Assignment 5: Data Visualization

David Liddle

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 Load necessary packages
#install.packages("tidyverse")
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 1.9.3     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
```

```
#install.packages("lubridate")
library(lubridate)
#install.packages("here")
library(here)
```

```
## here() starts at /Users/davidliddle/Documents/EDA-Spring2023
```

```
#install.packages("cowplot")
library(cowplot)
```

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

```
#Assign a variable to the processed data folder location
processed_data = "Data/Processed_KEY"

#Read in data
PeterPaul.chem.nutrients <- read.csv(
here(processed_data,
      "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
stringsAsFactors = TRUE)
Neon_NIWO_Litter <- read.csv(
here(processed_data,
       "NEON_NIWO_Litter_mass_trap_Processed.csv"))


#2 Fix dates
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Neon_NIWO_Litter$collectDate <- ymd(Neon_NIWO_Litter$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 Create a personal plot theme
david_theme <- theme_classic(base_size = 14) +
theme(
axis.text = element_text(color = "black"),
legend.position = "bottom",
panel.grid.minor = element_line(color = "gray", linetype = "solid"),
panel.grid.major = element_line(color = "gray", linetype = "solid"),
legend.background = element_rect(fill = "gray"))
#Set the theme
theme_set(david_theme)
```

Create graphs

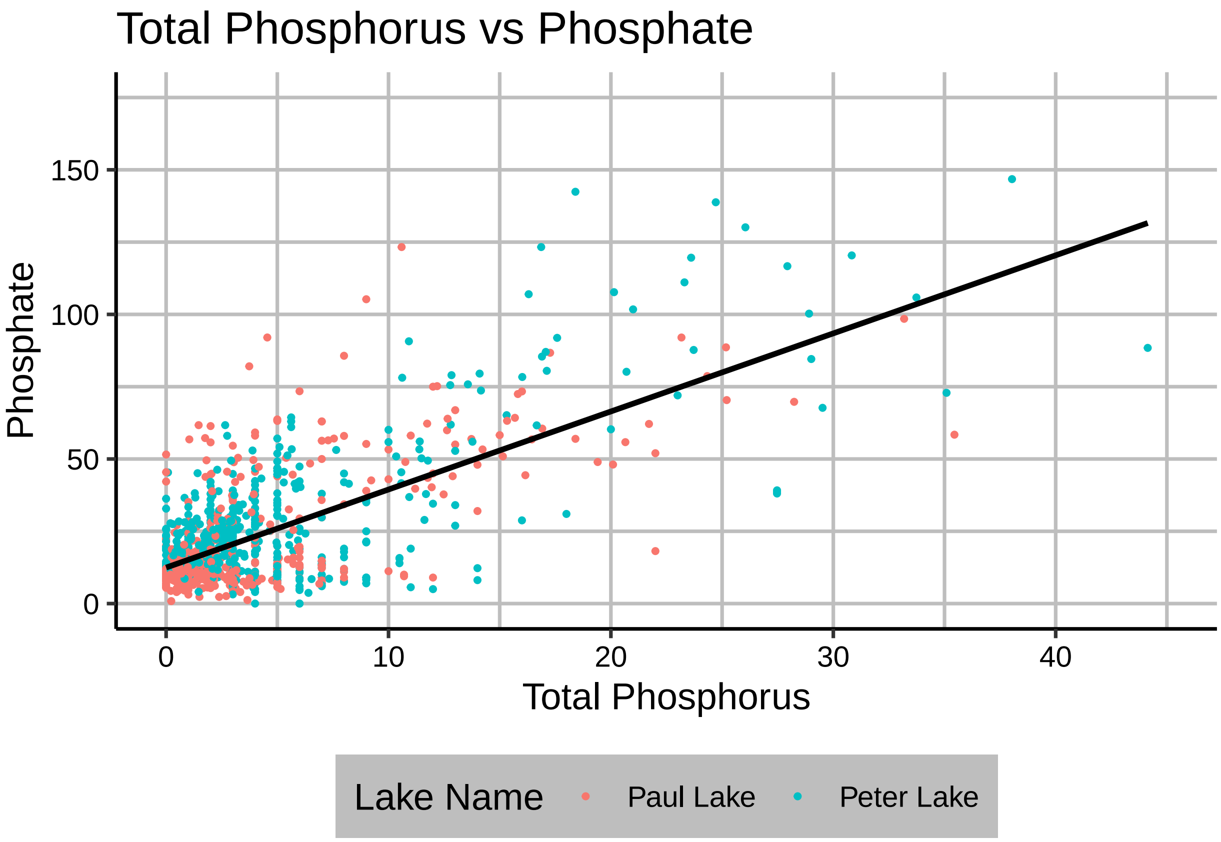
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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()`).

```
#4 Create Plot for total phosphorus by phosphate
plot <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug,
                                             color = lakename)) +
  # Add points for each data point, remove n/a values, and resize
  geom_point(na.rm = TRUE, size = 0.75) +
  # Add a line of best fit in black
  geom_smooth(method = "lm", se = FALSE, col = "black") +
  labs(title = "Total Phosphorus vs Phosphate",
       x = "Total Phosphorus",
       y = "Phosphate", color = "Lake Name") + # Name axes
   xlim(c(0, 45)) + ylim(c(0, 175)) #Add x and y limits
# Show the plot
print(plot)
```

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

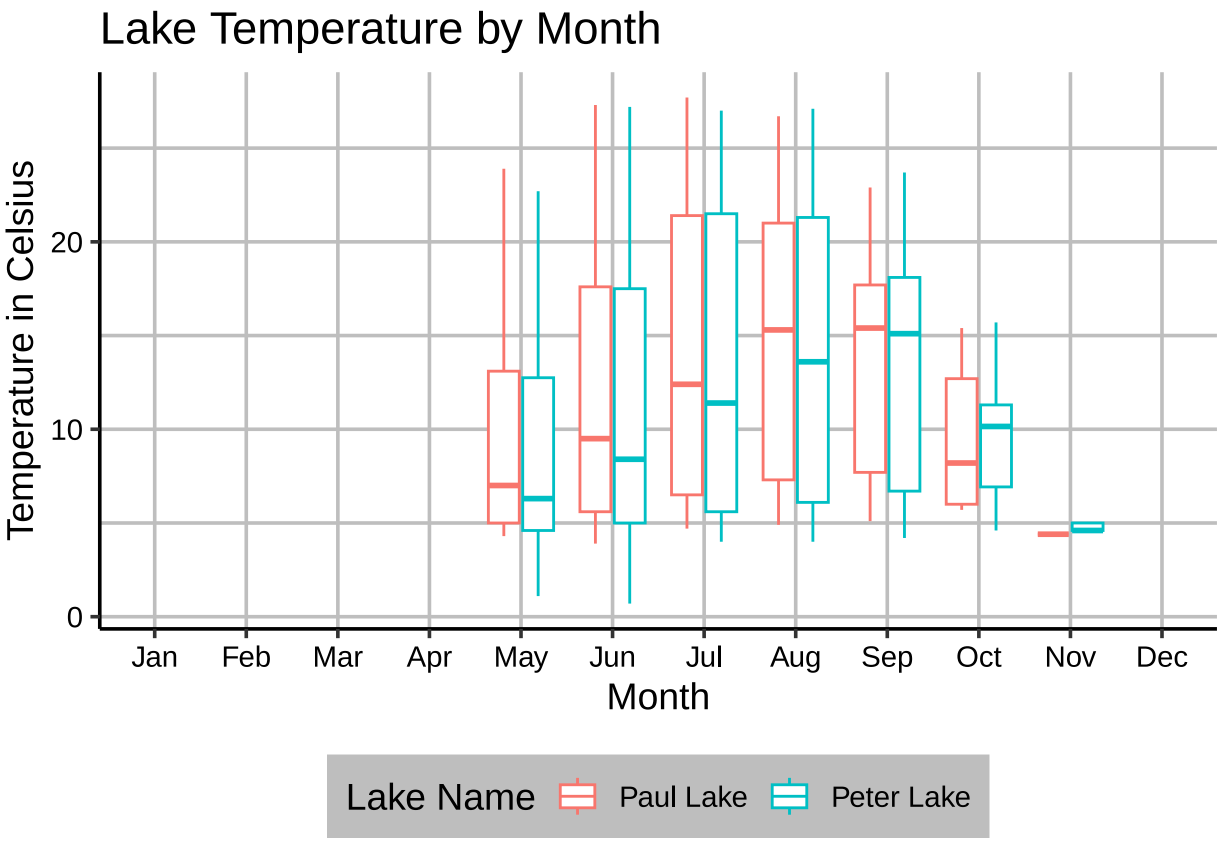
```
## Warning: Removed 21948 rows containing non-finite values (‘stat_smooth()‘).
```



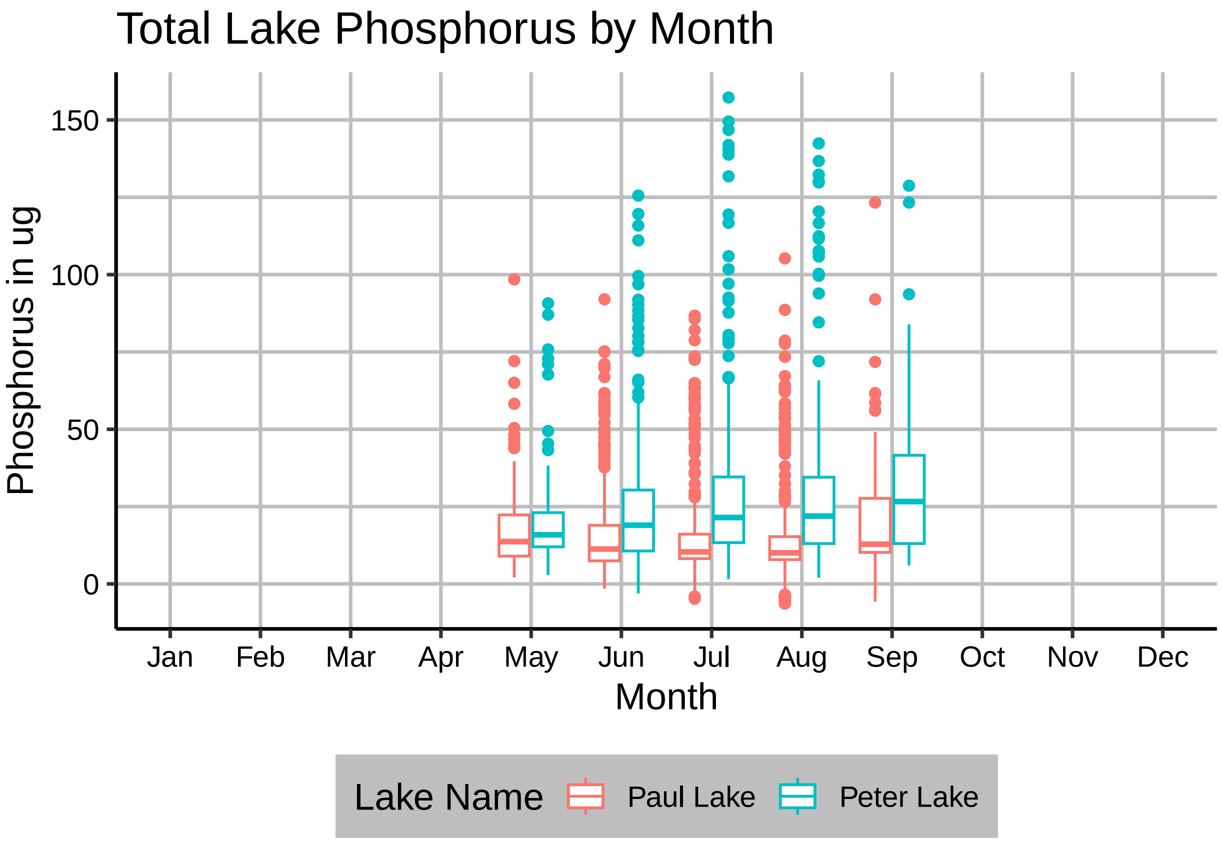
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

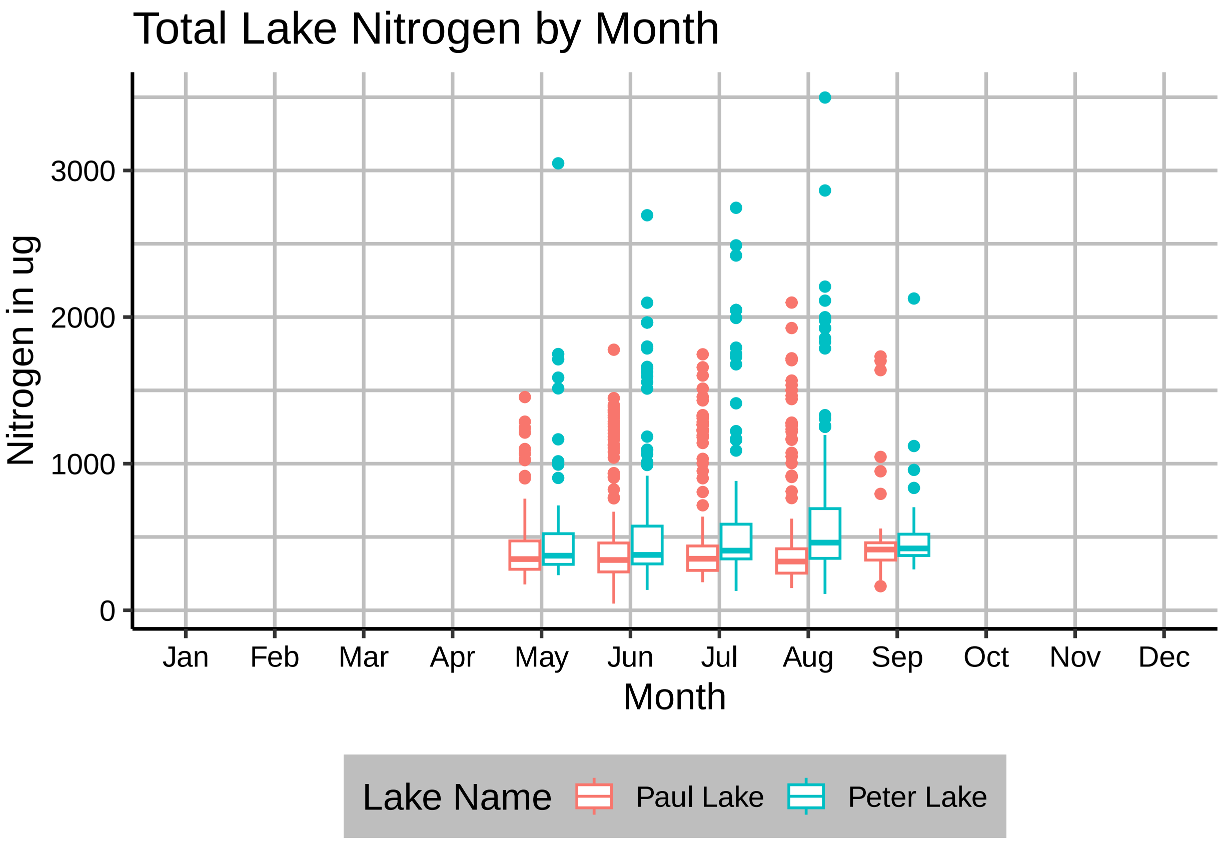
```
#5 a) Create temperature boxplot
temperature_boxplot <- ggplot(PeterPaul.chem.nutrients,
aes( x = month.abb[month], y = temperature_C, color = lakename)) +
geom_boxplot(na.rm = TRUE) + scale_x_discrete(limits = month.abb) +
  #Found this function on ChatGPT, it put the months in correct order
   labs(title = "Lake Temperature by Month", #Create title and label axes
       x = "Month",
       y = "Temperature in Celsius", color = "Lake Name")
print(temperature_boxplot)
```



```
#b) Create TP boxplot
TP_boxplot <- ggplot(PeterPaul.chem.nutrients,
aes( x = month.abb[month], y = tp_ug, color = lakename)) +
geom_boxplot(na.rm = TRUE) +
  scale_x_discrete(limits = month.abb) +
  #Found this function on ChatGPT, it put the months in correct order
   labs(title = "Total Lake Phosphorus by Month", #Create title and label axes
       x = "Month",
       y = "Phosphorus in ug", color = "Lake Name")
print(TP_boxplot)
```

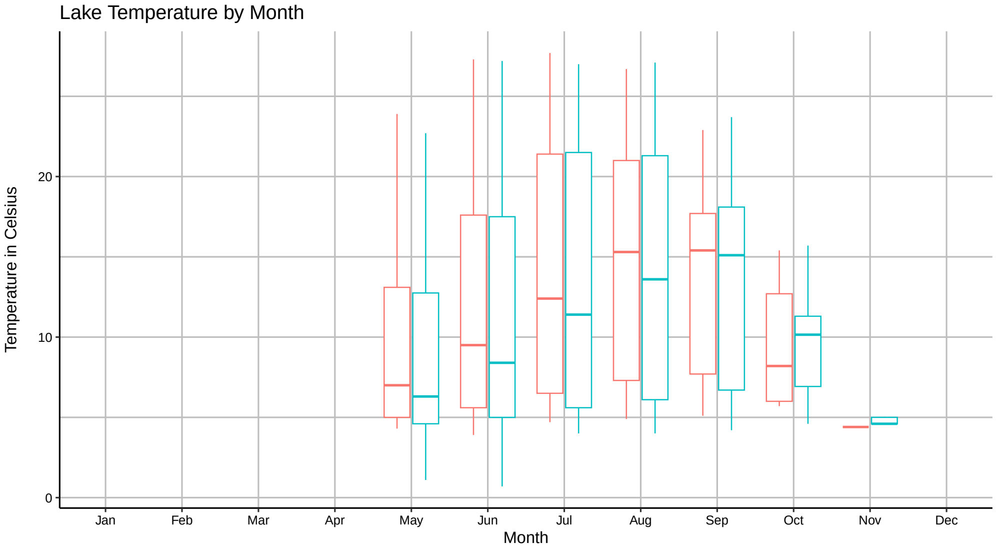


```
#c) Create TN boxplot
TN_boxplot <- ggplot(PeterPaul.chem.nutrients,
aes( x = month.abb[month], y = tn_ug, color = lakename)) +
geom_boxplot(na.rm = TRUE) +
  scale_x_discrete(limits = month.abb) +
  #Found this function on ChatGPT, it put the months in correct order
  labs(title = "Total Lake Nitrogen by Month", #Create title and label axes
       x = "Month",
       y = "Nitrogen in ug", color = "Lake Name")
print(TN_boxplot)
```

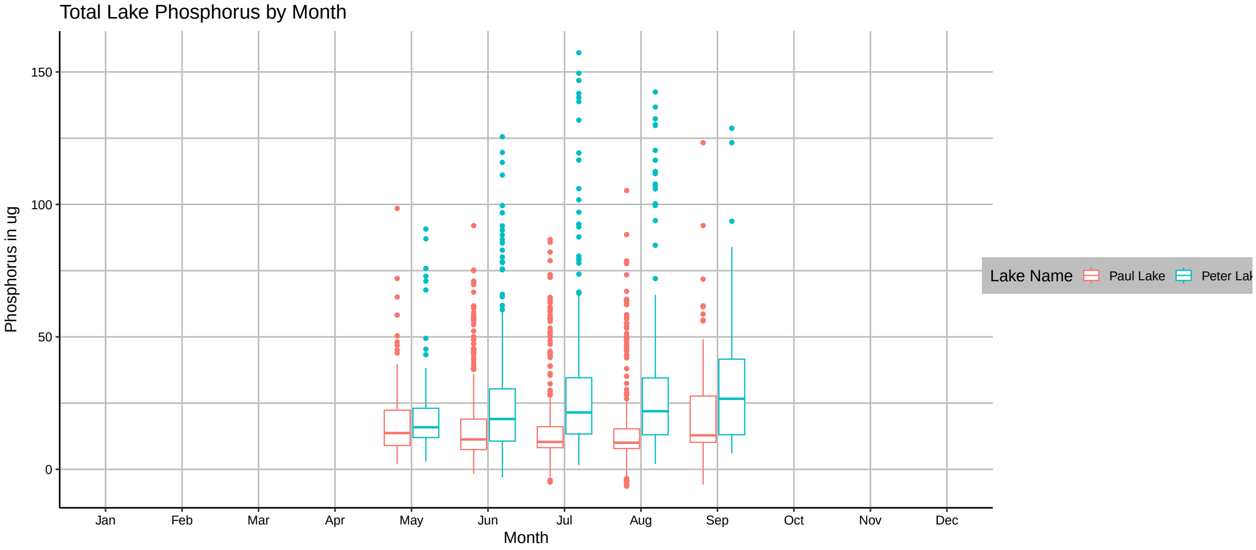


```
# Combine the three boxplots onto one graph
combined_plot <- plot_grid(
# Turn off legends in individual plots
temperature_boxplot + theme(legend.position = "none"),
 TP_boxplot + theme(legend.position = "none"),
  TN_boxplot+ theme(legend.position = "none"),
nrow = 3, align = "hv", rel_heights = c(1,1,1)) +
theme(axis.text = element_text(size = 12))
```

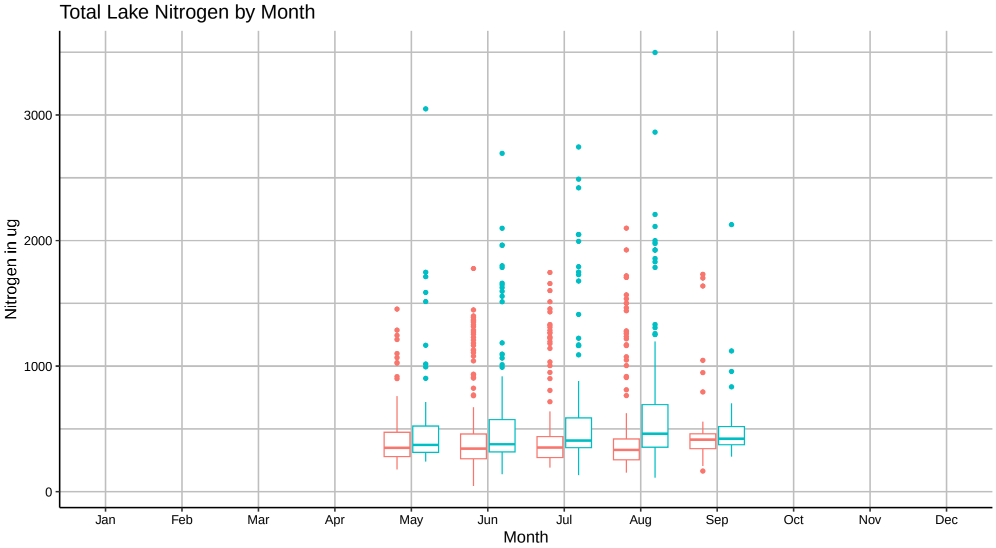
```
#Extract one legend for the combined plots
combined_plot_legend <- get_legend(TN_boxplot)
#Add legend to combined plot
combined_plot <- plot_grid(combined_plot, combined_plot_legend,
ncol = 2, rel_widths = c(1, 0.25))
print(combined_plot)
```



Lake Temperature by Month



Total Lake Phosphorus by Month



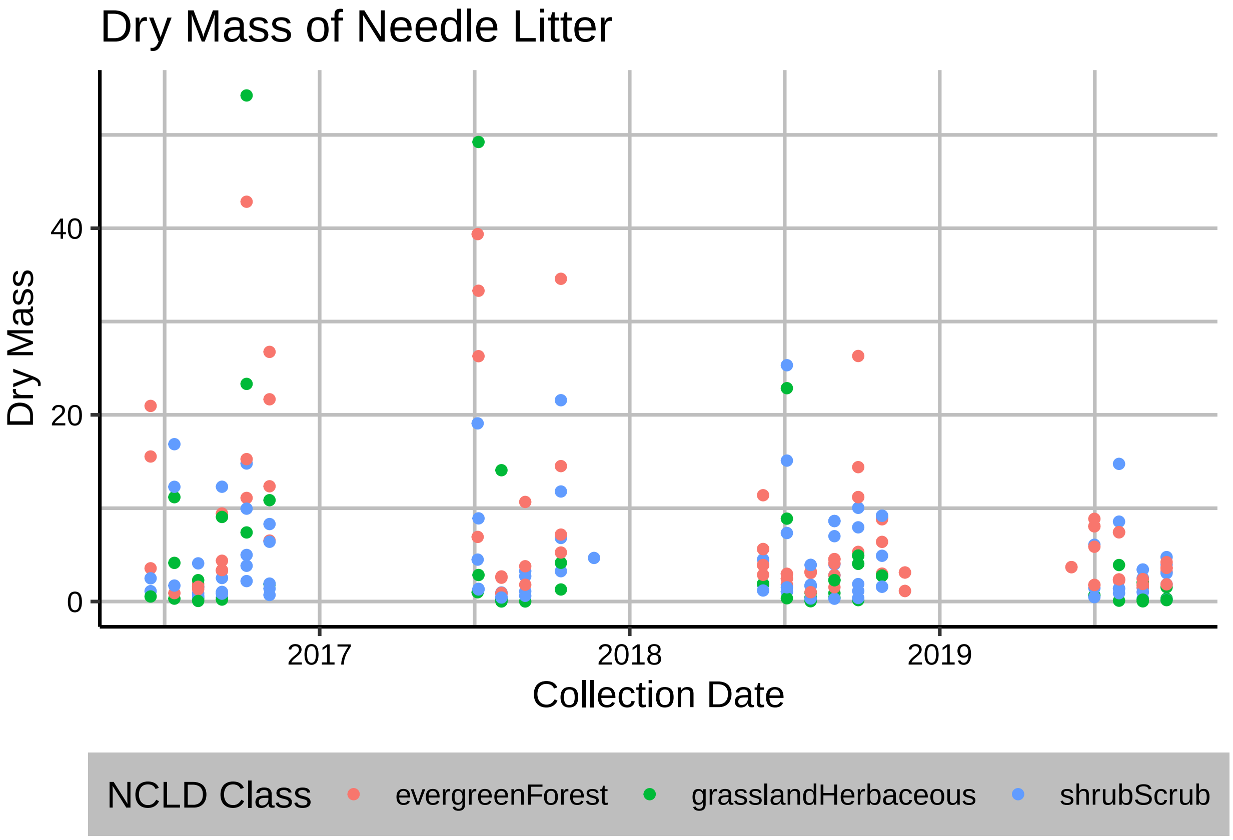
Total Lake Nitrogen by Month

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

Answer: One thing that was noted was that the average phosphorus and nitrogen in Peter Lake is slightly higher than Lake Paul for all of the measured months. Additionally, phosphorus levels in Peter Lake increased slightly as the water temperature increased, while levels decreased in Paul Lake as water temperature increased. Temperature did not appear to affected the nitrogen levels significantly in either lake. Also, there was a greater spread in the range of the data in Peter Lake for both nitrogen and phosphorus levels.

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.

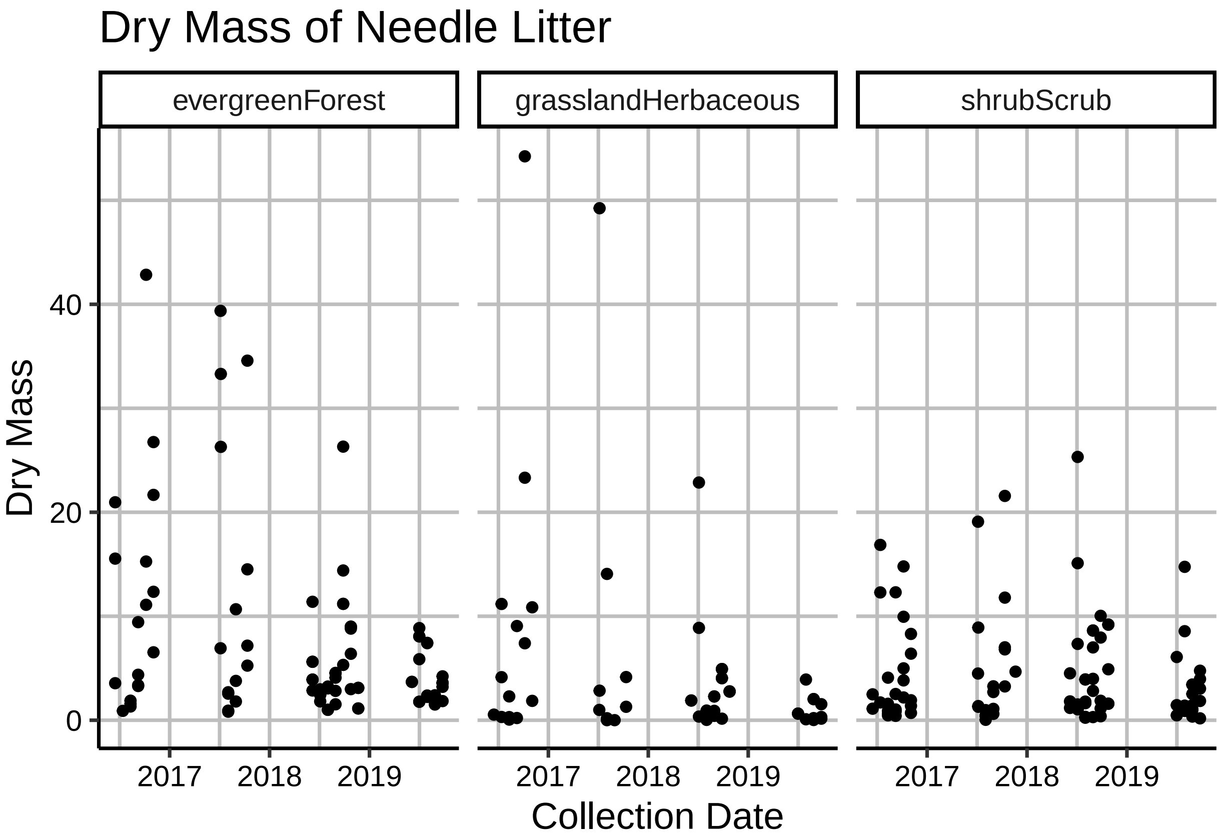
```r
#6 Plot the litter subset
litter_plot <- ggplot(subset(Neon_NIWO_Litter, functionalGroup == "Needles"),
               aes(x = collectDate, y = dryMass,
               color = nlcdClass)) + geom_point(na.rm = TRUE) +
    labs(title = "Dry Mass of Needle Litter", #Create title and label axes
       x = "Collection Date",
       y = "Dry Mass", color = "NCLD Class")
litter_plot
```



```r
#7 Recreate the same graph but separated into facets

litter_plot_faceted <- ggplot(subset(Neon_NIWO_Litter,
                          functionalGroup == "Needles"),
                          aes(x = collectDate, y = dryMass)) +
  geom_point(na.rm = TRUE) +
  labs(title = "Dry Mass of Needle Litter",
       x = "Collection Date",
       y = "Dry Mass") +
  # Use facet_wrap to create facets based on nlcdClass
  facet_wrap(vars(nlcdClass), ncol = 3)

# Print the plot
print(litter_plot_faceted)
```



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

Answer: I think that plot 7 is more effective since there is no overlap of the points between the NCLD classes. This makes it so that the trends between the NCLD class are more easily observable,