Assignment 5: Data Visualization

Jenn McNeill

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
library(tidyverse)
```

```
## -- 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
```

```
library(lubridate)
library(here)
```

```
## here() starts at /Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

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

```
here()
```

```
## [1] "/Users/jennifermcneill/EDE_Fall2023/EDE_Fall2023"
```

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

#2
PeterPaul.chem.nutrients$sampledate <-
  ymd(PeterPaul.chem.nutrients$sampledate)
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

```r
#3
jenn_default_theme <- theme_classic() +
  theme(
    #customize the plot background, title text, and axis title text
    panel.background = element_rect(fill = "grey97"),
    plot.title = element_text(size = 14,
                              face = "bold",
                              color = "violetred4"),
    axis.title.x = element_text(size = 10,
                                face = "bold",
                                color = "tomato3"),
    axis.title.y = element_text(size = 10,
                                face = "bold",
                                color = "tomato3"),

    # customize the axis ticks and gridlines
    axis.text = element_text(size = 8, color = "grey44"),
    axis.line = element_line(color = "grey44"),
    panel.grid.major = element_line(color = "grey44",
                                    linetype = "dotted"),

    # customize the legend
    legend.position = "right",
    legend.title = element_text(size = 12,
                                face = "bold",
                                color = "darkolivegreen"),
    legend.text = element_text(size = 10,
                               color = "darkolivegreen"),
    line = element_line(0.5))

#set as default theme
theme_set(jenn_default_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 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()`).

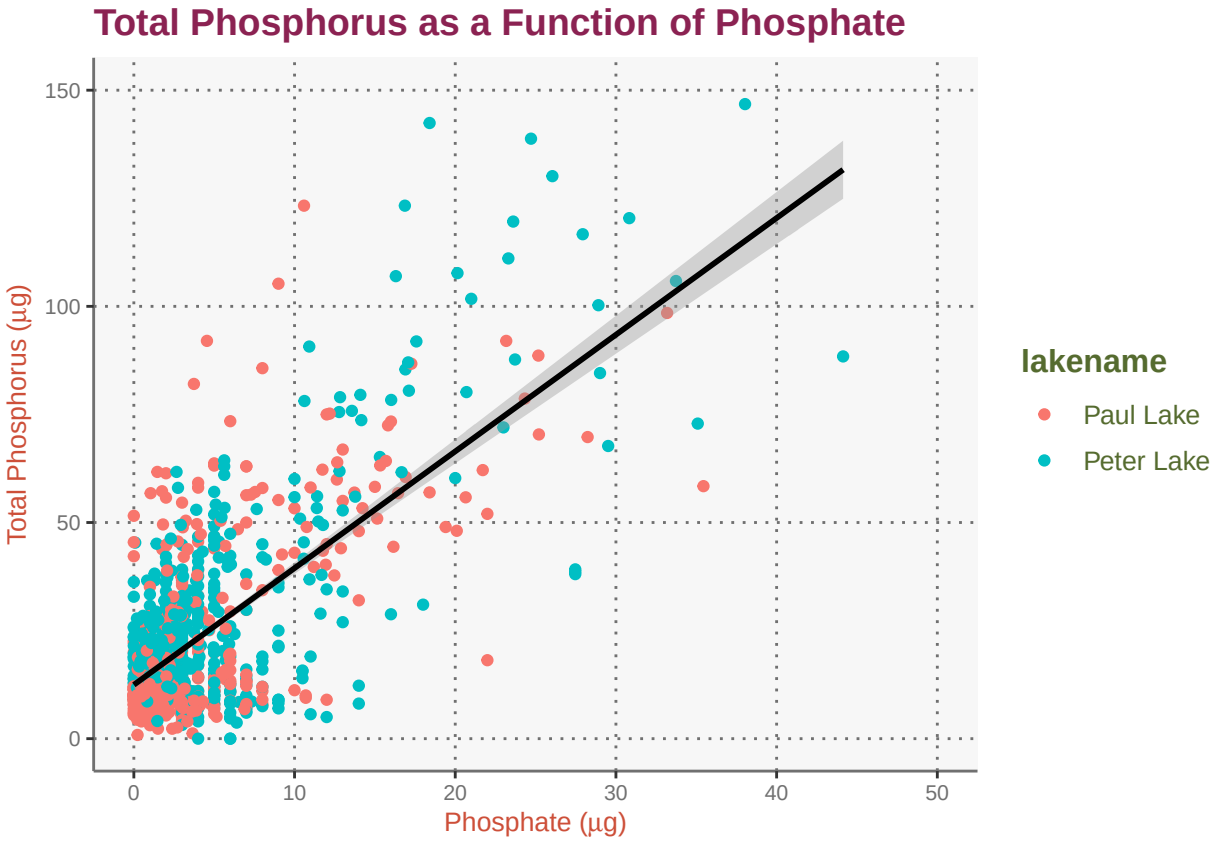
```r
#4
total_phosphorus_plot <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = po4,
             y = tp_ug,
             color = lakename))+
  geom_point()+
  geom_smooth(method="lm", color="black")+
  xlim(0, 50)+
  ylim(0, 150)+
```

```
  xlab(expression(paste("Phosphate (", mu, "g)")))+
  ylab(expression(paste("Total Phosphorus (", mu, "g)")))+
  ggtitle("Total Phosphorus as a Function of Phosphate")

print(total_phosphorus_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()`).
```



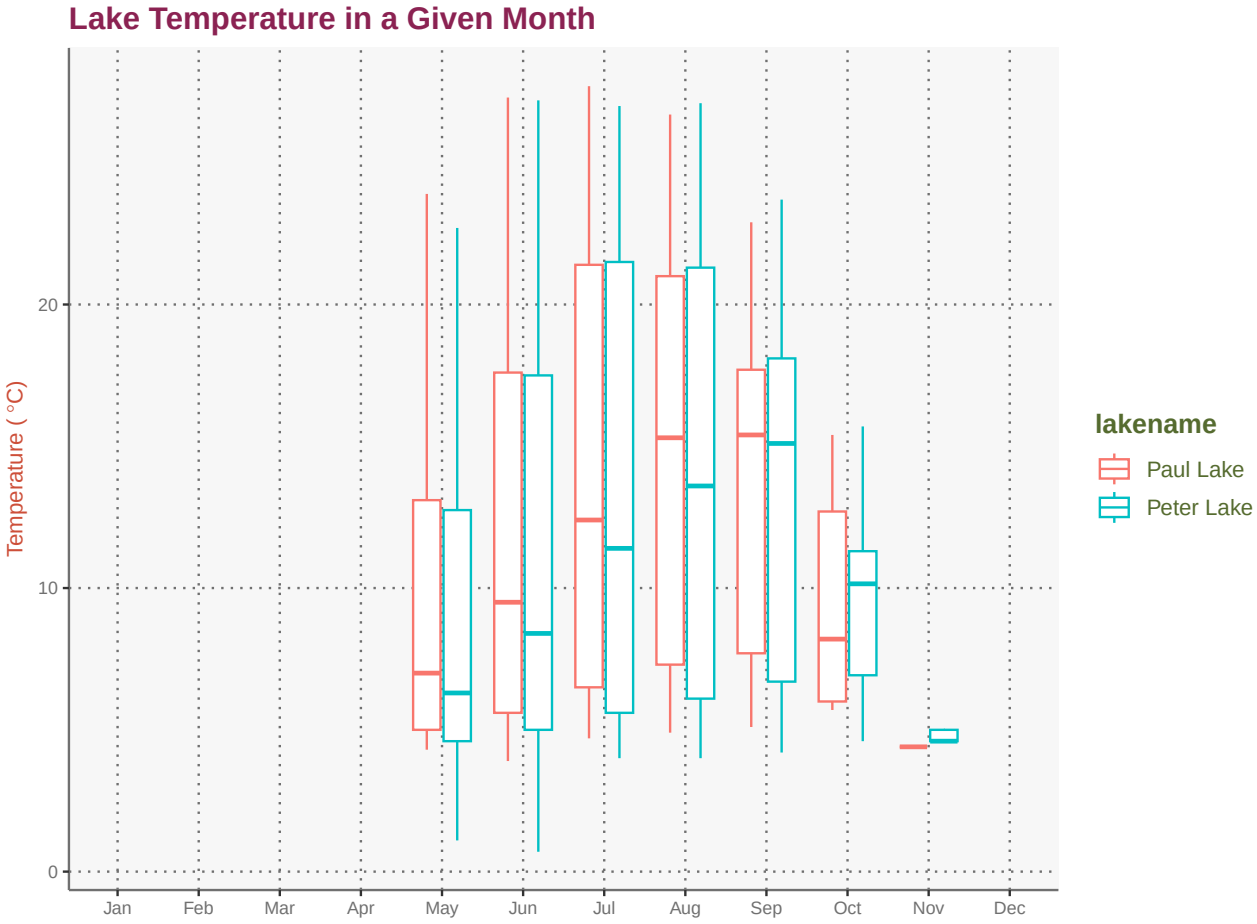
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
temperature_plot <-
  ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=factor(month,
                            levels=1:12,
```

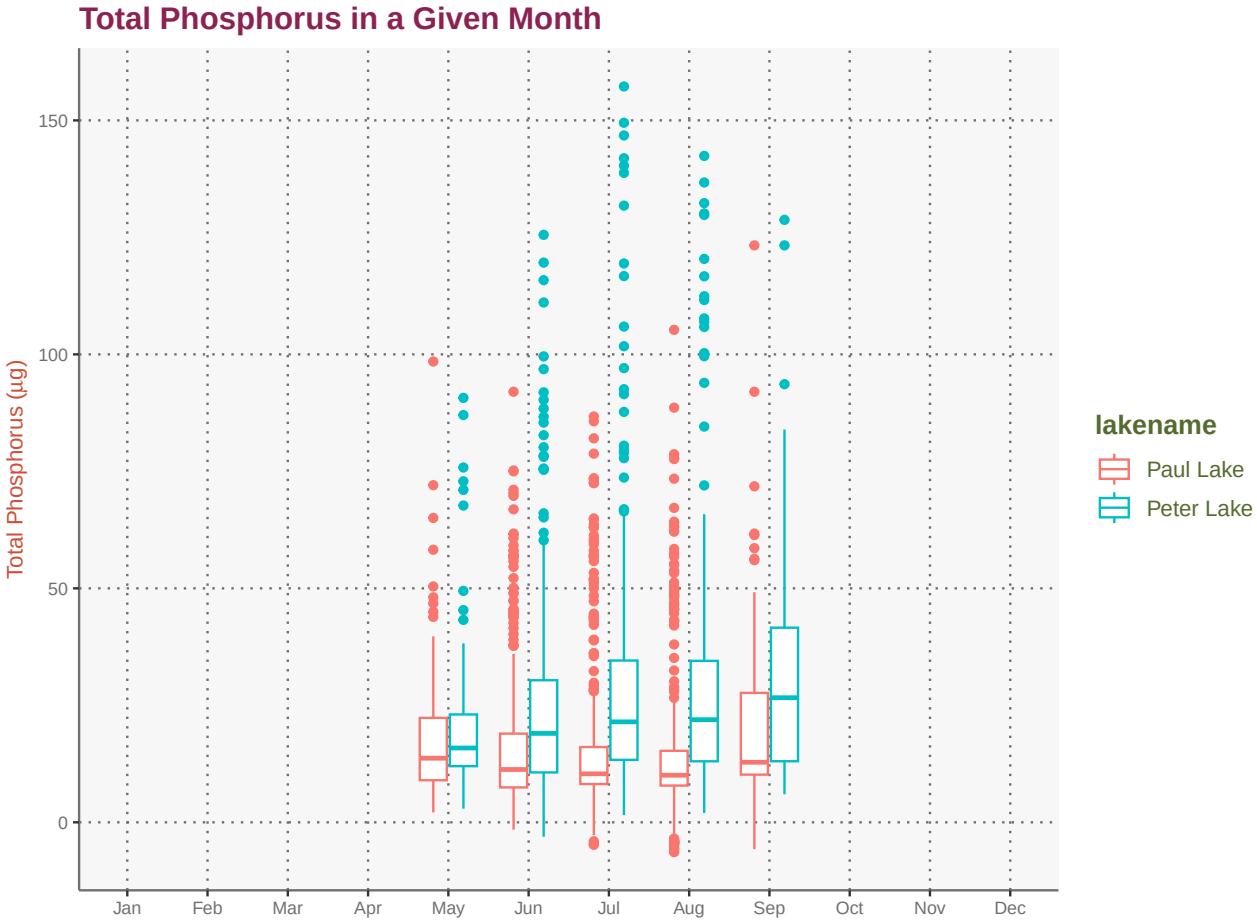
```
                           labels=month.abb),
                   y=temperature_C,
                   color = lakename))+
  xlab("")+
  scale_x_discrete(drop = FALSE)+
  ylab(expression(paste("Temperature (", ~degree, "C)")))+
  ggtitle("Lake Temperature in a Given Month")
print(temperature_plot)
```

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



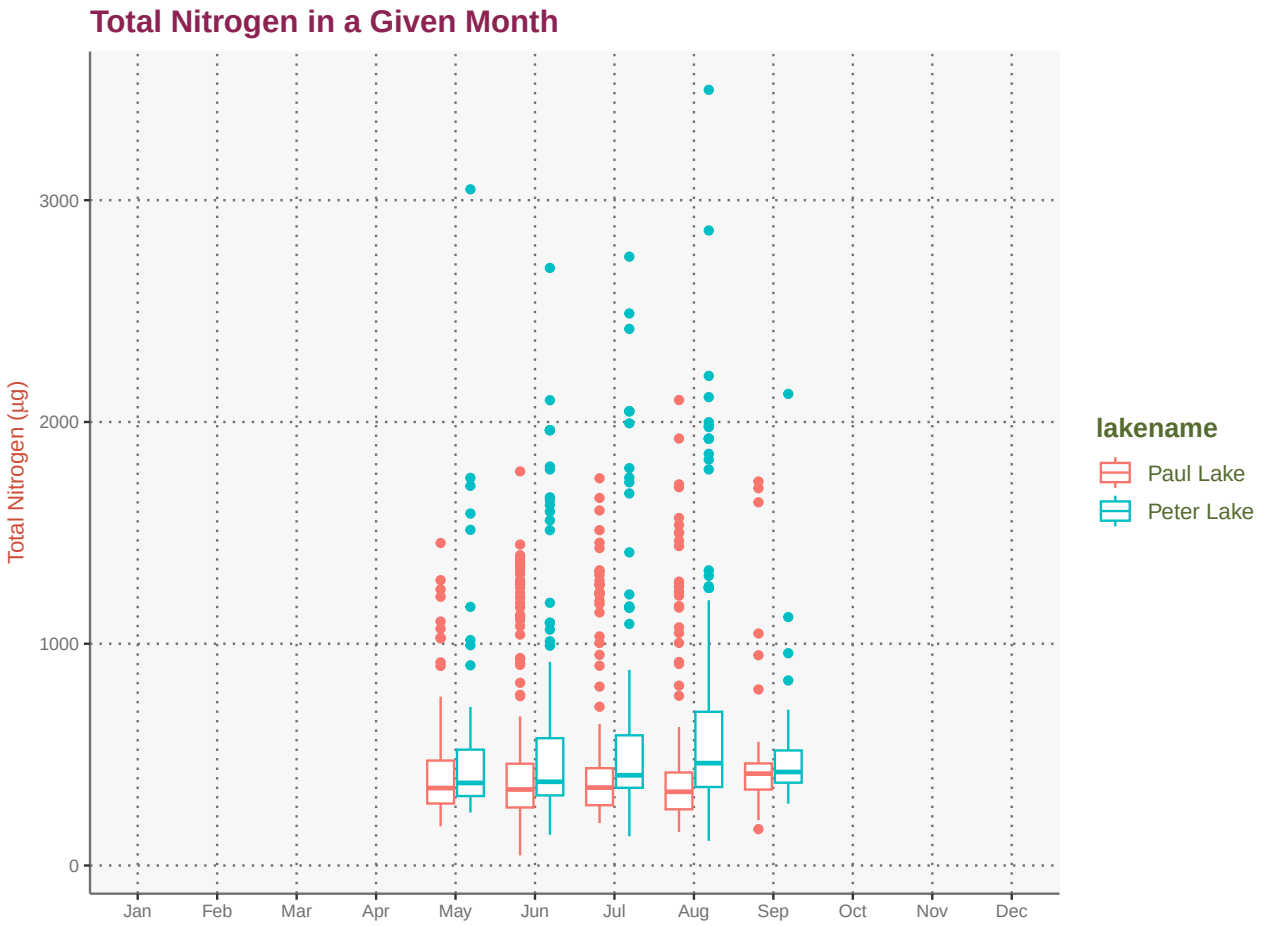

```
TP_plot <-
  ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=factor(month,
                           levels=1:12,
                           labels=month.abb),
                   y = tp_ug,
                   color = lakename))+
  xlab("")+
  scale_x_discrete(drop = FALSE)+
  ylab(expression(paste("Total Phosphorus (", mu, "g)")))+
  ggtitle("Total Phosphorus in a Given Month")
print(TP_plot)
```

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




```
TN_plot <-
  ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=factor(month,
                            levels=1:12,
                            labels=month.abb),
                   y = tn_ug,
                   color = lakename))+
  xlab("")+
  scale_x_discrete(drop = FALSE)+
  ylab(expression(paste("Total Nitrogen (", mu, "g)")))+
  ggtitle("Total Nitrogen in a Given Month")
print(TN_plot)
```

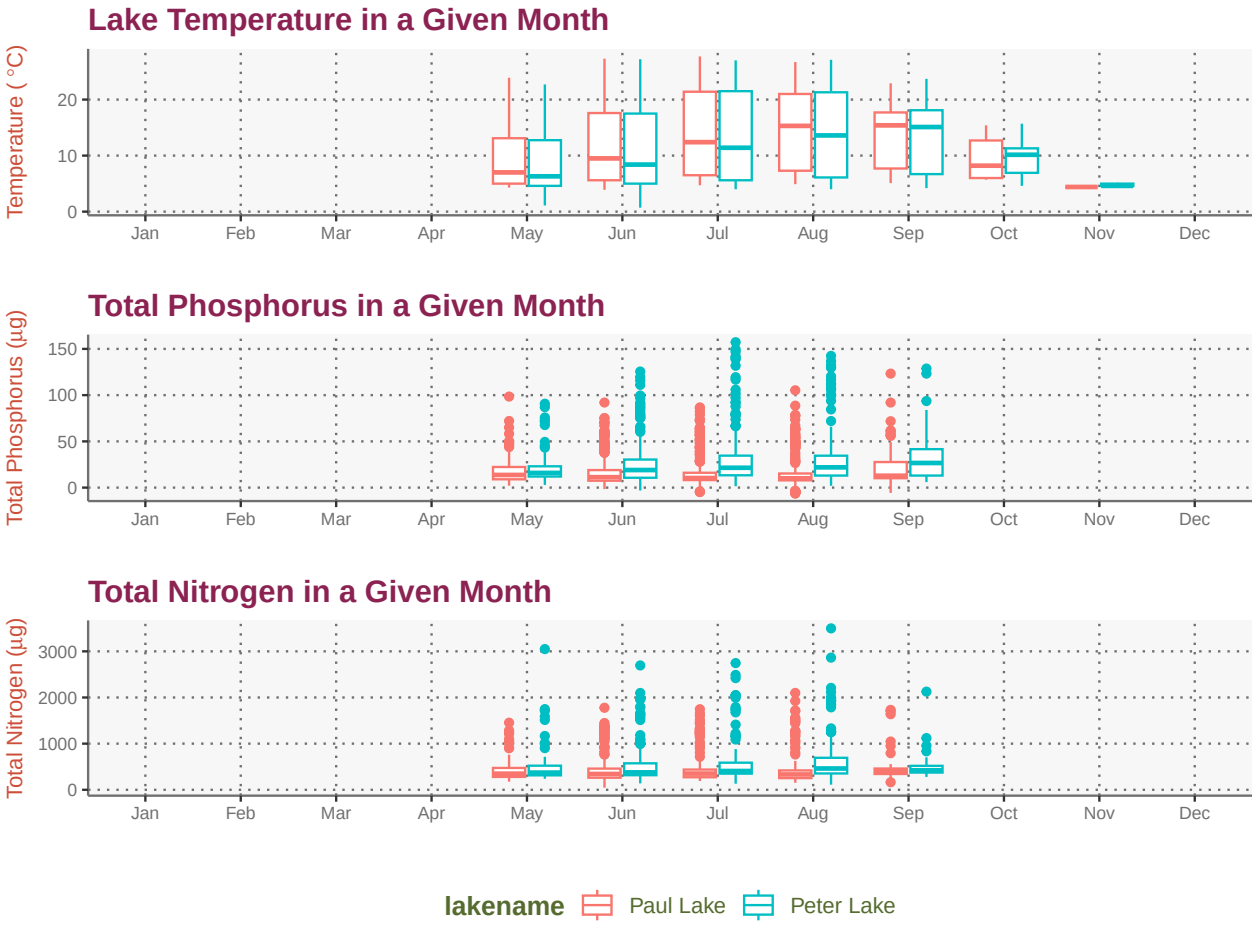
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```




```
combined_plot <-
  plot_grid(temperature_plot + theme(legend.position = "none"),
            TP_plot + theme(legend.position = "none"),
            TN_plot + theme(legend.position = "bottom"),
            ncol = 1,
            align = "vh",
            rel_heights = c(1.5, 1.5, 2))+
  scale_x_discrete(drop = FALSE)
```

```
## 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 horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.

## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
print(combined_plot)
```



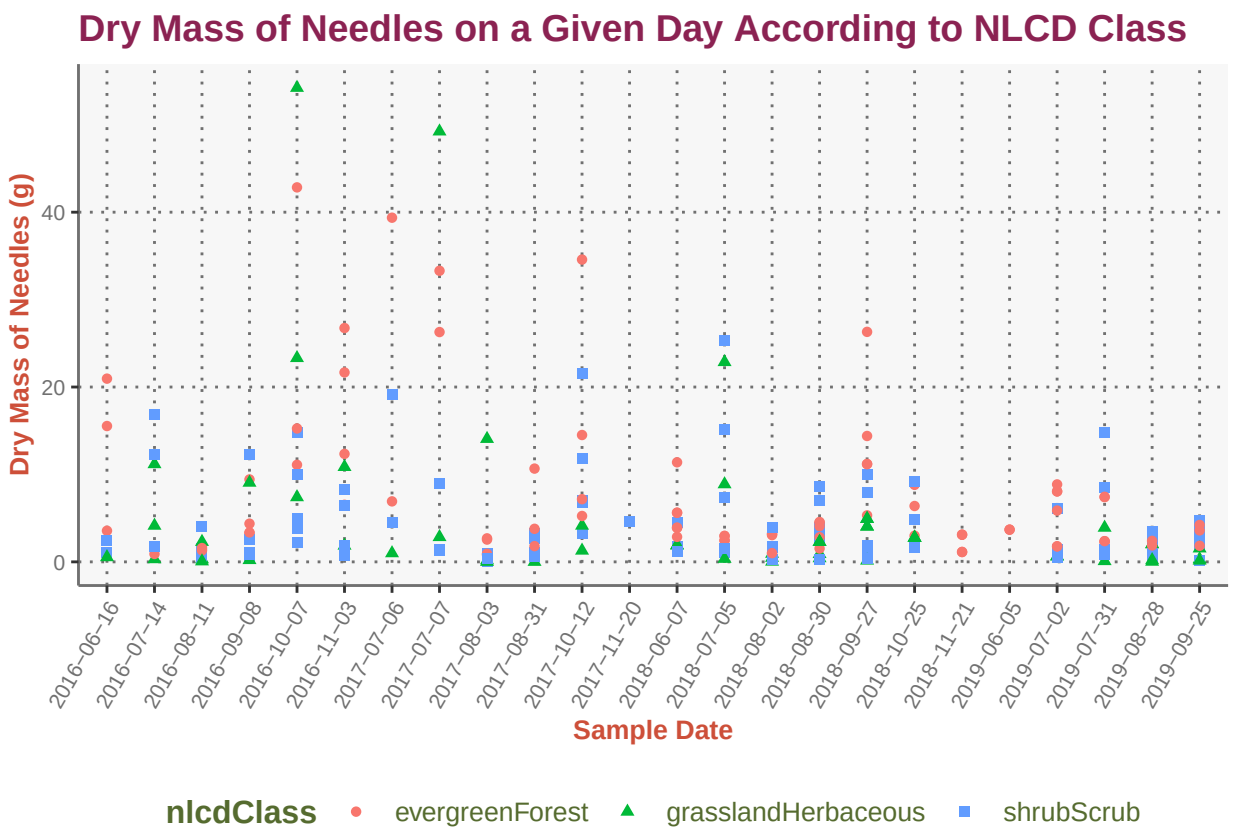
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: From May until September, the average temperature in Paul Lake is slightly higher than the average temperature in Peter Lake. Once fall begins in October, the temperature in Peter Lake is higher than Paul Lake. These averages likely demonstrate that Paul Lake is more shallow and thus more suseptible to temperature change as the weather outside changes. Peter Lake, likely being the deeper of the two, can hold its heat more effectively as the summer ends and the weather gets colder. The relative temperature drop from September to October in both lakes supports this hypothesis. The Total Phosphorus and Total Nitrogen, on average, are slightly higher in Peter Lake than in Paul Lake for all months. This fact would also support the hypothesis that Peter Lake is deeper than Paul Lake and thus has more volume in which phosphorus and nitrogen can thrive.

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)

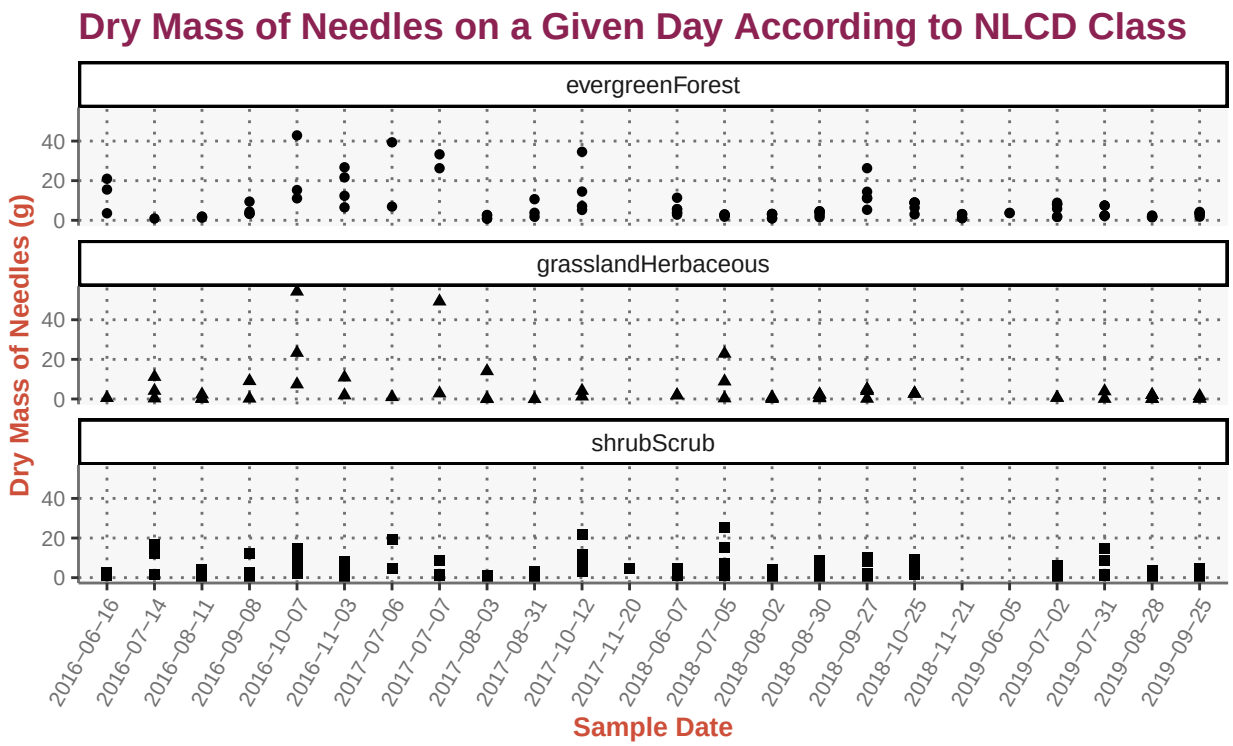
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
litter_subset_plot <-
  ggplot(filter(NiwotRidge_litter,
                functionalGroup=="Needles"),
           aes(x = factor(collectDate),
               y = dryMass,
               color = nlcdClass,
               shape = nlcdClass))+
  geom_point()+
  xlab("Sample Date")+
  ylab("Dry Mass of Needles (g)")+
  ggtitle("Dry Mass of Needles on a Given Day According to NLCD Class")+
  jenn_default_theme+
  theme(axis.text.x = element_text(angle = 60,
                                   vjust = 1,
                                   hjust = 1),
        legend.position = "bottom")
print(litter_subset_plot)
```




```
#7
litter_facet_plot <-
  ggplot(filter(NiwotRidge_litter,
                functionalGroup=="Needles"),
           aes(x = factor(collectDate),
               y = dryMass,
               shape = nlcdClass))+
  geom_point()+
```

```
  facet_wrap(vars(nlcdClass), nrow = 3)+
  xlab("Sample Date")+
  ylab("Dry Mass of Needles (g)")+
  ggtitle("Dry Mass of Needles on a Given Day According to NLCD Class")+
  jenn_default_theme+
  theme(axis.text.x = element_text(angle = 60,
                                   vjust = 1,
                                   hjust = 1),
        legend.position = "bottom")
print(litter_facet_plot)
```



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

Answer: I think that the plot showing NLCD Class separated by color aesthetic is more effective because it is easier for the viewer to compare all classes at one time. For any given sample date, the reader can quickly see which points are highest and lowest. The graph with three facets makes it more difficult to compare the data points because the reader has to jump between plots and may not remember how the numbers compare to one another. I prefer the plot with all of the data points together because it makes it easiest to compare the data and glean relationships between NLCD Classes.