Assignment 5: Data Visualization

Yuechen Huang

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 Load packages and data
library(tidyverse)
library(lubridate)
library(ggplot2)
library(here)
library(cowplot)
getwd()
```

```
## [1] "D:/ENV872_DataExploration/ENV872_DataExploration_Fall2023"
```

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

#2 Changed to date format
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(NEON.litter$collectDate)
```

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

```
PeterPaul.chem.nutrients$sampledate <-
  ymd(PeterPaul.chem.nutrients$sampledate)
NEON.litter$collectDate <-
  ymd(NEON.litter$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
Yuechen_theme <- theme_bw(base_size = 13) +
  theme(plot.title = element_text(color = 'black', size = 14, face = 'bold')) +
  theme(axis.text =  element_text(size = rel(0.8))) +
  theme(axis.title = element_text(face = 'italic'))
```

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()`).

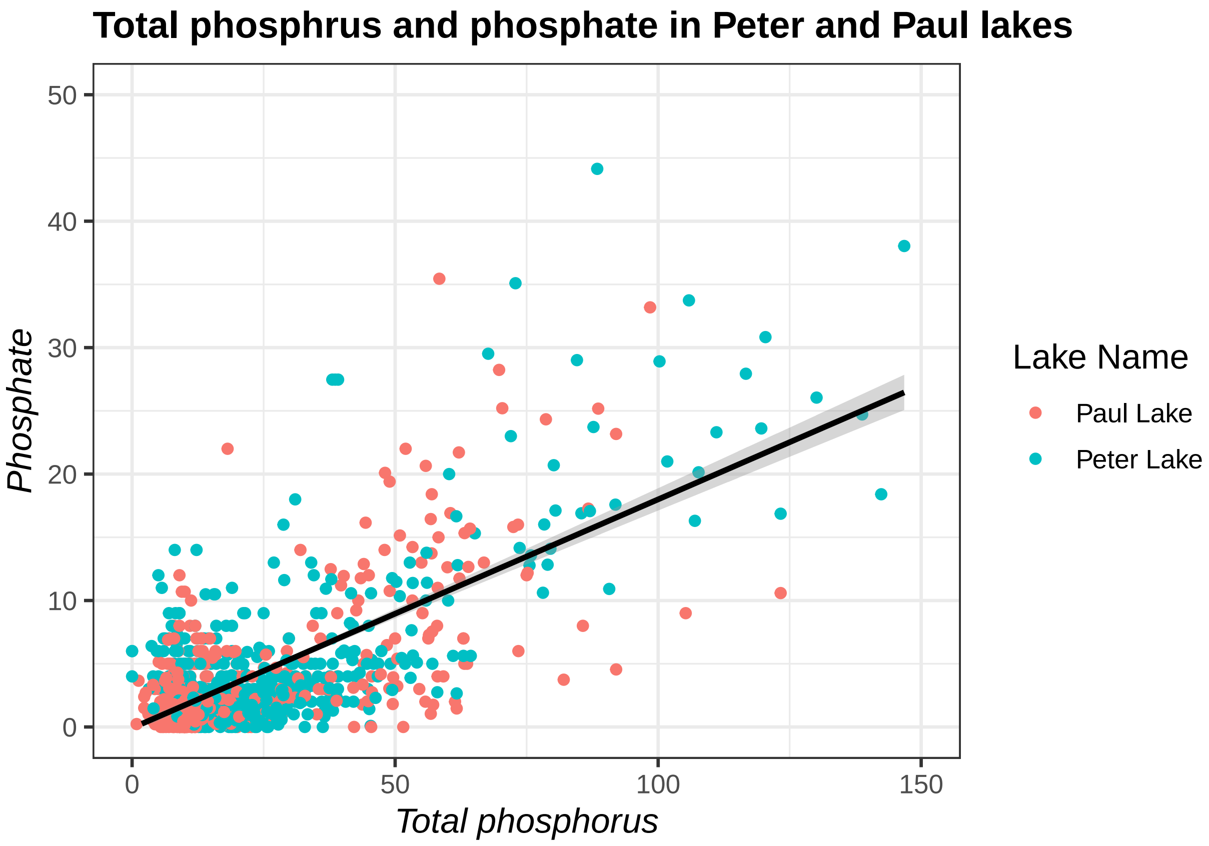
```
#4
ug_phosphate_plot <- ggplot(data = PeterPaul.chem.nutrients,
                            aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() + geom_smooth(method = lm, color = 'black') +
  xlim(0,150) + ylim(0, 50) +
  labs(x = 'Total phosphorus', y = 'Phosphate',
       title = 'Total phosphrus and phosphate in Peter and Paul lakes', color = 'Lake Name') +
  Yuechen_theme
ug_phosphate_plot
```

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

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



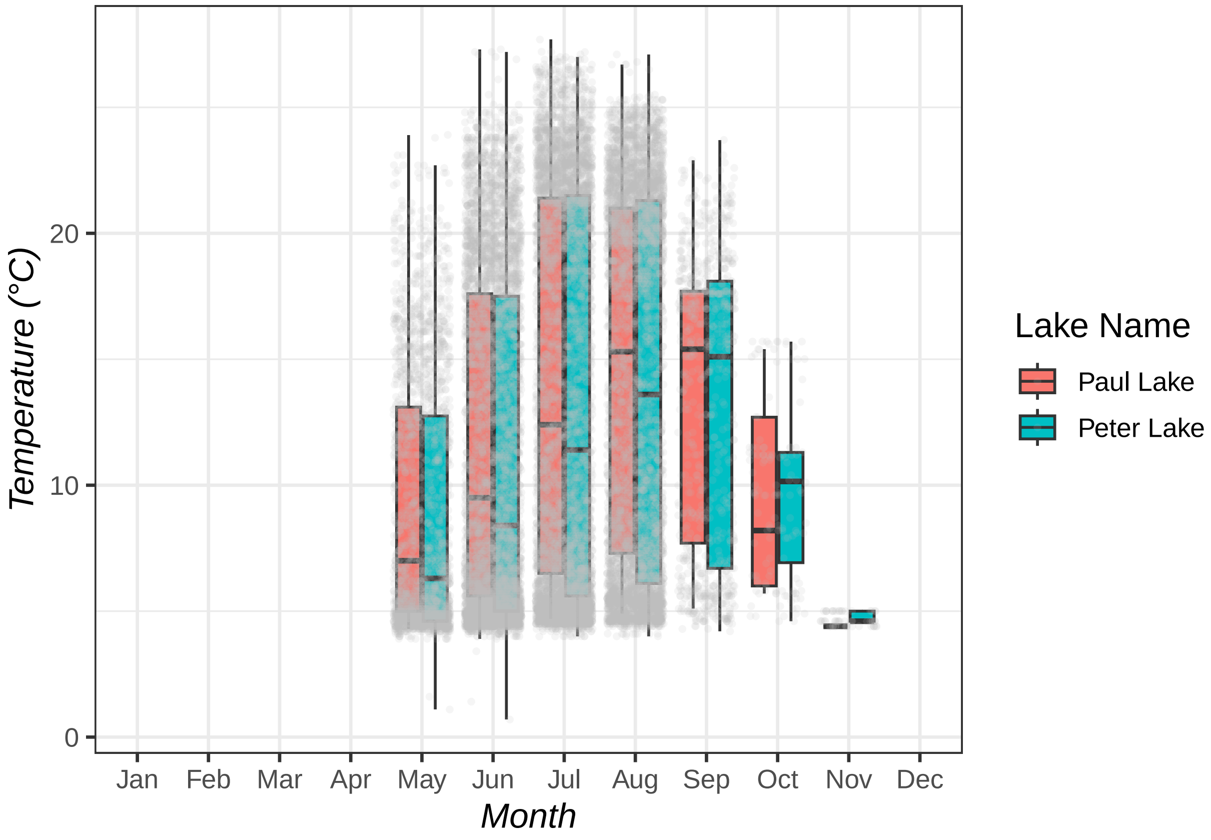
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
tem_box_plot <- ggplot(data = PeterPaul.chem.nutrients,
                       aes(x = factor(month,levels = 1:12, labels = month.abb),
                           y = temperature_C, fill = lakename)) +
  geom_boxplot() + scale_x_discrete(name = 'Month',drop = FALSE) +
  labs(y = 'Temperature (°C)', fill = 'Lake Name') +
  geom_jitter(color="grey", size=0.7, alpha=0.15) +
  Yuechen_theme
tem_box_plot
```

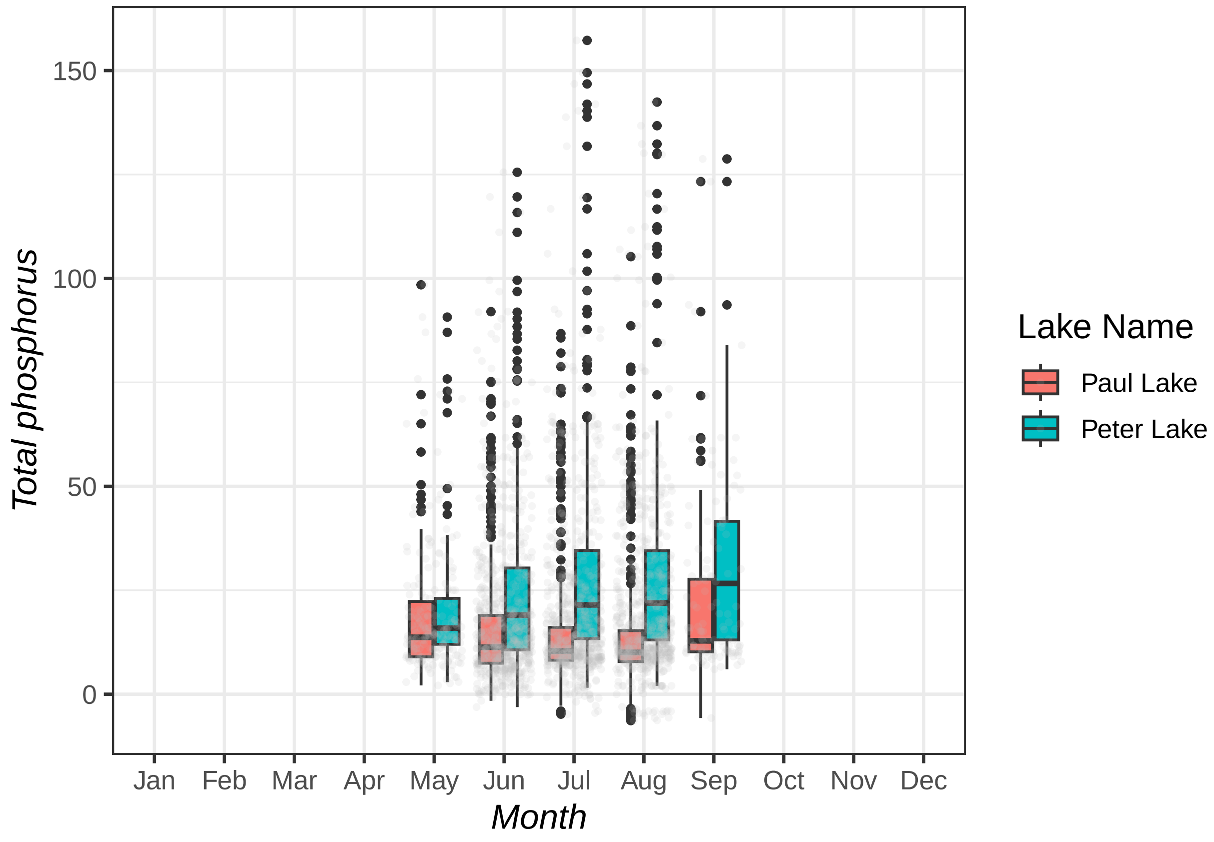
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```

```
## Warning: Removed 3566 rows containing missing values (‘geom_point()‘).
```



```
TP_box_plot <- ggplot(data = PeterPaul.chem.nutrients,
                      aes(x = factor(month,levels = 1:12, labels = month.abb),
                          y = tp_ug, fill = lakename)) +
  geom_boxplot(outlier.size = 1) + scale_x_discrete(name = 'Month',drop = FALSE) +
  labs(y = 'Total phosphorus', fill = 'Lake Name') +
  geom_jitter(color="grey", size=0.7, alpha=0.15) +
  Yuechen_theme
TP_box_plot
```

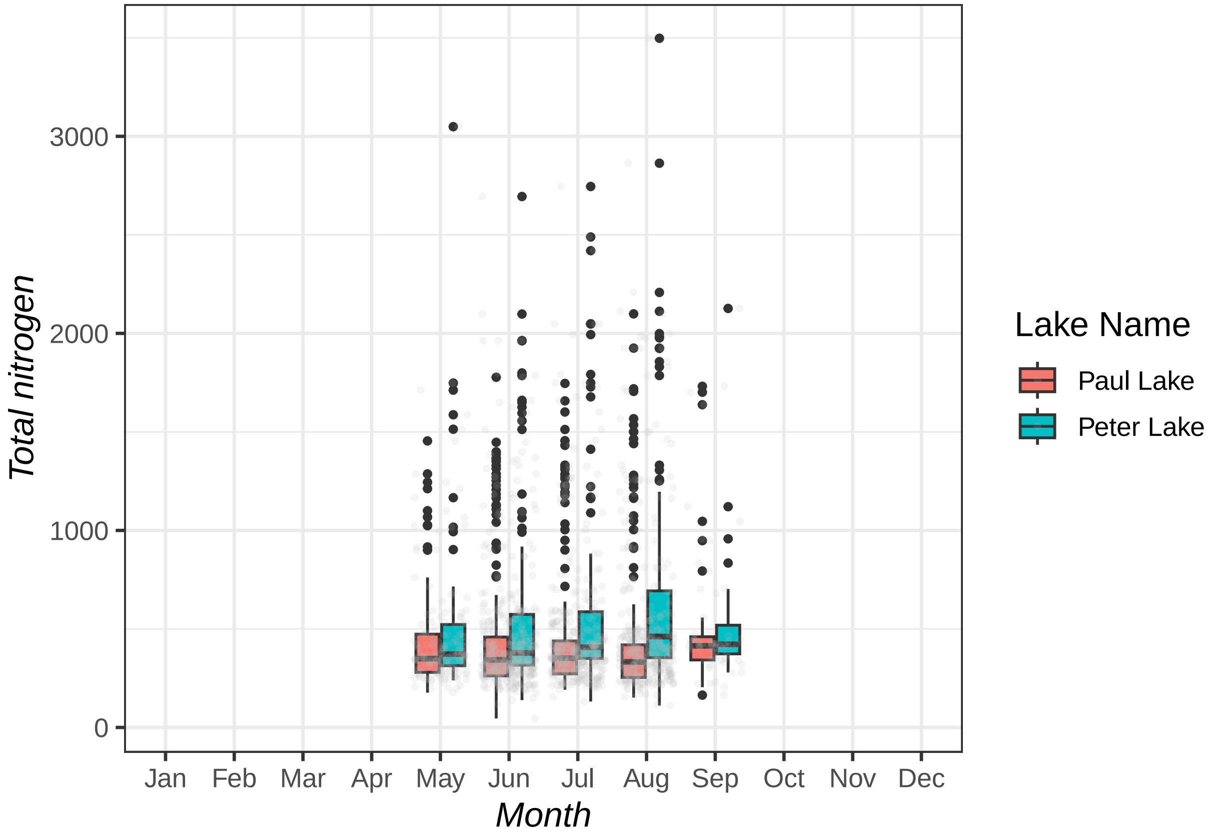
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
## Warning: Removed 20729 rows containing missing values (`geom_point()`).
```



```
TN_box_plot <- ggplot(data = PeterPaul.chem.nutrients,
                       aes(x = factor(month,levels = 1:12, labels = month.abb),
                           y = tn_ug, fill = lakename)) +
  geom_boxplot(outlier.size = 1) + scale_x_discrete(name = 'Month',drop = FALSE) +
  labs(y = 'Total nitrogen', fill = 'Lake Name') +
  geom_jitter(color="grey", size=0.7, alpha=0.15) +
  Yuechen_theme
TN_box_plot
```

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

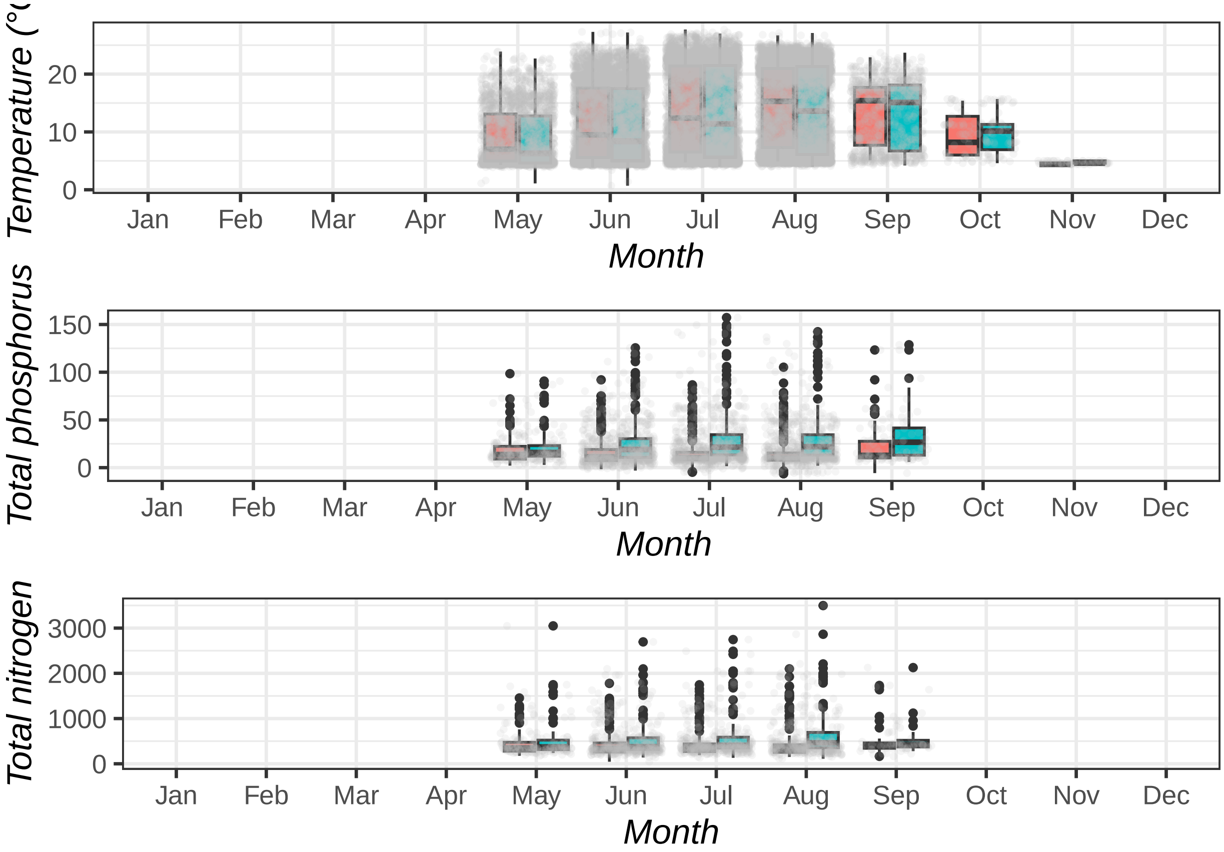
```
## Warning: Removed 21583 rows containing missing values (`geom_point()`).
```



```
gather_plot <- plot_grid(tem_box_plot + theme(legend.position="none"),
                         TP_box_plot + theme(legend.position="none"),
                         TN_box_plot + theme(legend.position="none"),
          nrow = 3, ncol = 1, align = 'h')
```

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

## Warning: Removed 3566 rows containing missing values (`geom_point()`).

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

## Warning: Removed 20729 rows containing missing values (`geom_point()`).

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

## Warning: Removed 21583 rows containing missing values (`geom_point()`).
```

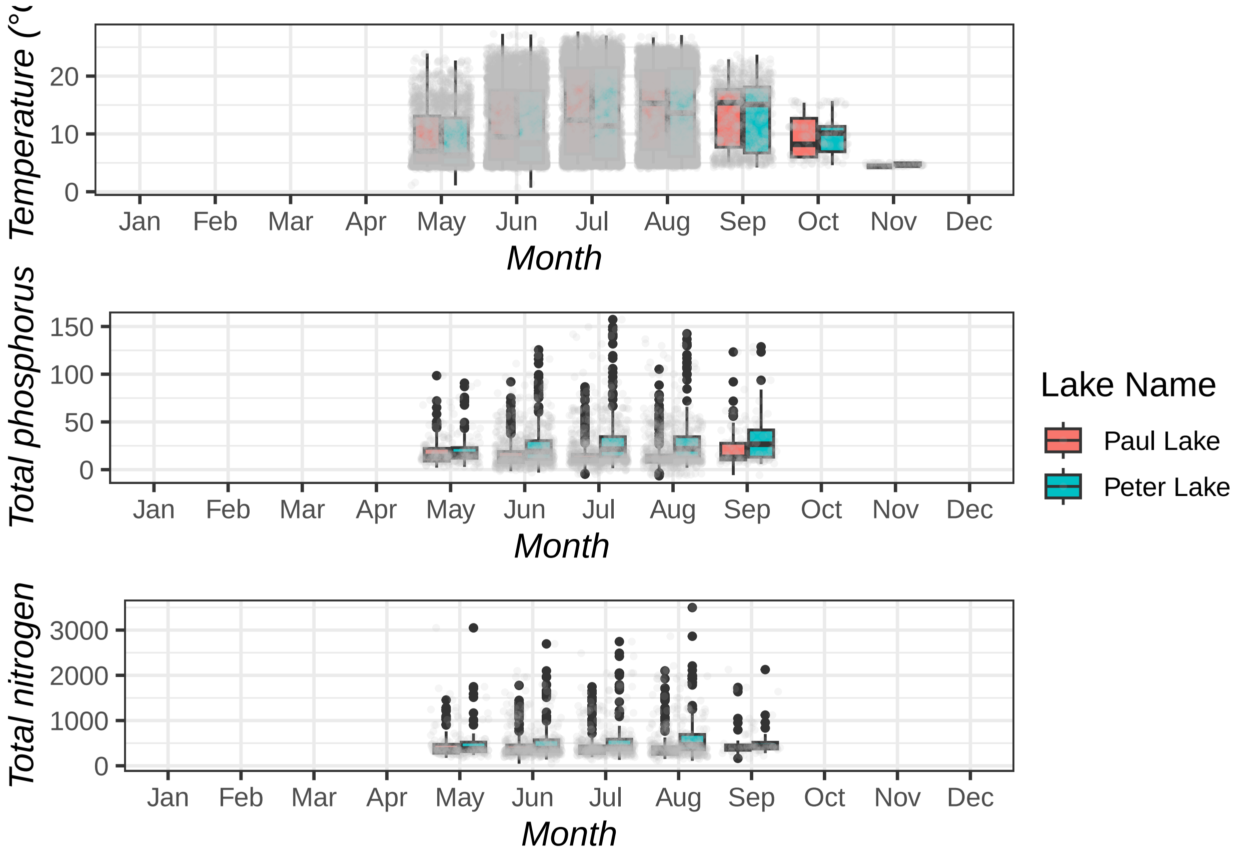
```
gather_plot
```



```
legend_com <- get_legend(tem_box_plot + theme(legend.position="right"))
```

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

## Warning: Removed 3566 rows containing missing values (‘geom_point()‘).
```

```
gather_plot_legend <- plot_grid(gather_plot, legend_com,
                                nrow = 1, rel_widths = c(5,1))
gather_plot_legend
```

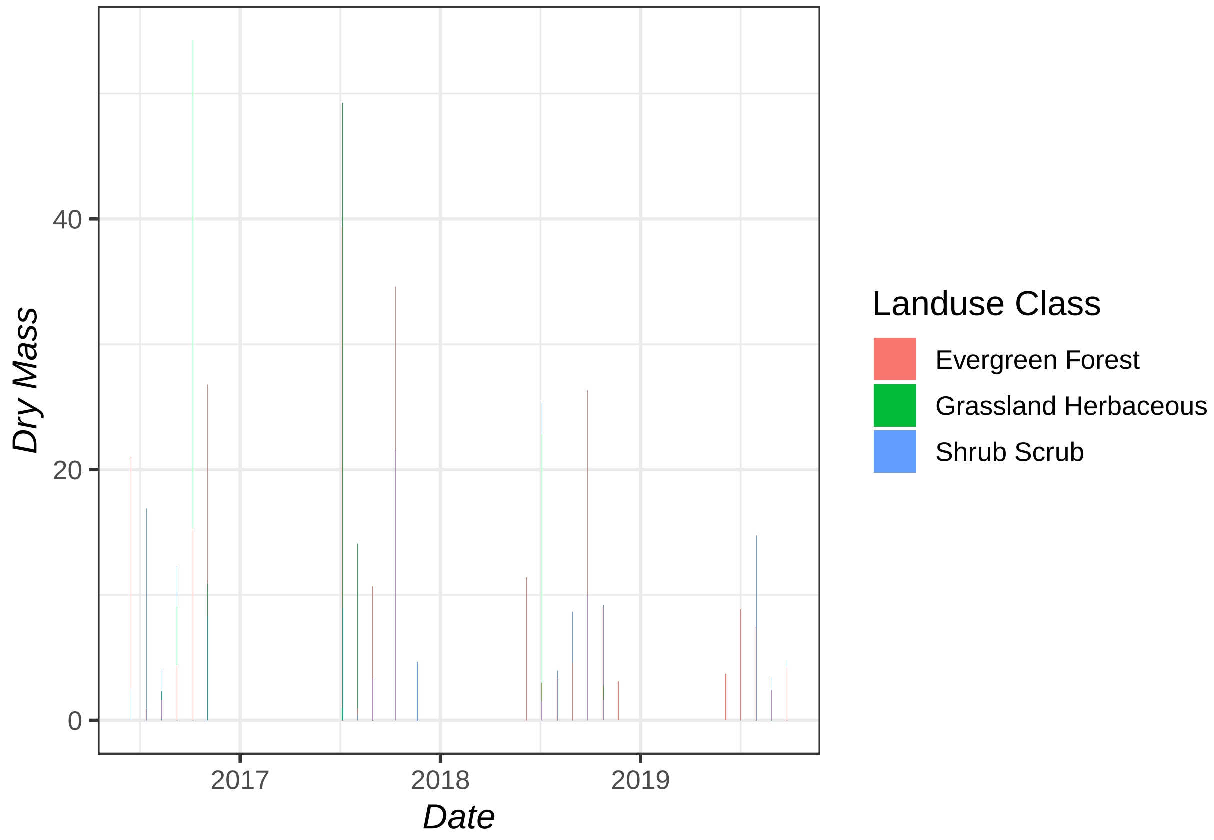


Question: What do you observe about the variables of interest over seasons and between lakes?

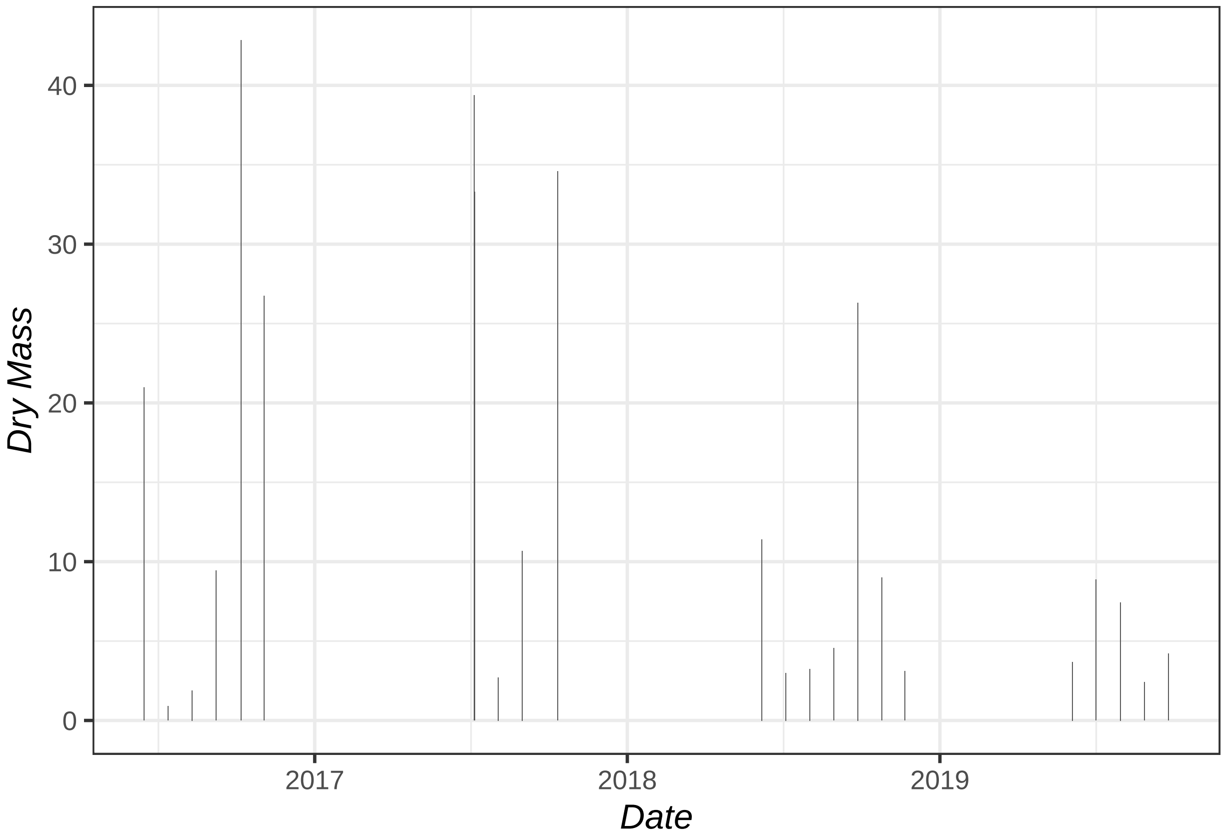
> Answer: Temperature shows up and down over seasons. The highest temperature is in July, August or Sepember. However, the total phosphrus and nitrogen levels are relative constant over seasons. Peter Lake has higher TP and TN values compared to Paul Lake.

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.

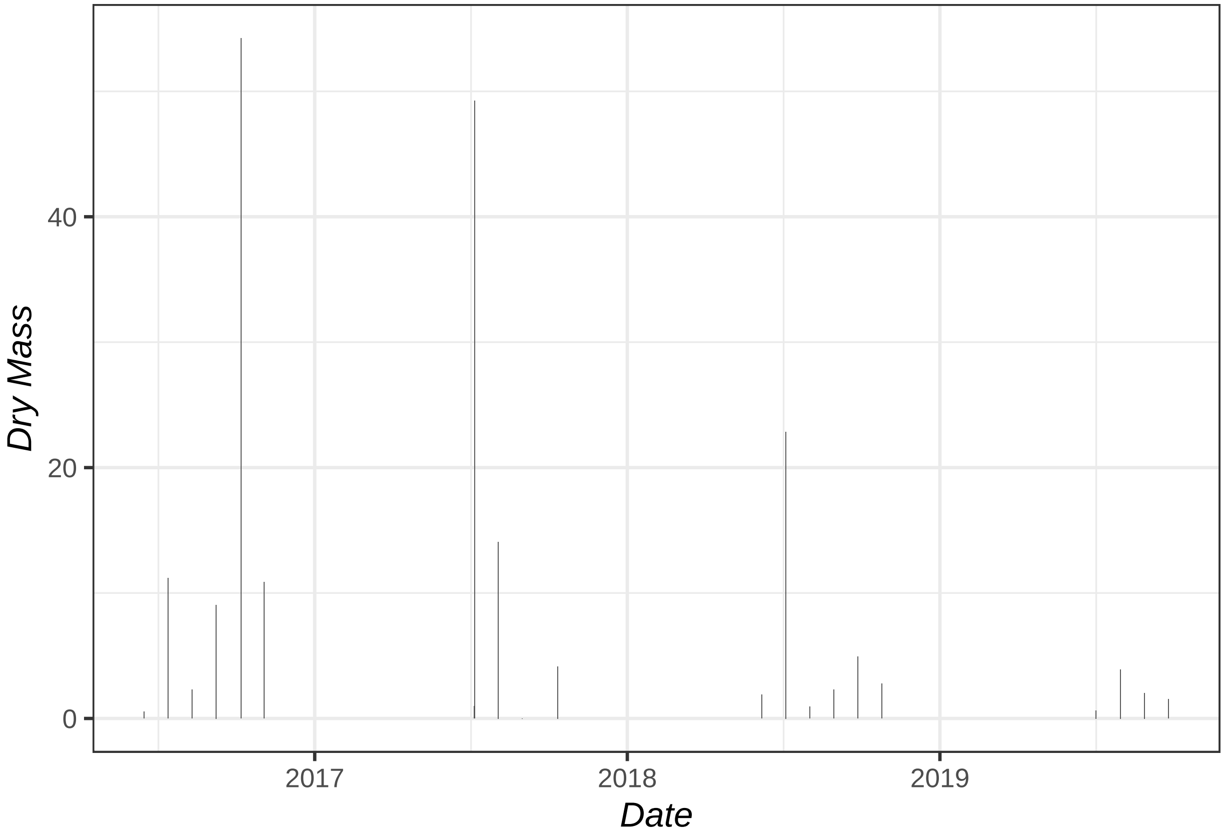
```
#6
Needle_litter <- filter(NEON.litter, functionalGroup == 'Needles')
dryMass_plot_color <- ggplot(Needle_litter, aes(x = collectDate, y = dryMass, fill = nlcdClass)) + geom_
dryMass_plot_color
```



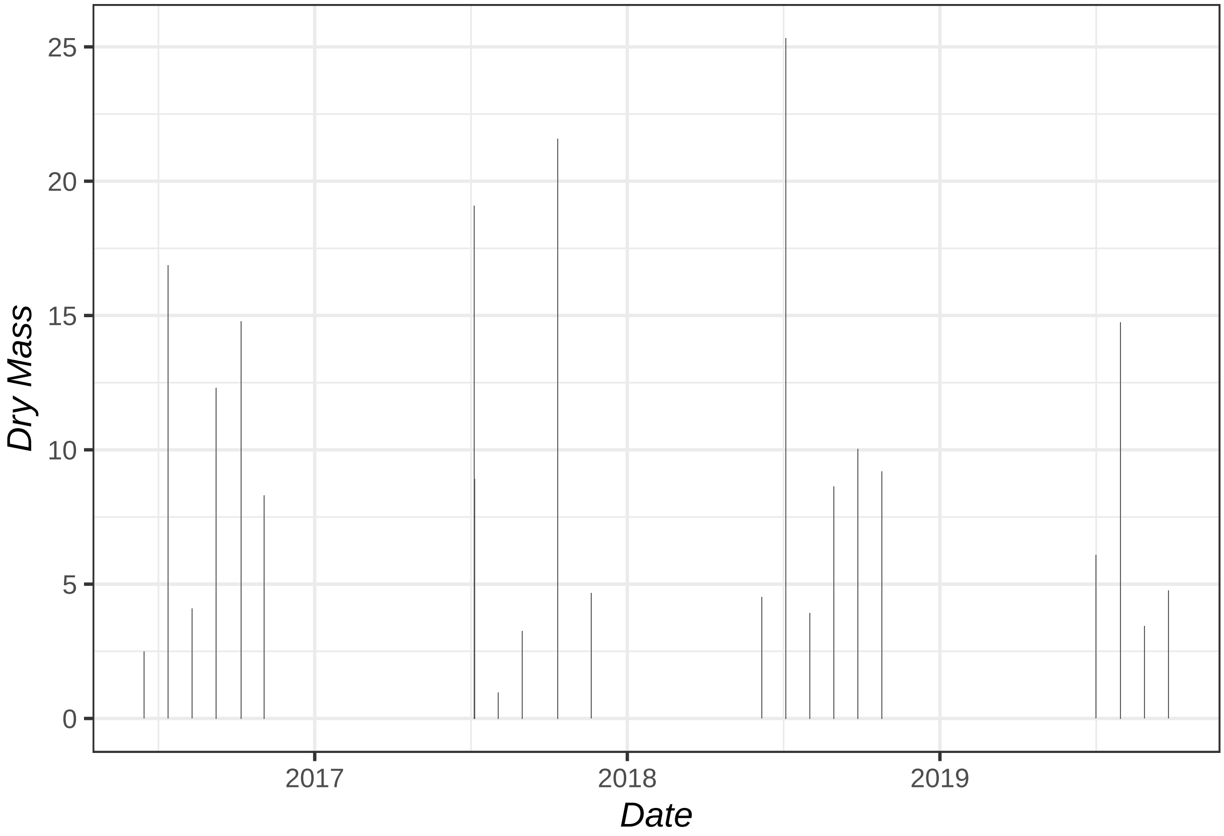
```
#7
dryMass_plot_evergreen <- ggplot(
  data = filter(Needle_litter, nlcdClass == 'evergreenForest'), aes(x = collectDate, y = dryMass)) +
  geom_bar(stat="identity", position = 'dodge') +
  labs(x = 'Date', y = 'Dry Mass') + Yuechen_theme
dryMass_plot_evergreen
```



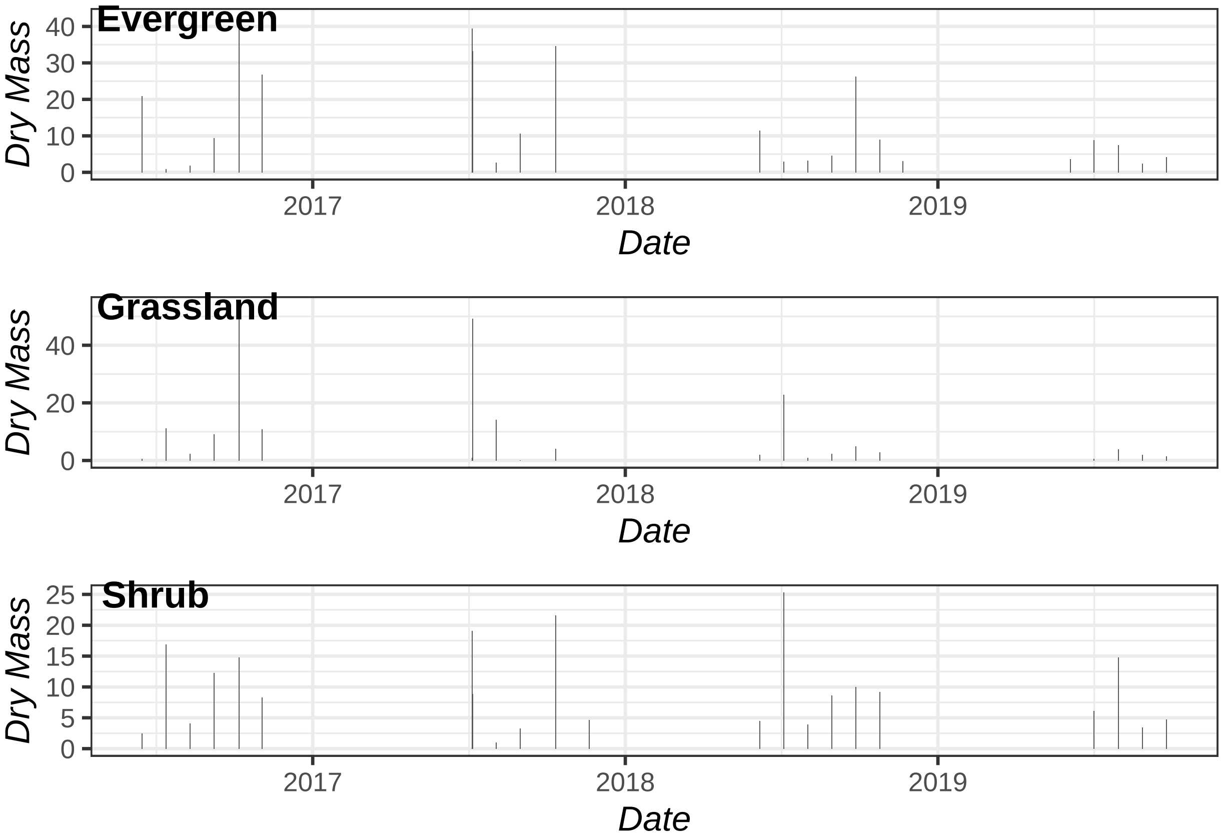
```r
dryMass_plot_grassland <- ggplot(data = filter(Needle_litter, nlcdClass == 'grasslandHerbaceous'), aes(
  geom_bar(stat="identity", position = 'dodge') +
  labs(x = 'Date', y = 'Dry Mass') +
  Yuechen_theme
dryMass_plot_grassland
```



```
dryMass_plot_shrub <- ggplot(data = filter(Needle_litter,
                                           nlcdClass == 'shrubScrub'),
                             aes(x = collectDate, y = dryMass)) +
  geom_bar(stat="identity", position = 'dodge') +
  labs(x = 'Date', y = 'Dry Mass') +
  Yuechen_theme
dryMass_plot_shrub
```



```
gather_dryMass <- plot_grid(dryMass_plot_evergreen, dryMass_plot_grassland,
          dryMass_plot_shrub, nrow = 3,
          labels = c('Evergreen','Grassland','   Shrub'),
          hjust = -0.6, align = 'left')
gather_dryMass
```



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

Answer: I prefer the second plot (7) because has less data on each plot and can still convey the information effectively.