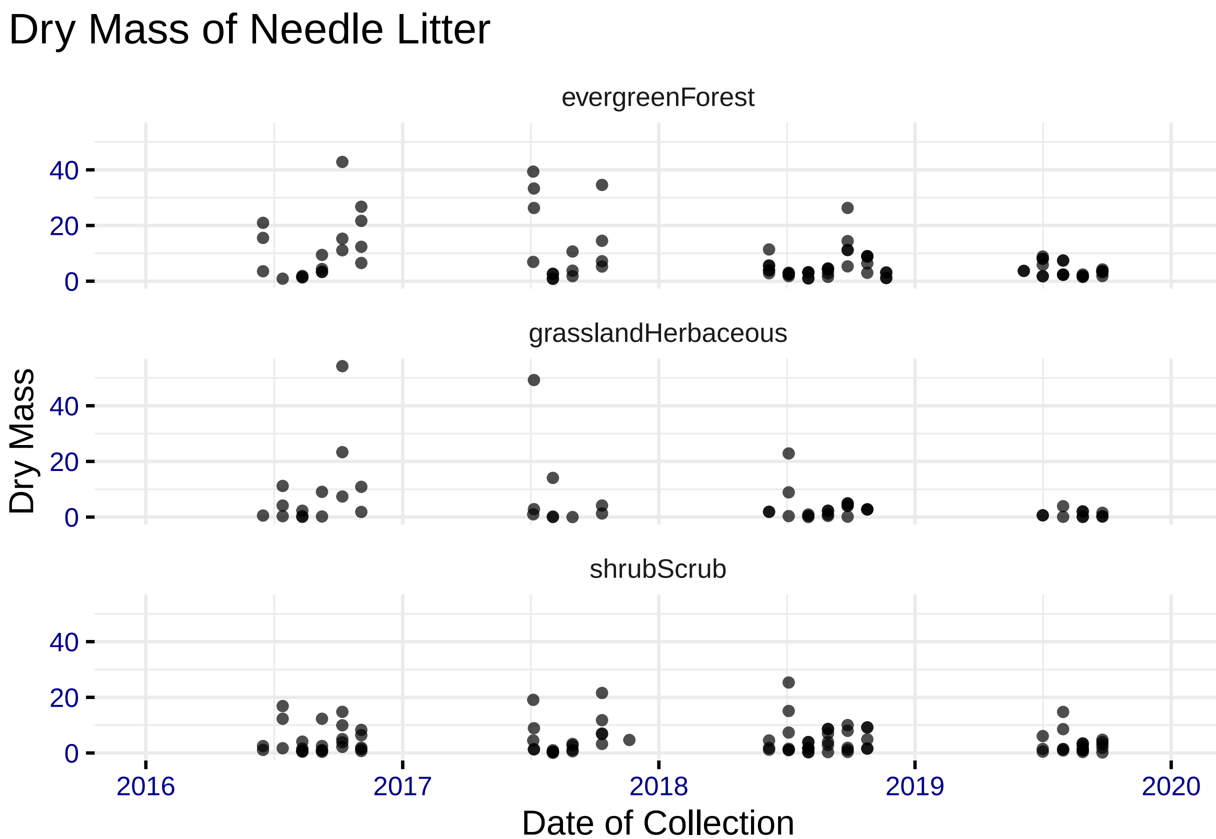
```
#6 Plot 3
needles_plot <-
  ggplot(subset(Litter_data, functionalGroup == "Needles"),
         aes(y = dryMass, x = collectDate, color = nlcdClass)) +
  geom_point(alpha = 0.7) +
  labs(y = "Dry Mass",
       x = "Date of Collection",
       color = "NLCD Class",
       title = "Dry Mass of Needle Litter") +
  scale_x_date(limits = as.Date(c("2016-01-01", "2019-12-31"))) +
  mytheme
```

```r
print(needles_plot)
```




```r
#7 Plot 4
needles_facet_plot <-
  ggplot(subset(Litter_data, functionalGroup == "Needles"),
         aes(y = dryMass, x = collectDate)) +
  geom_point(alpha = 0.7) +
  labs(y = "Dry Mass",
       x = "Date of Collection",
       title = "Dry Mass of Needle Litter") +
  scale_x_date(limits = as.Date(c("2016-01-01", "2019-12-31"))) +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  mytheme

print(needles_facet_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 separates each NLCD class into its own panel, making it easier to compare trends within each land cover type without overlapping points. In contrast, Plot 6 can be cluttered, especially when data points from different classes overlap, making it harder to distinguish patterns.