Assignment 5: Data Visualization

Fiona Kelley

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.

```r
#1 Loading and installing the necessary packages for this project.

library(tidyverse)
library(lubridate)
library(here)
#install.packages("cowplot")
library(cowplot)
library(ggplot2)

#Checking the working directory.
getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

```r
#Uploading datasets.
PeterPaul_Nutrients <- read.csv(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)
Niwot_Litter <- read.csv(
  "./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

#2 Converting the sample date and collection date columns from factors to dates.
PeterPaul_Nutrients$sampledate <- as.Date(PeterPaul_Nutrients$sampledate)
Niwot_Litter$collectDate <- as.Date(Niwot_Litter$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 Defining a theme.

MyTheme <- theme_classic() +
  theme(
        #Sets text on the axes as black.
        axis.text = element_text(color = 'black'),
        #Positions the legend to the right of plots.
        legend.position = ("right"),
        #Outlines and fills the legend pink.
        legend.background = element_rect(color = 'pink', fill = 'pink'),
        #Outlines and fills the background grey.
        plot.background = element_rect(color = 'grey', fill = 'grey'),
        #Sets grid lines for the plot as grey.
        panel.grid.minor = element_line(color = "grey"),
        panel.grid.major = element_line(color = "grey")
)

#Setting MyTheme as the default theme.
theme_set(MyTheme)
```

Create graphs

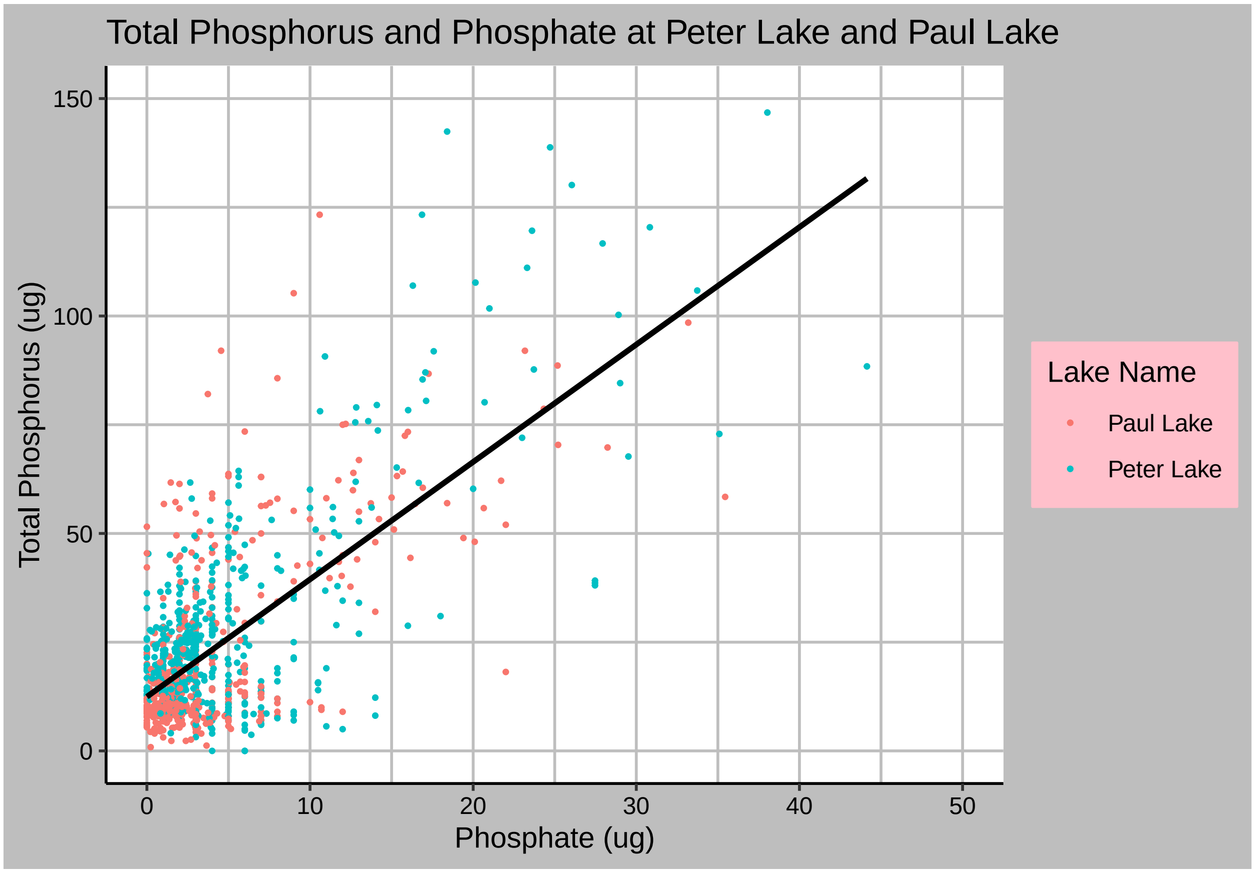
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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 Creating a scatter plot of total phosphorus and
# phosphate for Peter Lake and Paul Lake.

PeterPaul_phosphorus_po4_plot <- ggplot(PeterPaul_Nutrients,
  aes(x = po4, y = tp_ug, color = lakename)) + geom_point(na.rm = TRUE, size = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  #Changing the extent of the x and y axes.
  xlim(c(0, 50)) + ylim(c(0, 150)) +
  labs(x = "Phosphate (ug)", y = "Total Phosphorus (ug)", color = "Lake Name",
       title = "Total Phosphorus and Phosphate at Peter Lake and Paul Lake")

PeterPaul_phosphorus_po4_plot
```

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

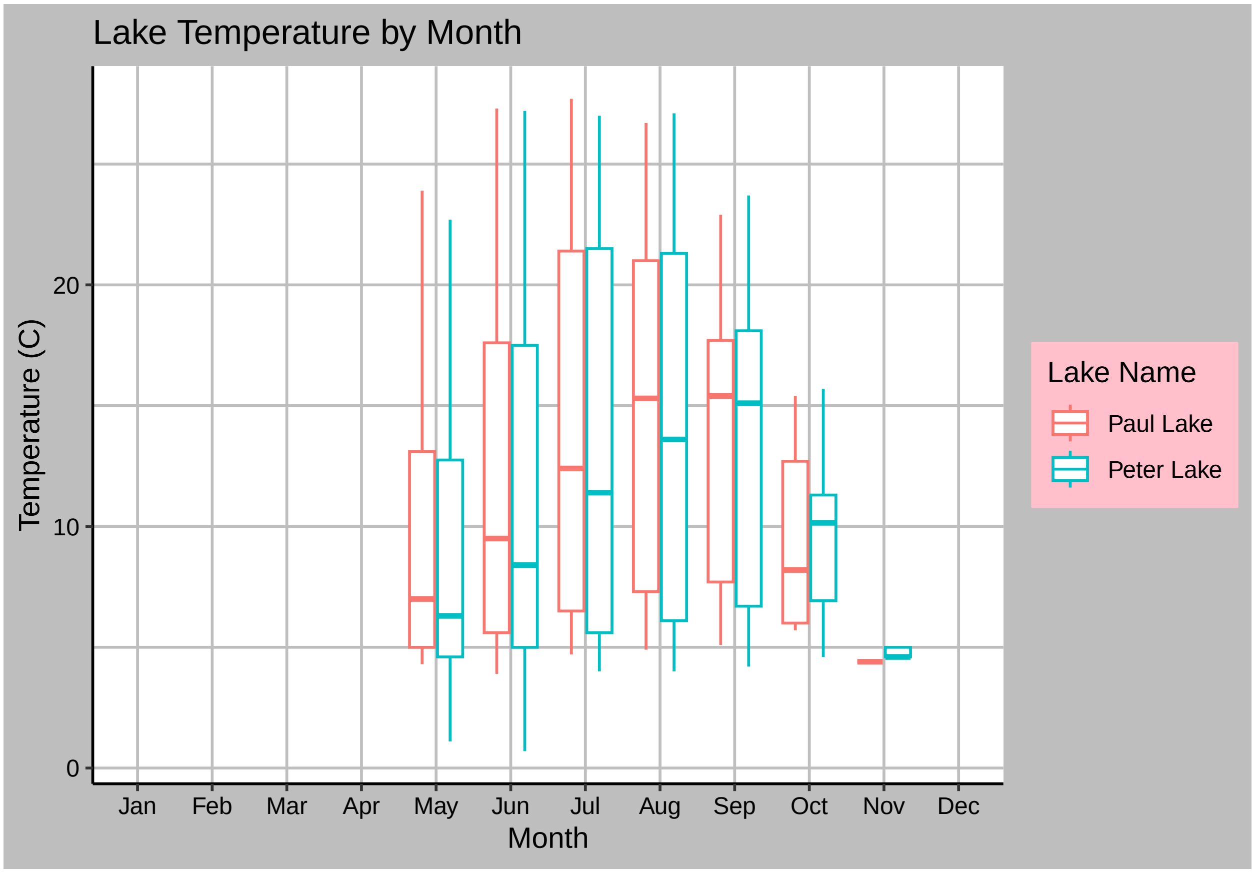
```
## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).
```



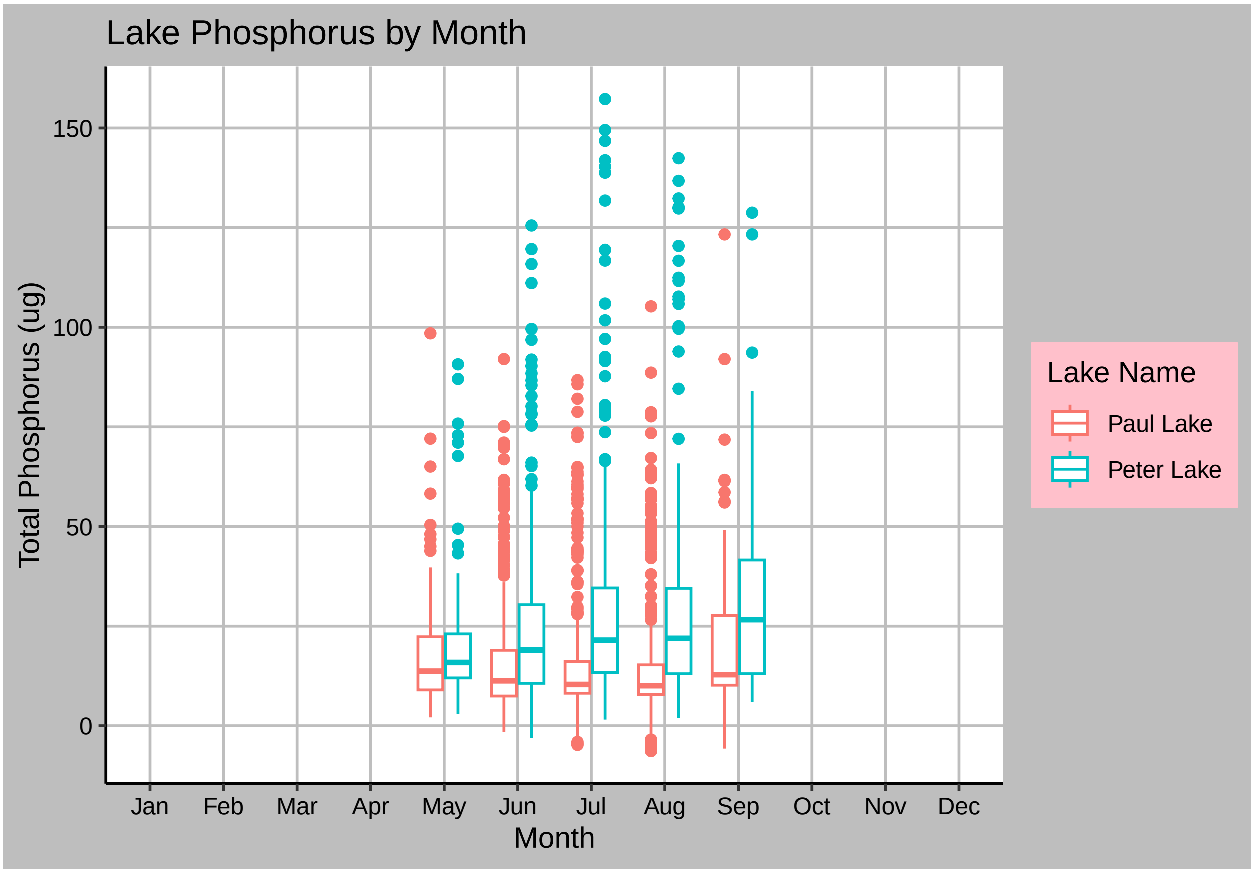
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

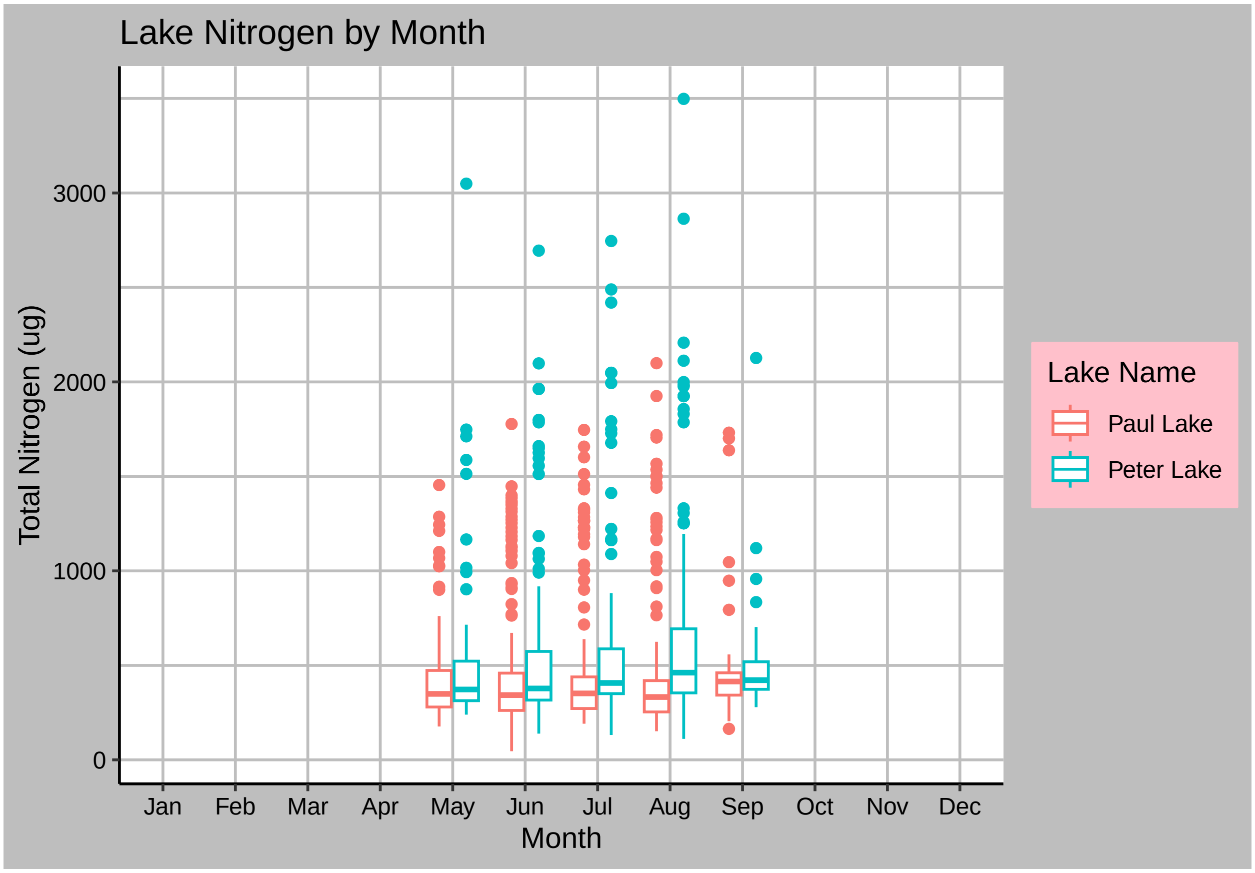
```
#5 Creating separate boxplots for the desired variables.

# Creating a boxplot for temperature by month.
PeterPaul_temp_plot <- ggplot(
  PeterPaul_Nutrients, aes(x = month.abb[month], y = temperature_C,
  color = lakename)) +
  geom_boxplot(na.rm = TRUE) +
  scale_x_discrete(limits = month.abb) +
  labs(x = "Month", y = "Temperature (C)", color = "Lake Name",
       title = "Lake Temperature by Month")

PeterPaul_temp_plot
```



```
# Creating a boxplot for total phosphorus by month.
PeterPaul_tp_plot <- ggplot(
  PeterPaul_Nutrients, aes(x = month.abb[month], y = tp_ug, color = lakename)) +
  geom_boxplot(na.rm = TRUE) +
  scale_x_discrete(limits = month.abb) +
  labs(x = "Month", y = "Total Phosphorus (ug)", color = "Lake Name",
       title = "Lake Phosphorus by Month")

PeterPaul_tp_plot
```

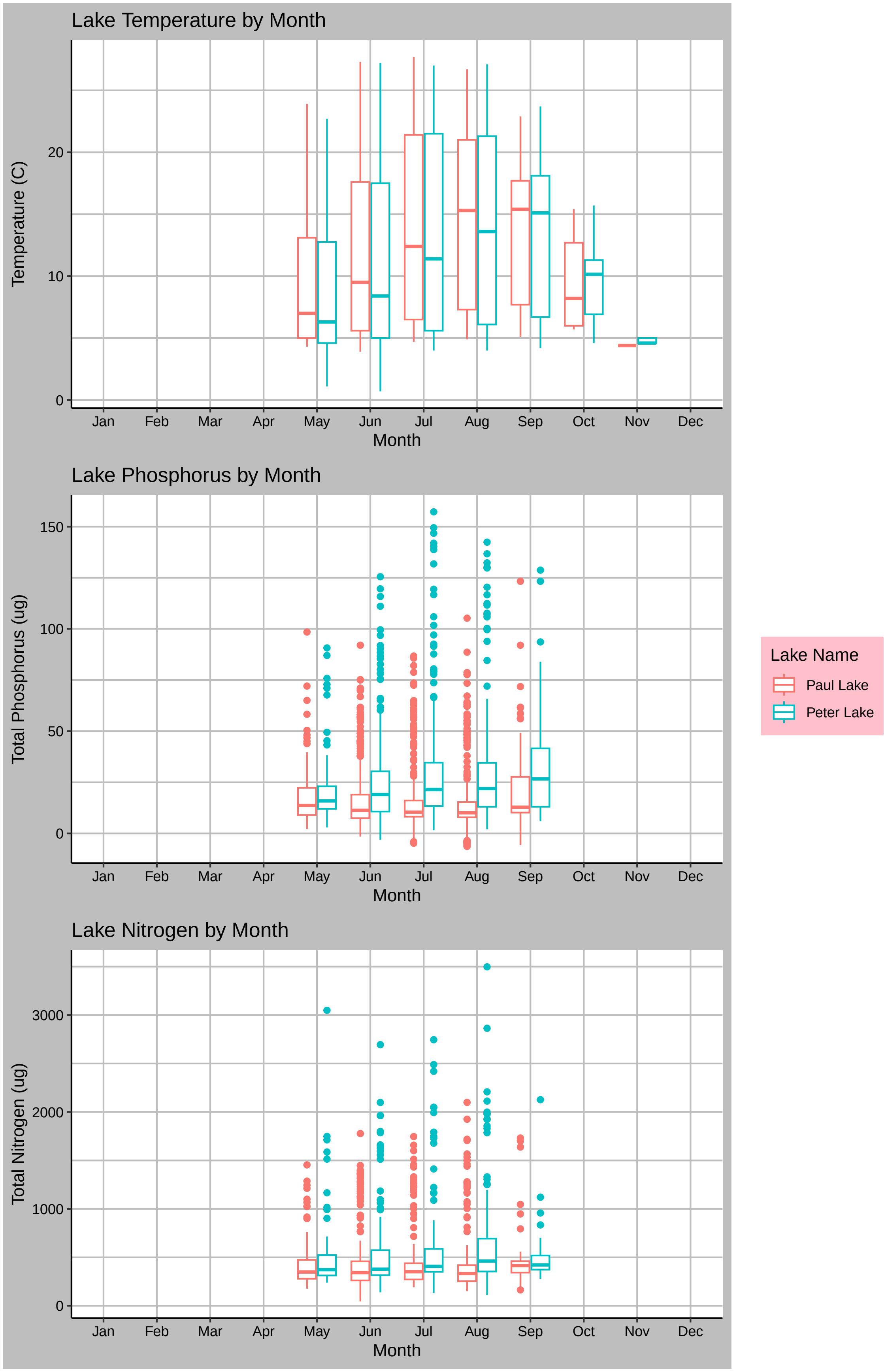


```
# Creating a boxplot for total nitrogen by month.
PeterPaul_tn_plot <- ggplot(
  PeterPaul_Nutrients, aes(x = month.abb[month], y = tn_ug, color = lakename)) +
  geom_boxplot(na.rm = TRUE) +
  scale_x_discrete(limits = month.abb) +
  labs(x = "Month", y = "Total Nitrogen (ug)", color = "Lake Name",
       title = "Lake Nitrogen by Month")

PeterPaul_tn_plot
```



```
# Chat GPT was utilized in this section for understanding and
# implementing the month.abb function.
```

```
# Combining the three boxplots onto one graph.
PeterPaul_cowplot <- plot_grid(
  PeterPaul_temp_plot + theme(legend.position = "none"),
  PeterPaul_tp_plot + theme(legend.position = "none"),
  PeterPaul_tn_plot + theme(legend.position = "none"),
  nrow = 3, align = "hv", rel_heights = c(1,1,1)) +
  theme(axis.text = element_text(size = 14))

# Extract one legend for the plot.
Legend_cowplot <- get_legend(PeterPaul_temp_plot)

# Add legend to plot.
PeterPaul_cowplot <- plot_grid(PeterPaul_cowplot,
                      Legend_cowplot,
                      rel_widths = c(1,0.25)
)

PeterPaul_cowplot
```

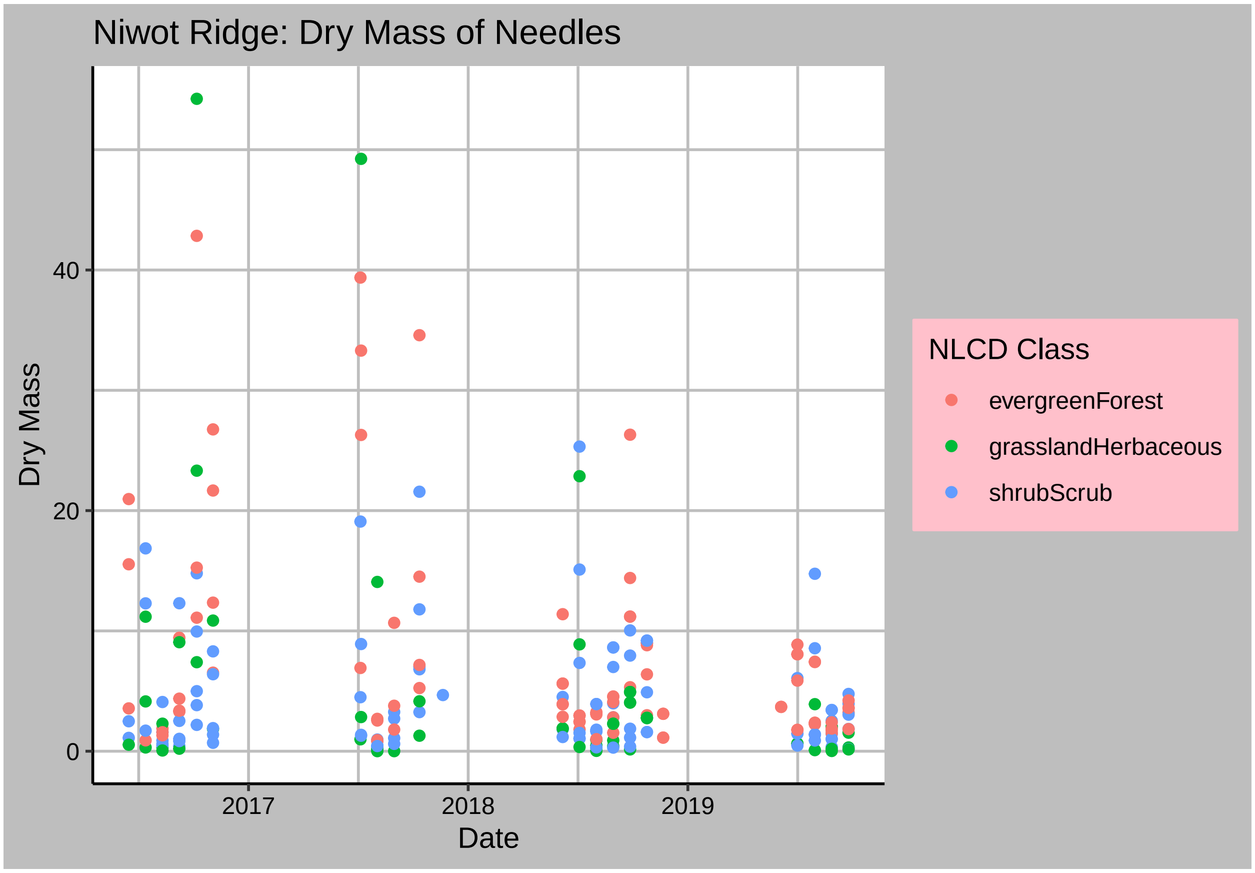

Lake Temperature by Month
Temperature (C)
0
10
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
Lake Phosphorus by Month
Total Phosphorus (ug)
0
50
100
150
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
Lake Nitrogen by Month
Total Nitrogen (ug)
0
1000
2000
3000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
Lake Name
Paul Lake
Peter Lake

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

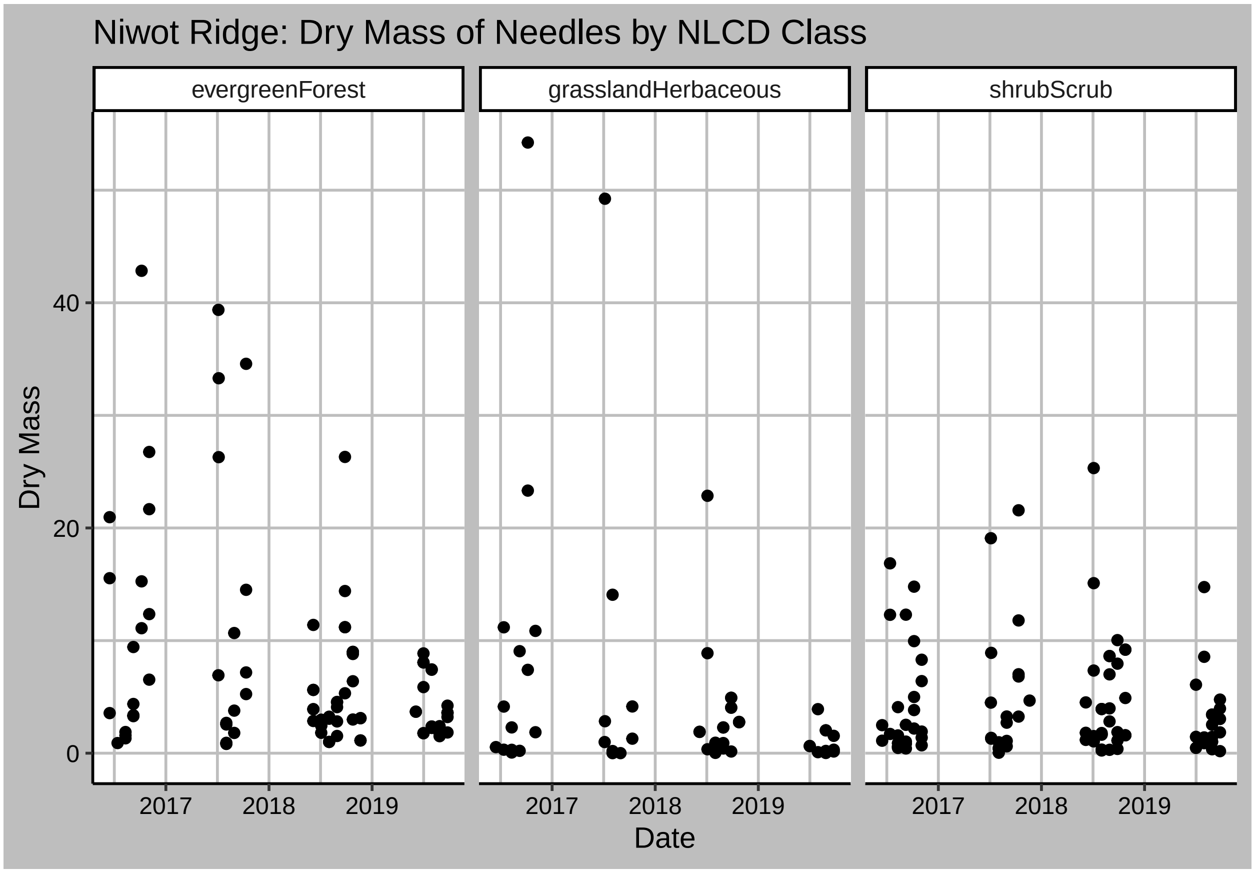
> Answer: Looking at Peter Lake, median values for total phosphorus and nitrogen were higher than that of Paul Lake. The ranges for phosphorus and nitrogen also have a greater spread at Peter Lake. In addition to this, I observed higher median values for total phosphorus during months with higher median temperatures, specifically in July, August, and September.

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 Creating a scatter plot for the dry mass of "Needles" functional group.

NW_Needles_plot <- ggplot(
  subset(Niwot_Litter, functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass, color = nlcdClass)) + geom_point() +
  labs(x = "Date", y = "Dry Mass", color = "NLCD Class",
       title = "Niwot Ridge: Dry Mass of Needles")

NW_Needles_plot
```



```r
#7 Separating the plots by NLCD class using facet_wrap.
NW_Needles_facets <- ggplot(
  subset(Niwot_Litter, functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass)) + geom_point() +
  labs(x = "Date", y = "Dry Mass", color = "NLCD Class",
       title = "Niwot Ridge: Dry Mass of Needles by NLCD Class") +
  facet_wrap(vars(nlcdClass), ncol = 3)

NW_Needles_facets
```



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

> Answer: The graph separated into three facets by NLCD Class is more effective because the data between classes do not overlap one another, making it easier to determine trends between class types.