Assignment 5: Data Visualization

Rosie Wu

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.

```r
#1
# Import basic libraries
library(tidyverse);library(lubridate);library(here)
```

```
## -- 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
## here() starts at /home/guest/EDE_Fall2024
```

```
#Library for creating ridge plots
library(ggridges)
#Enhanced color ramps
library(viridis)
```

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

```
library(RColorBrewer)
library(colormap)
#Add pre-set themes <--NEW
library(ggthemes)

# verify home
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

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

#2 format date using lubridate ymd
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Litter_mass_trap$collectDate <- ymd(Litter_mass_trap$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 build a theme using theme_classic
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        plot.title = element_text(size = 16, face = "bold", hjust = 0.5), #title
        axis.title.x = element_text(size = 12, color = "black"), # X-axis title
        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.

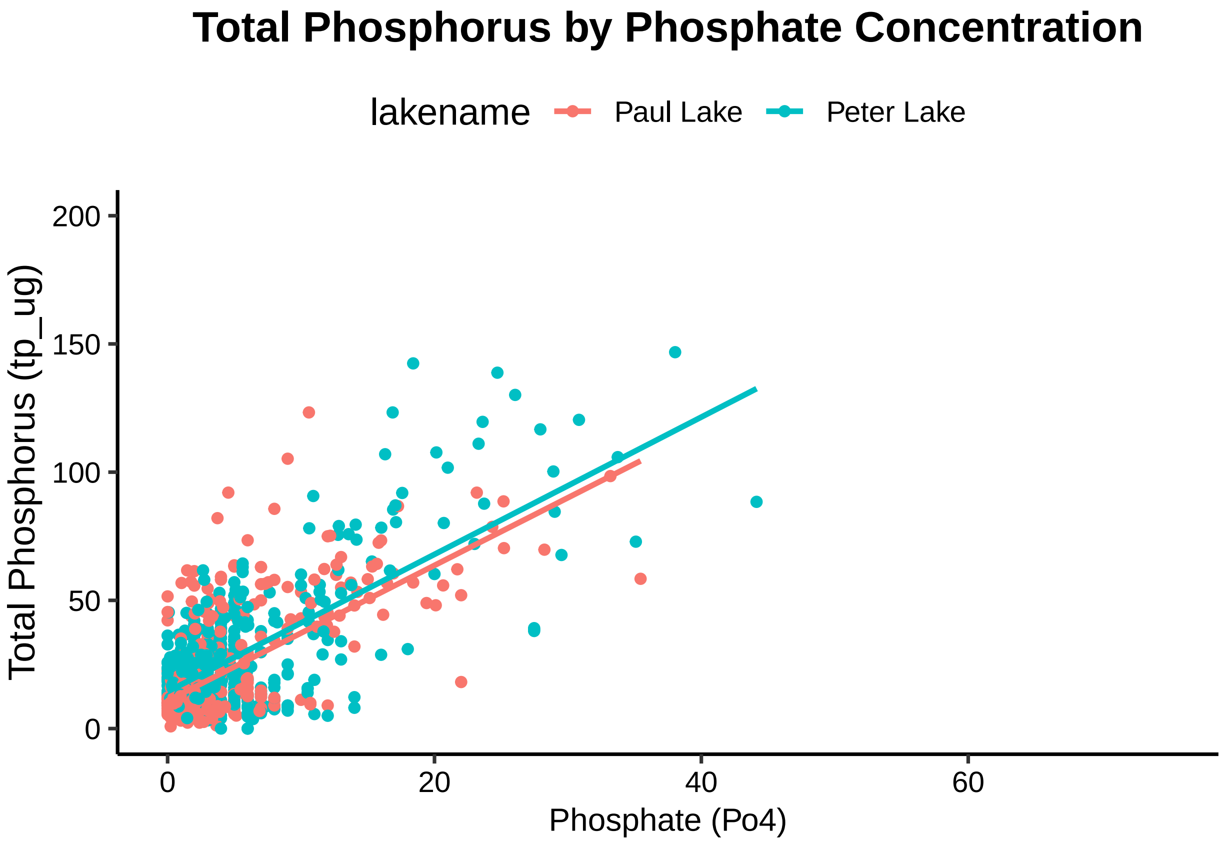
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4 build plot, y = tp_ug, x = po4
q4_plot <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +  # Scatter points
  geom_smooth(method = "lm", se = FALSE) +  # Line of best fit
  labs(title = "Total Phosphorus by Phosphate Concentration",
    x = "Phosphate (Po4)",
    y = "Total Phosphorus (tp_ug)")+
  xlim(0, 75) +  # Adjust x and y axis limits after viewing the plot first
  ylim(0, 200) +  mytheme

print(q4_plot)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 21948 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 21948 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

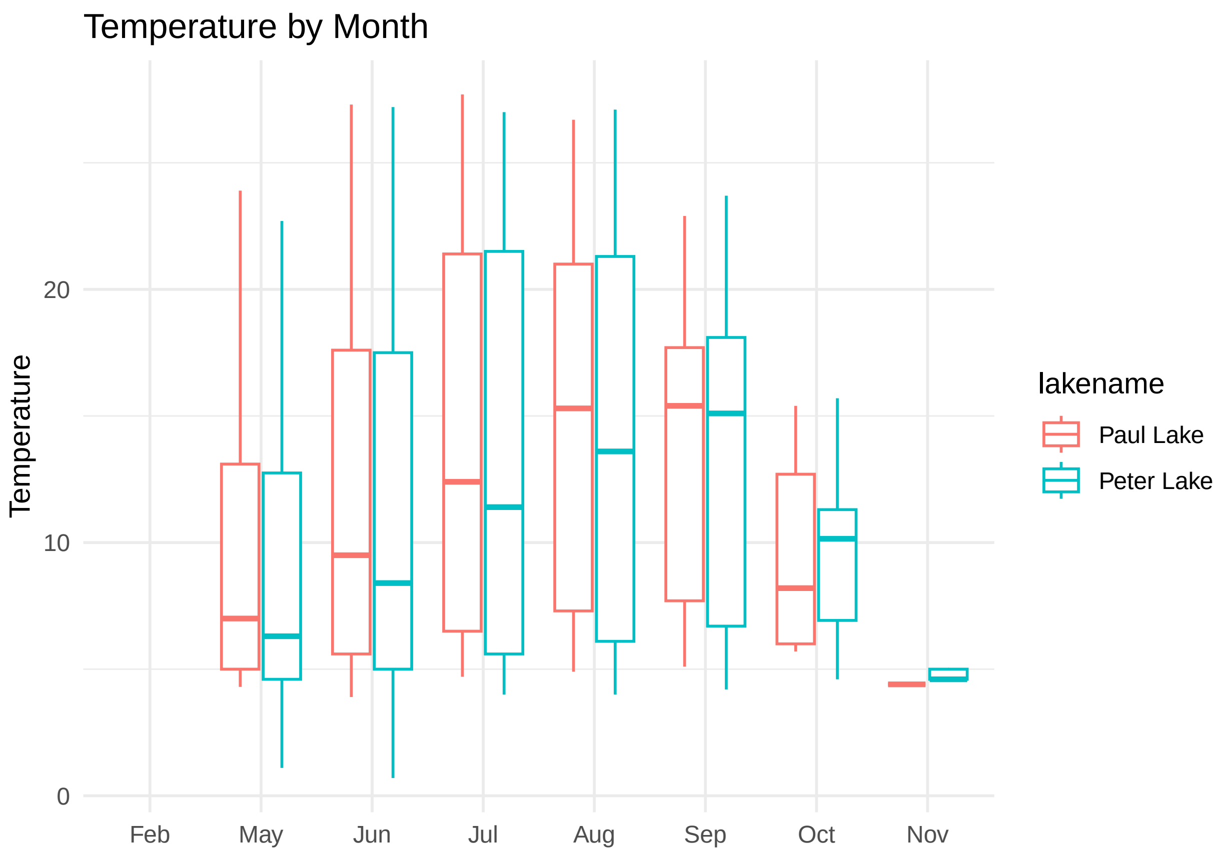


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

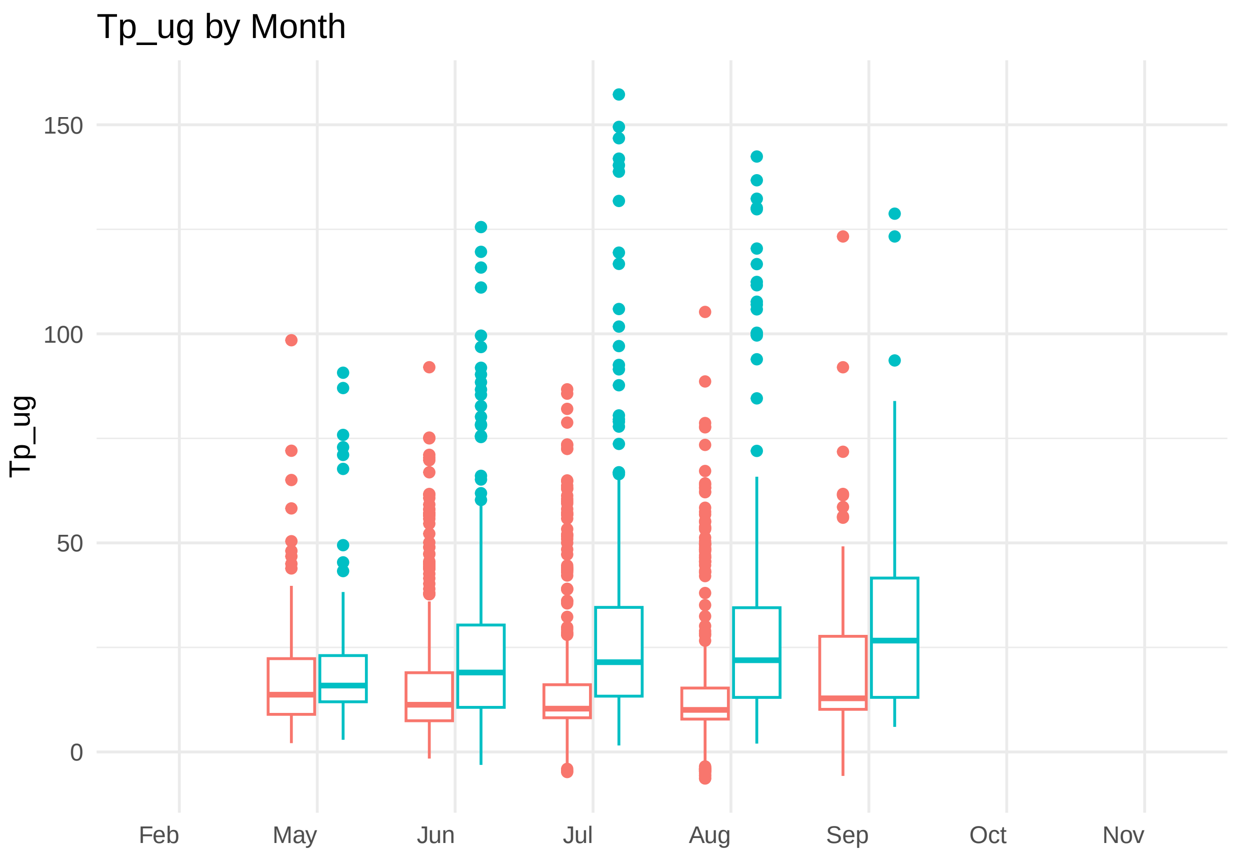
```
#5
#Convert month to a factor -- with 12 levels, labelled with month names
PeterPaul.chem.nutrients$month <- factor(PeterPaul.chem.nutrients$month,
                                         levels = 1:12,
                                         labels = month.abb)

# plot temperature, as x as month, y as temperature, lakename as color asthetic?
temp_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C,
                                       color=lakename)) +
  geom_boxplot() +
  labs(title = "Temperature by Month", y= "Temperature")+
  theme_minimal()+
  theme(
    axis.title.x = element_blank(),  # Remove x-axis title
    legend.position = "right")
print(temp_plot)
```

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



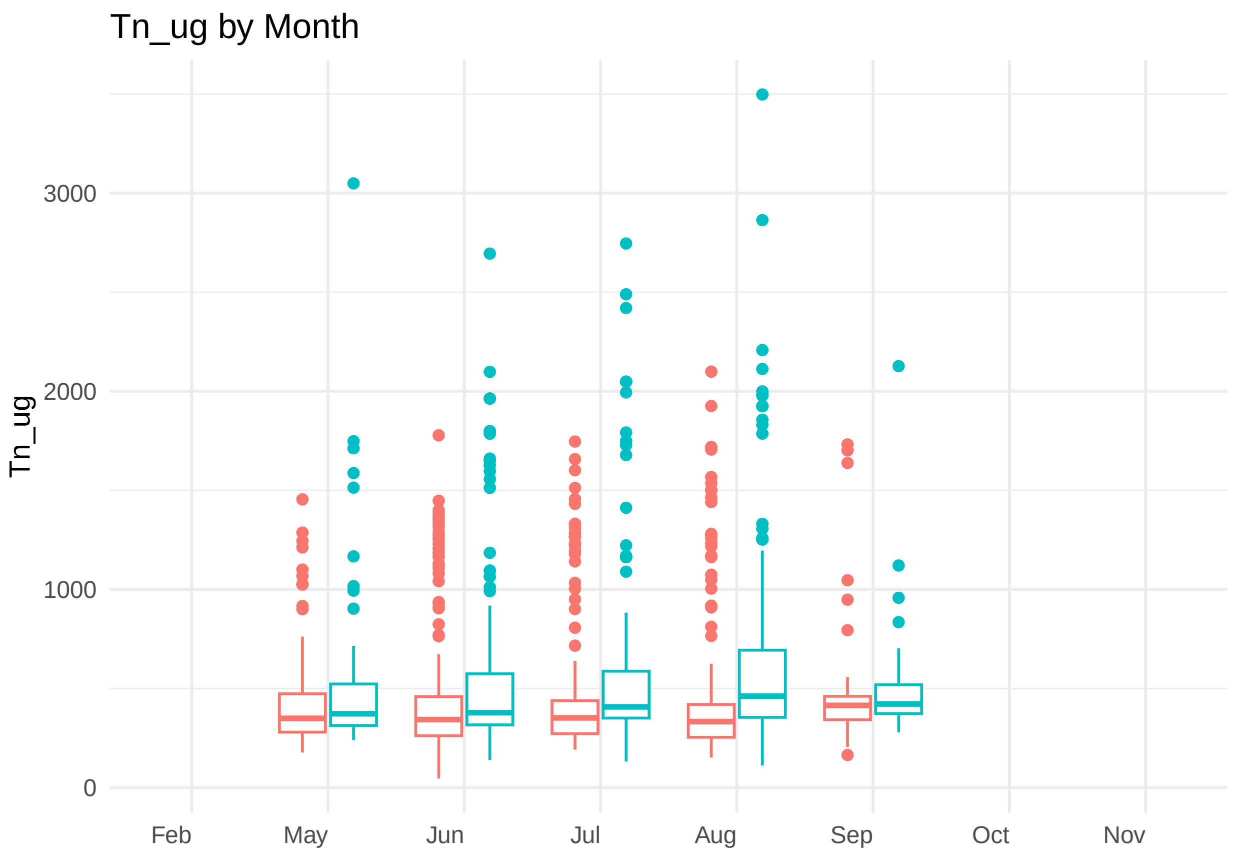
```r
# plot TP, x as month, y as tp_ug, lakename as color asthetic?
TP_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug,
                                       color=lakename)) +
  geom_boxplot() +
  labs(title = "Tp_ug by Month", y = "Tp_ug")+
  theme_minimal()+
  theme(
    axis.title.x = element_blank(),  # Remove x-axis title
    axis.text.x = element_text(hjust = 1),
    legend.position = "none")
print(TP_plot)
```

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



```
# plot TN, x=month, y=tn_ug, lakename as color
TN_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug,
                                         color=lakename)) +
  geom_boxplot() +
  labs(title = "Tn_ug by Month", y = "Tn_ug")+
  theme_minimal()+
  theme(
    axis.title.x = element_blank(),  # Remove x-axis title
    axis.text.x = element_text(hjust = 1),
    legend.position = "none")
print(TN_plot)
```

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

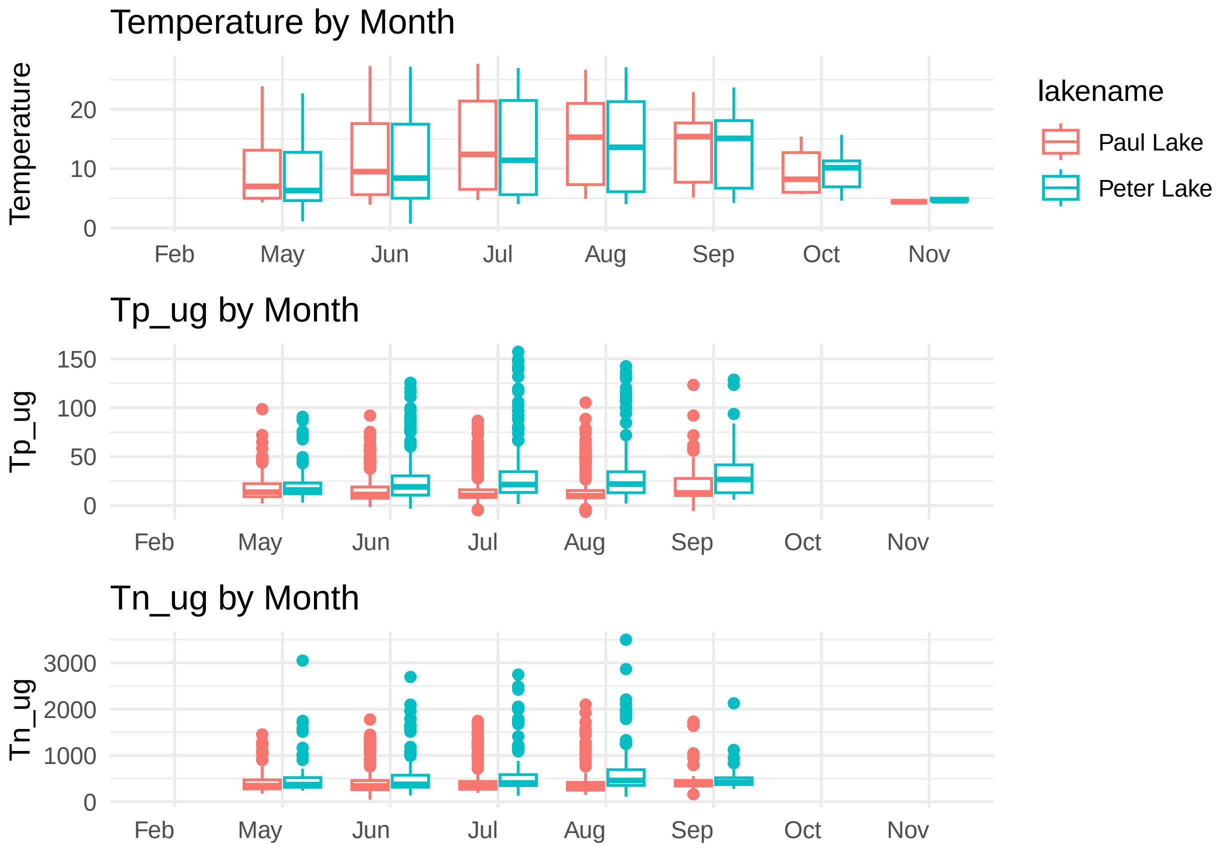


```
# use cowplot to combine all three plots
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':
##
##     theme_map

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

```
plot_grid(temp_plot, TP_plot, TN_plot, nrow = 3, align = 'v',
          rel_heights = c(5, 5, 5))
```

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

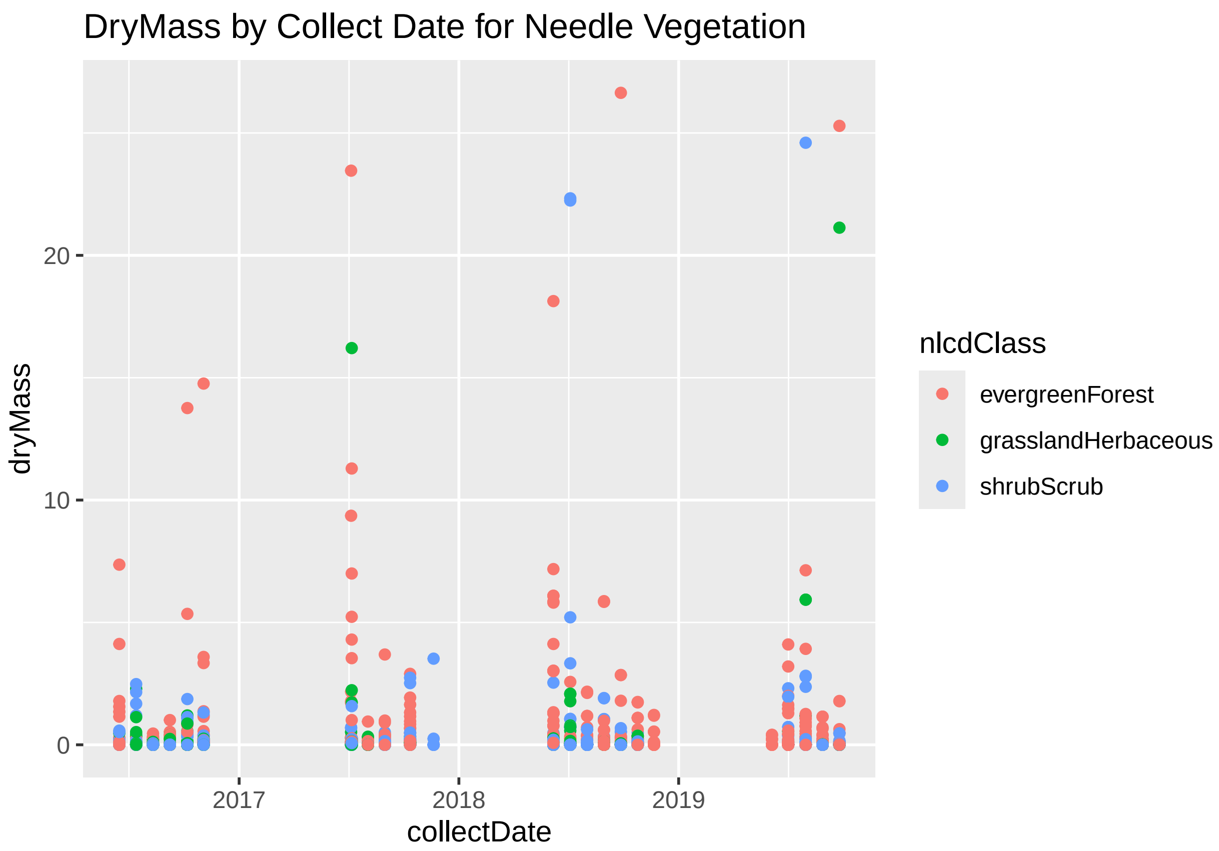


Question: What do you observe about the variables of interest over seasons and between lakes?

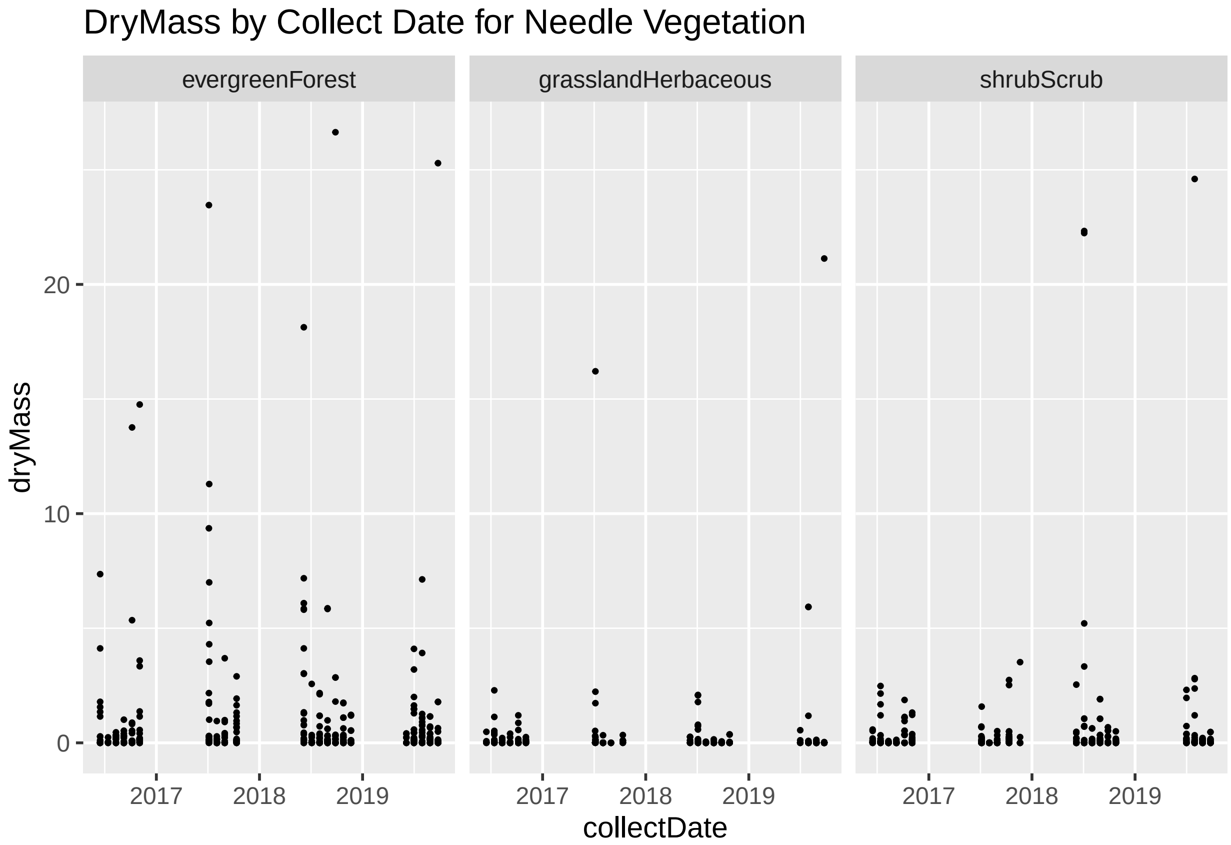
> Answer: The seasonal patterns for temperatures, TN, and, TP are very similar between Peter and Paul lake, which summer (June- Aug) tends to be the warmer months with higher activity/ accumulation in chemicals. The area and depth range for Peter is larger compared to Paul Lake, due to the range can be more variable/ larger in terms of temperature seasonability, as well as chemical contents Tp_ug and Tn_ug.

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.

```r
#6 first filter out only Needle rows
Needle_data <- Litter_mass_trap %>%
  filter(functionalGroup != "Needles")
# Plot the dryMass by collect date, separate by nlcdClass
Needle_plot_q6 <-
  ggplot(Needle_data, aes(x = collectDate, y =dryMass, color = nlcdClass))+
  geom_point()+
  labs(title = "DryMass by Collect Date for Needle Vegetation")
print(Needle_plot_q6)
```



```
#7 Plot the same plot with NLCD classes separated into three facets
Needle_plot_q7 <-
  ggplot(Needle_data, aes(x = collectDate, y =dryMass))+
  facet_wrap(facets = vars(nlcdClass), nrow = 1 , ncol=4)+
  geom_point(size = 0.5)+
  labs(title = "DryMass by Collect Date for Needle Vegetation")
print(Needle_plot_q7)
```



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

> Answer: I think 7 is more effective, since even without colors varying by classes, it's easier to see a side by side comparison between the dryMass trends and ranges of the three nlcd Needle classes. The colored map in q6 may in fact seem less precise as the points from different classes can be overlapping each other and causing confusion when being compared.