Assignment 5: Data Visualization

Kendall Barton

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/kendallbarton/Downloads/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
```

```
getwd()
```

```
## [1] "/Users/kendallbarton/Downloads/EDE_Fall2023"
```

```
lake_chem_df <- read.csv("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
                         stringsAsFactors = TRUE)
litter_df <- read.csv("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
                      stringsAsFactors = TRUE)

#2
class(lake_chem_df$sampledate) #they are factors
```

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

```
class(litter_df$collectDate)
```

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

```
lake_chem_df$sampledate <- as.Date(lake_chem_df$sampledate) #make into date
litter_df$collectDate <-as.Date(litter_df$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
theme1 <- theme_base() +
  theme(
    plot.title = element_text(
      color = 'gray35', hjust = 0.5, size = 16 #hjust for centering
    ),
    plot.background =   element_rect(
      fill = 'mistyrose', color = 'gray35' #color is just border
    ),
    legend.background = element_rect(
      color='gray35', fill = 'rosybrown' #color is just border
    ),
    legend.title = element_text(
      color='white', size = 12
    ),
    legend.text = element_text(
      size=10
    ),
    axis.title = element_text( #x and y labels
      size=12
    ),
    axis.text = element_text( #numbers
      size=10
    )
  )
```

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()`).

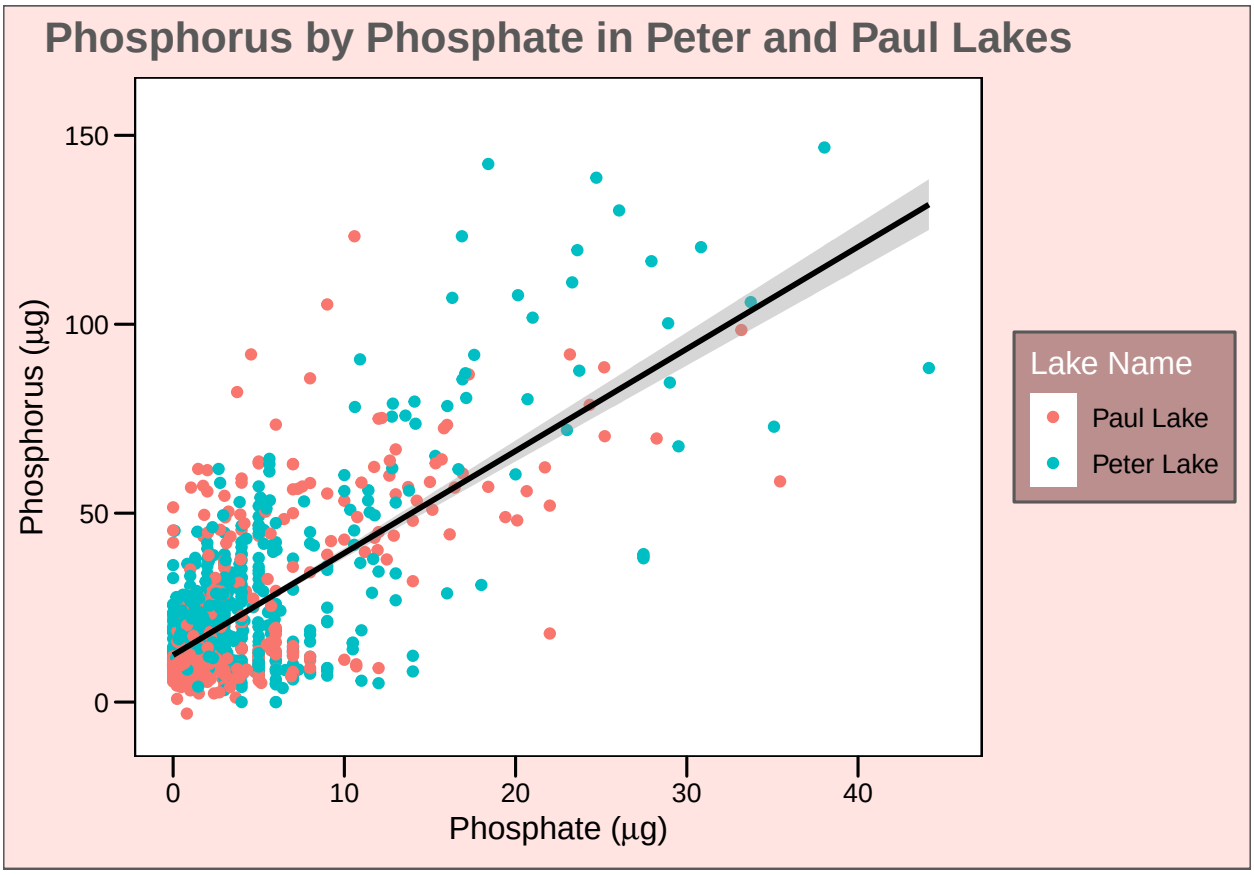
```
#4
ggplot(lake_chem_df, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  xlim(0,45) + #eliminate extreme values
  labs(title = "Phosphorus by Phosphate in Peter and Paul Lakes",
       x = expression(paste("Phosphate (", mu,"g)")), #slack said units are micrograms
       y = expression(paste("Phosphorus (", mu,"g)")), color = "Lake Name") +
```

```
  geom_smooth(method = lm, color = "black") + #line of best fit
  theme1
```

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

```
## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21947 rows containing missing values (`geom_point()`).
```



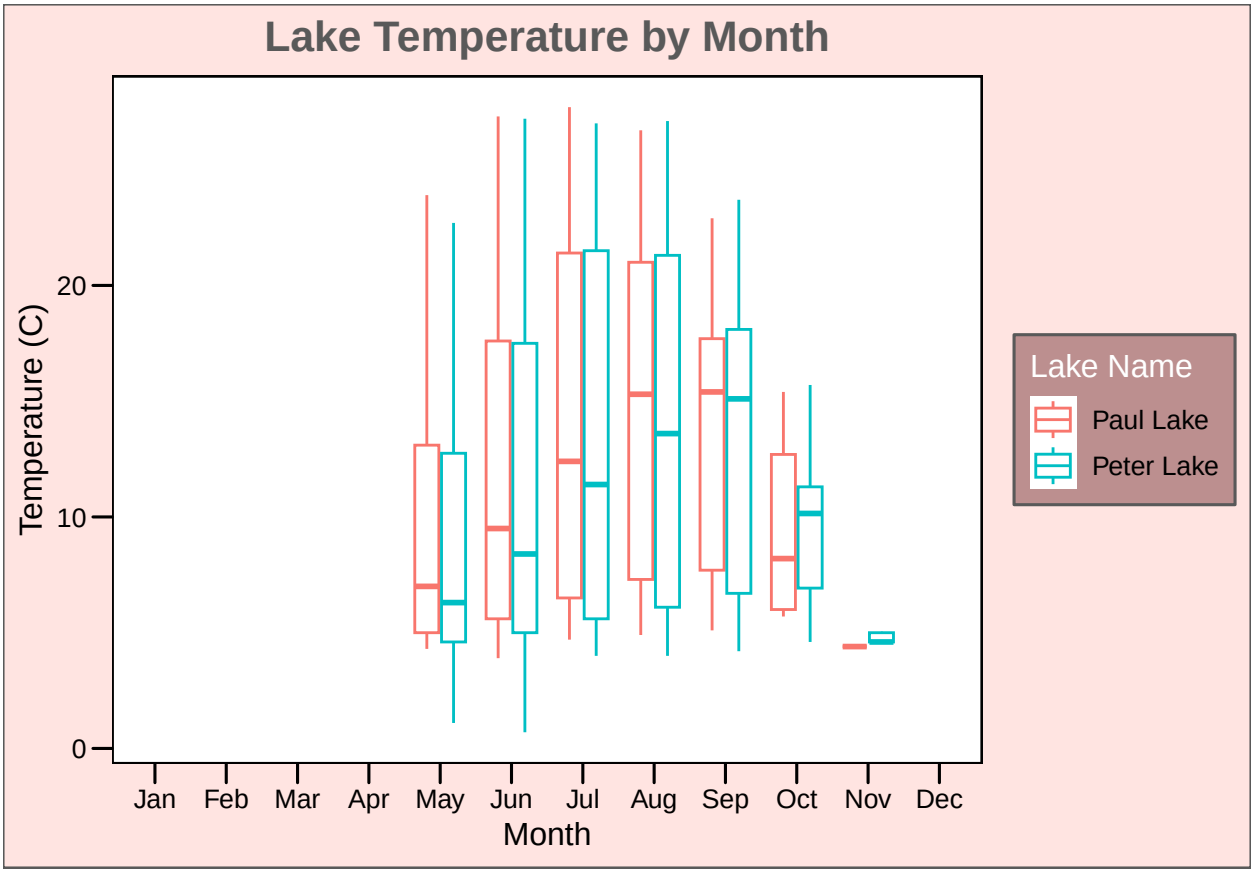
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
temp_bp <- ggplot(lake_chem_df,
                  aes(x = factor(month, levels = 1:12, labels = month.abb),
                      y = temperature_C, color = lakename)) +
  #ensure all 12 months will be present
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop = F) +
```

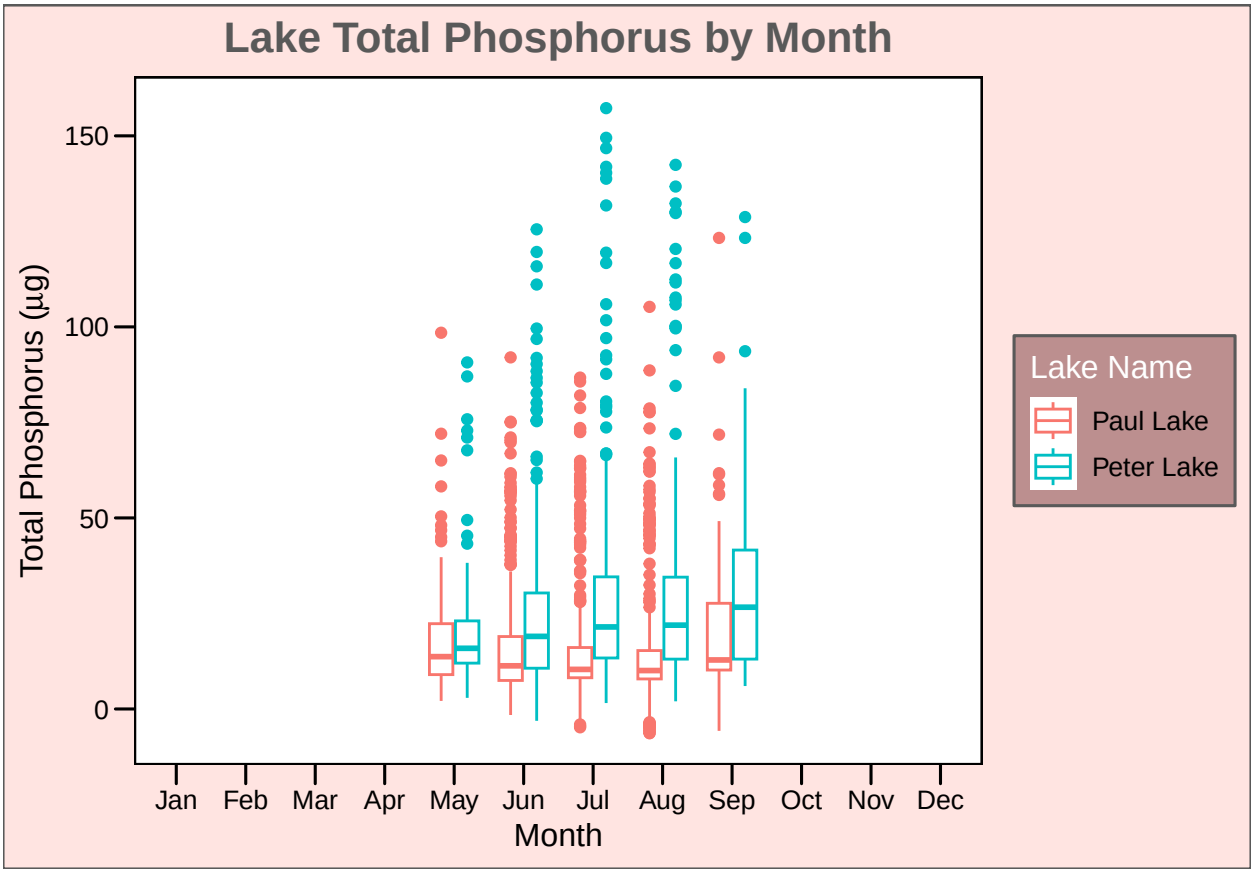
```
  labs(title = "Lake Temperature by Month", y = "Temperature (C)",
       color = "Lake Name") +
  theme1
temp_bp
```

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



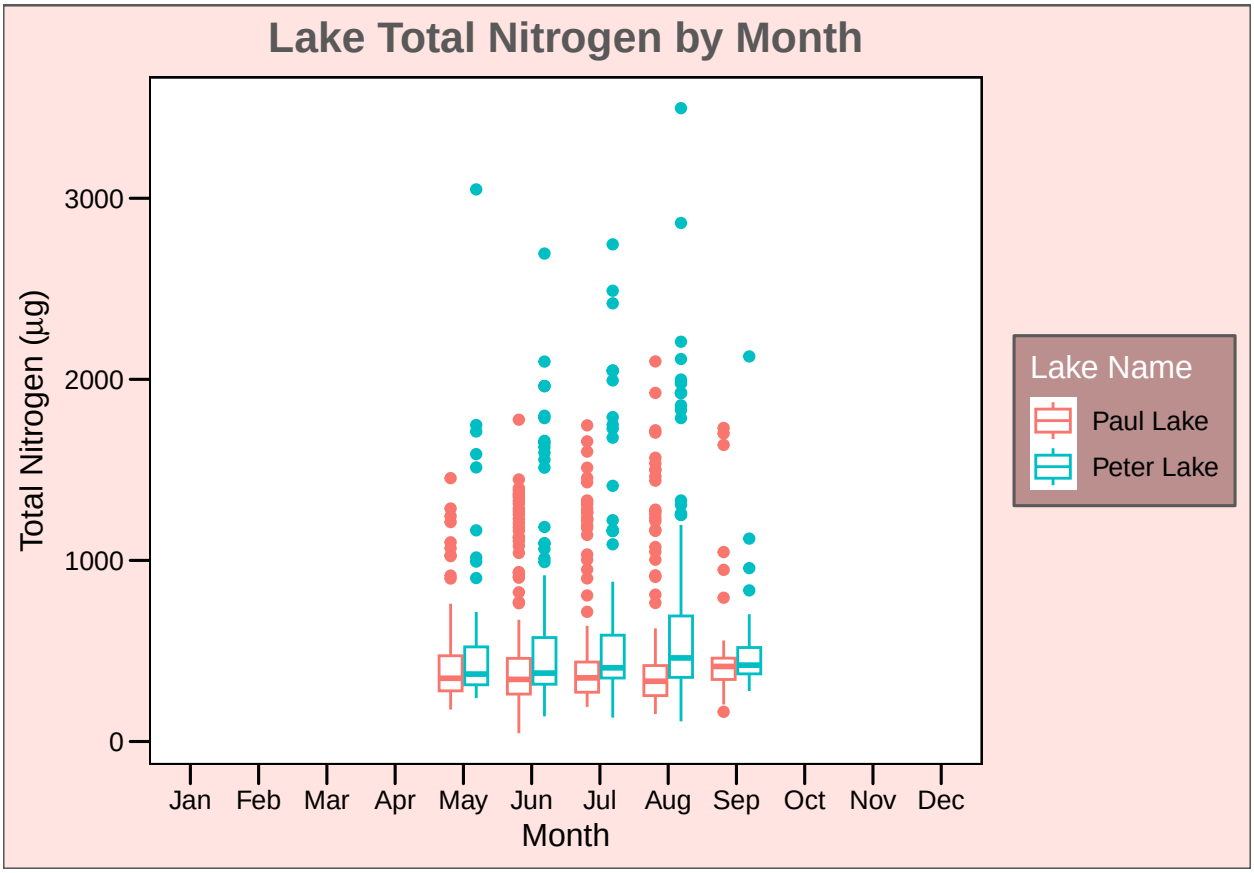

```
tp_bp <- ggplot(lake_chem_df,
                aes(x = factor(month, levels = 1:12, labels = month.abb),
                    y = tp_ug, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop = F) +
  labs(title = "Lake Total Phosphorus by Month", x = "Month",
       y = expression(paste("Total Phosphorus (", mu,"g)")),
       color = "Lake Name") +
  theme1
tp_bp
```

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



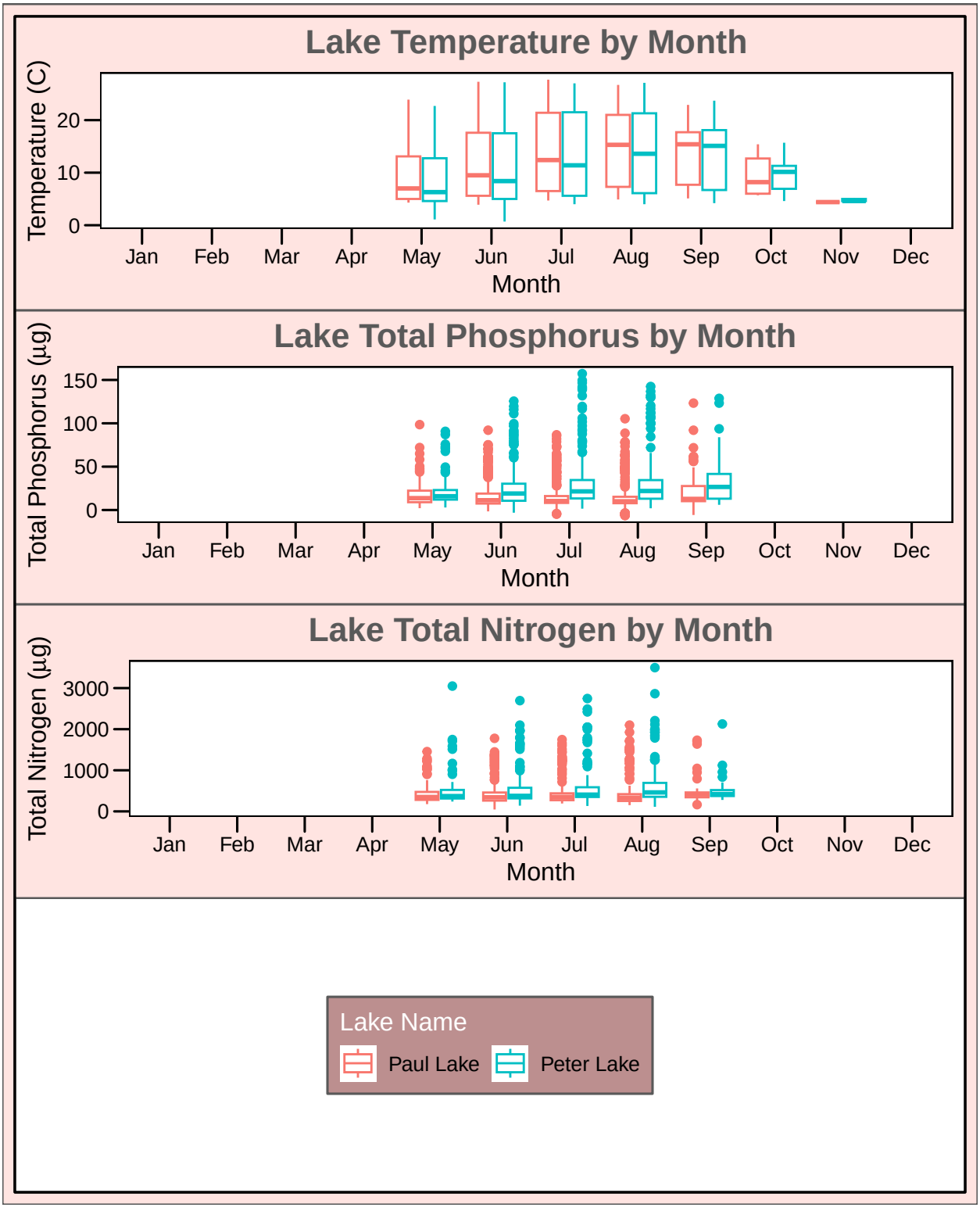

```
tn_bp <- ggplot(lake_chem_df,
                aes(x = factor(month, levels = 1:12, labels = month.abb),
                    y = tn_ug, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop = F) +
  labs(title = "Lake Total Nitrogen by Month", x = "Month",
       y = expression(paste("Total Nitrogen (", mu,"g)")),
       color = "Lake Name") +
  theme1
tn_bp
```

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




```
#I made a new chunk so I could change chunk options for this graph (fig.height=8)

plot_grid(temp_bp + theme(legend.position="none"),
          tp_bp + theme(legend.position="none"),
          tn_bp + theme(legend.position="none"), ncol = 1, align = "h", axis = "lr",
          legend = get_legend(temp_bp +
                              guides(color = guide_legend(nrow = 1)))) +
  theme1
```



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

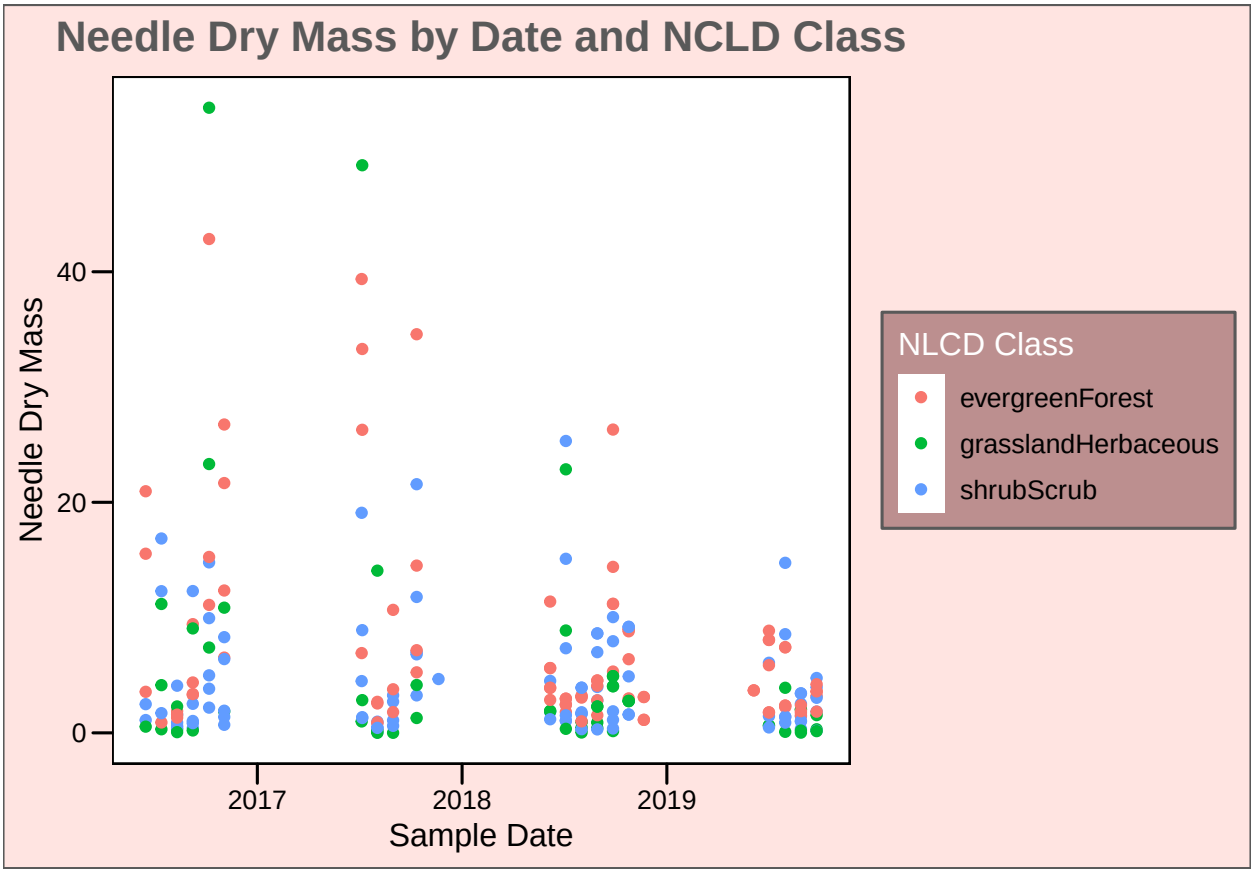
Answer: The temperature increases from May until August and September, then falls in October and November in both lakes. In Peter Lake, phosphorus increases from May to September, while in Paul Lake it changes very little. There is also higher phosphorus on average in Peter Lake

than Paul Lake. I can't see a clear trend across total nitrogen, except that Peter Lake tends to be higher than Paul Lake.

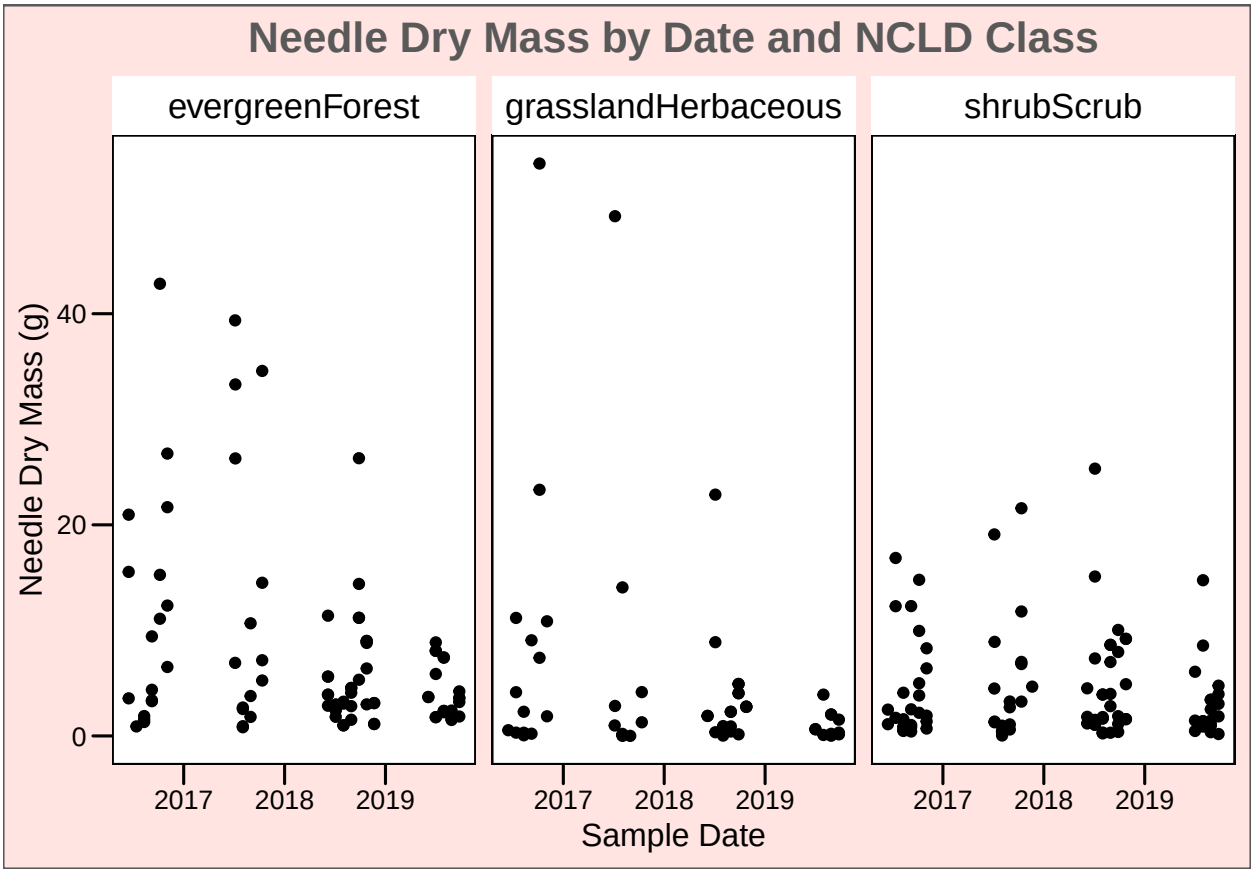
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
ggplot(filter(litter_df, functionalGroup == "Needles"), #use only Needles functional group
       aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = "Needle Dry Mass by Date and NCLD Class", x = "Sample Date",
       y = "Needle Dry Mass", color = "NLCD Class") + #color = for different colors
  theme1
```




```
#7
ggplot(filter(litter_df, functionalGroup == "Needles"),
       aes(x = collectDate, y = dryMass)) +
  geom_point() +
  labs(title = "Needle Dry Mass by Date and NCLD Class", x = "Sample Date",
       y = "Needle Dry Mass (g)") +
  facet_wrap(vars(nlcdClass)) + #facet wrap for multiple graphs
  theme1
```



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

Answer: I like the faceted graph (7) better. The points in 6 are somewhat overlapping, which makes reading the graph hard. Additionally, I think it would be easier for people with color-blindness to read graph 7 because color isn't as important of a component.