Assignment 5: Data Visualization

Nikki Shintaku

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Salk_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and cowplot packages. Upload the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (tidy and gathered) and the processed data file for the Niwot Ridge litter dataset.

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
getwd()
```

```
## [1] "/Users/nikkishintaku/Desktop/Environmental872/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)
```

```
##
## ******************************************************
## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ********************************************************
```

```r
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
```

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

```r
PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
Neonics <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d"

PeterPaul.chem.nutrients.gathered$sampledate <- as.Date(PeterPaul.chem.nutrients.gathered$sampledate, f

Neonics$collectDate <- as.Date(Neonics$collectDate, format = "%Y-%m-%d")
```

Define your theme

3. Build a theme and set it as your default theme.

```r
mytheme <- theme_stata(base_size = 14, base_family = "sans", scheme = "s2mono") +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)
```

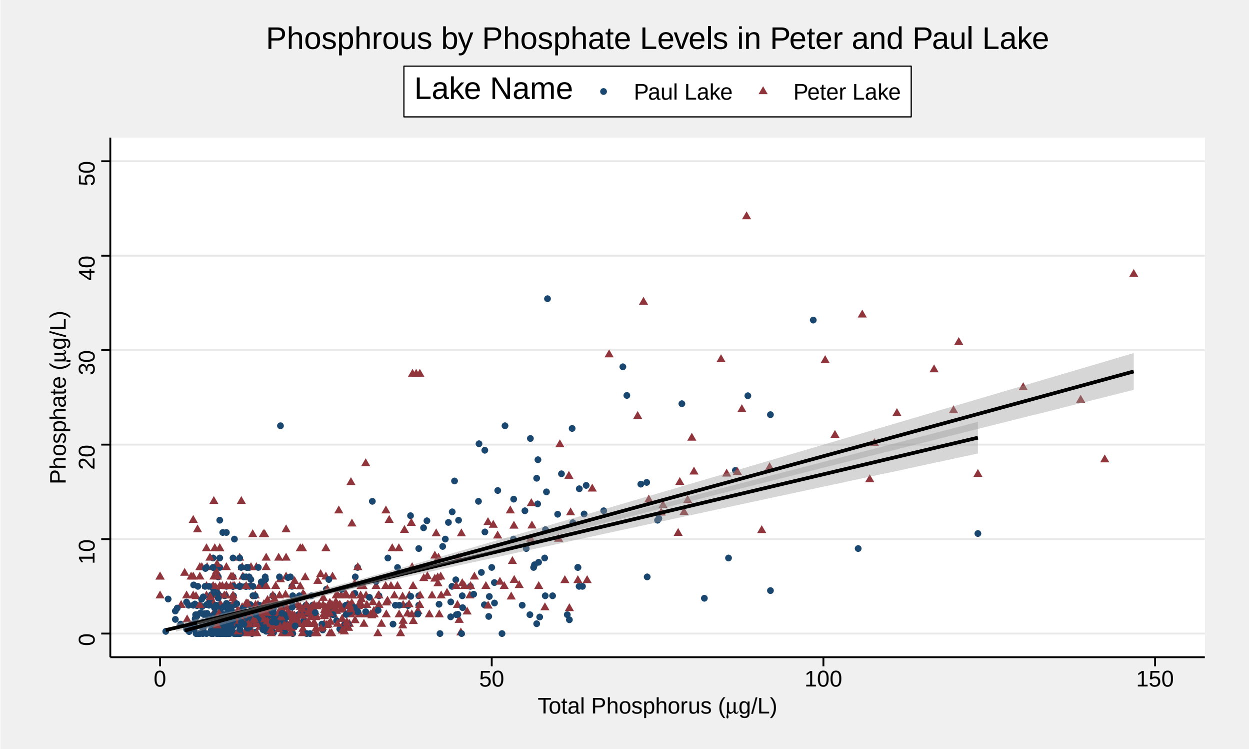
Create graphs

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 by phosphate, 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.

```
PO4_plot <- ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4, group = lakename)) +
  geom_point(aes(shape = lakename, color = lakename)) +
  labs(x = expression(paste("Total Phosphorus (", mu, "g/L)")), y = expression(paste("Phosphate (", mu,
  scale_color_stata("s2color") +
  xlim(0, 150) +
  ylim(0, 50) +
  geom_smooth(method = lm, color = "black")
  #geom_smooth(data = subset(PeterPaul.chem.nutrients, lakename=="Paul Lake"), method = lm, color = "bl

print(PO4_plot)
```

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

## Warning: Removed 21948 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_smooth).
```

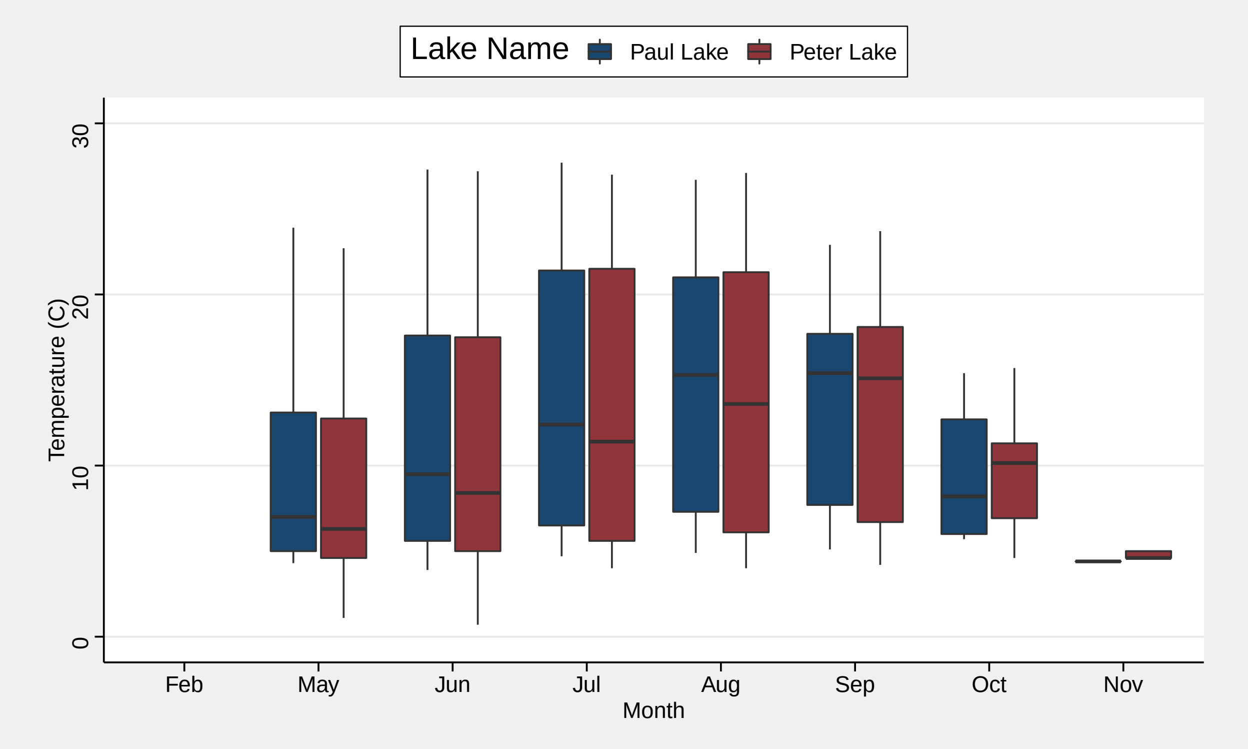


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
#change month number to month abbreviation
PeterPaul.chem.nutrients$month <- month.abb[PeterPaul.chem.nutrients$month]

#put months in the correct order
level_order <- c("Feb", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov")
```

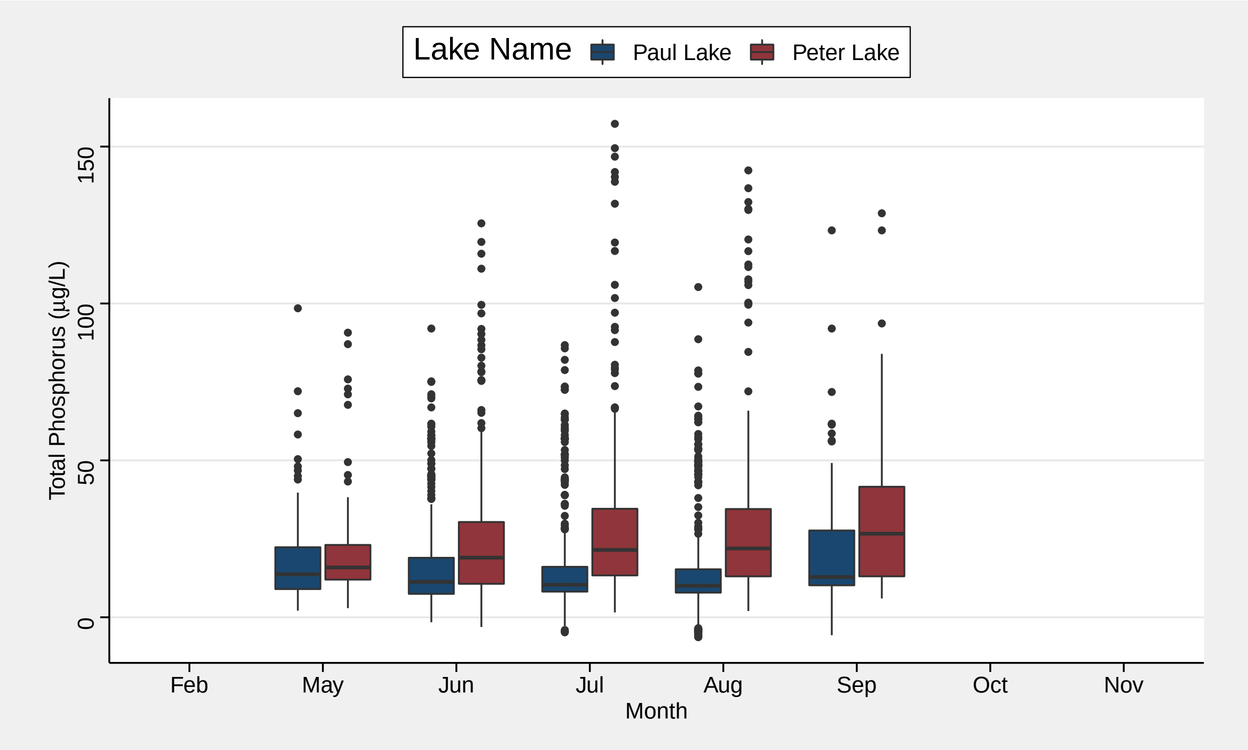
```
#Month vs Temperature boxplot
temp_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, level = level_order), y = temperature_C, fill
  geom_boxplot() +
  labs(x = "Month", y = "Temperature (C)", fill = "Lake Name") +
  scale_fill_stata(scheme = "s2color") +
  ylim(0,30)
print(temp_boxplot)
```

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



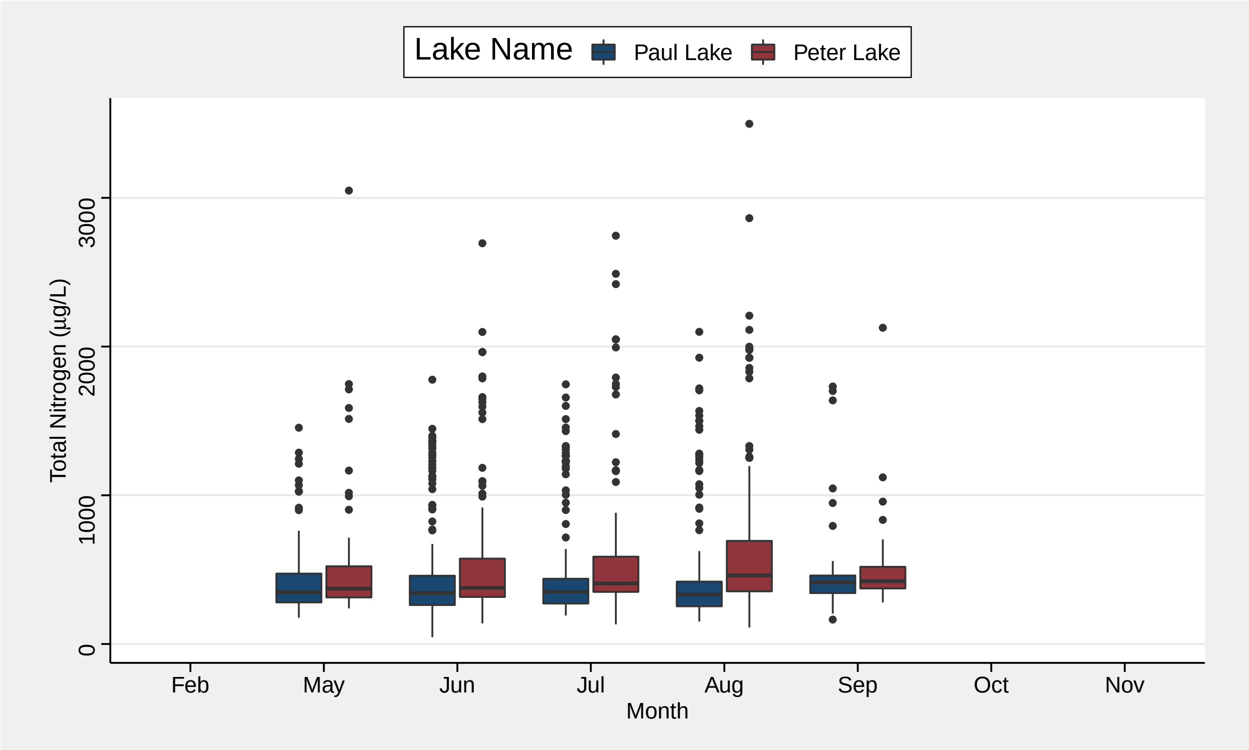
```
#Month vs Total phosphorus
TP_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, level = level_order),
        y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("Total Phosphorus (", mu, "g/L)")),
       fill = "Lake Name") +
  scale_fill_stata("s2color")
print(TP_boxplot)
```

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



```
#month vs total nitrogen
TN_boxplot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = factor(month, level = level_order),
                                       y = tn_ug, fill = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = expression(paste("Total Nitrogen (", mu, "g/L)")),
       fill = "Lake Name") +
  scale_fill_stata("s2color")
print(TN_boxplot)
```

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



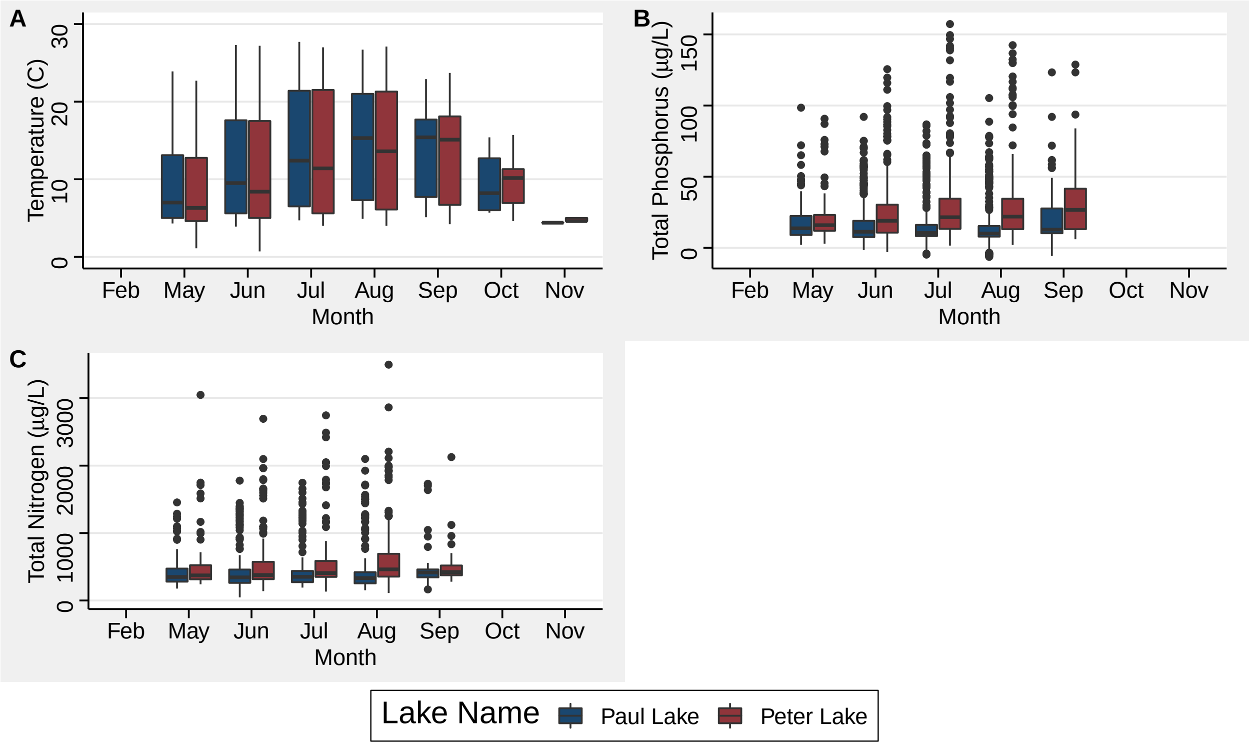
```
#create a grid of the three boxplots with the legends removed
boxplots_combined <- plot_grid(temp_boxplot + theme(legend.position = "none"),
          TP_boxplot + theme(legend.position = "none"),
          TN_boxplot + theme(legend.position = "none"),
          labels = "AUTO",
          align = "h",
          axis = "b")
```

```
## 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).
```

```
#print(boxplots_combined)

#grab legend from one of the boxplots
legend <- get_legend(temp_boxplot + guides(color = guide_legend(nrow = 1)) +
                       theme(legend.position = "bottom"))
```

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

```
#plot the three boxplots and manually add in the legend
plot_grid(boxplots_combined, legend, ncol = 1, rel_heights = c(1, .1))
```

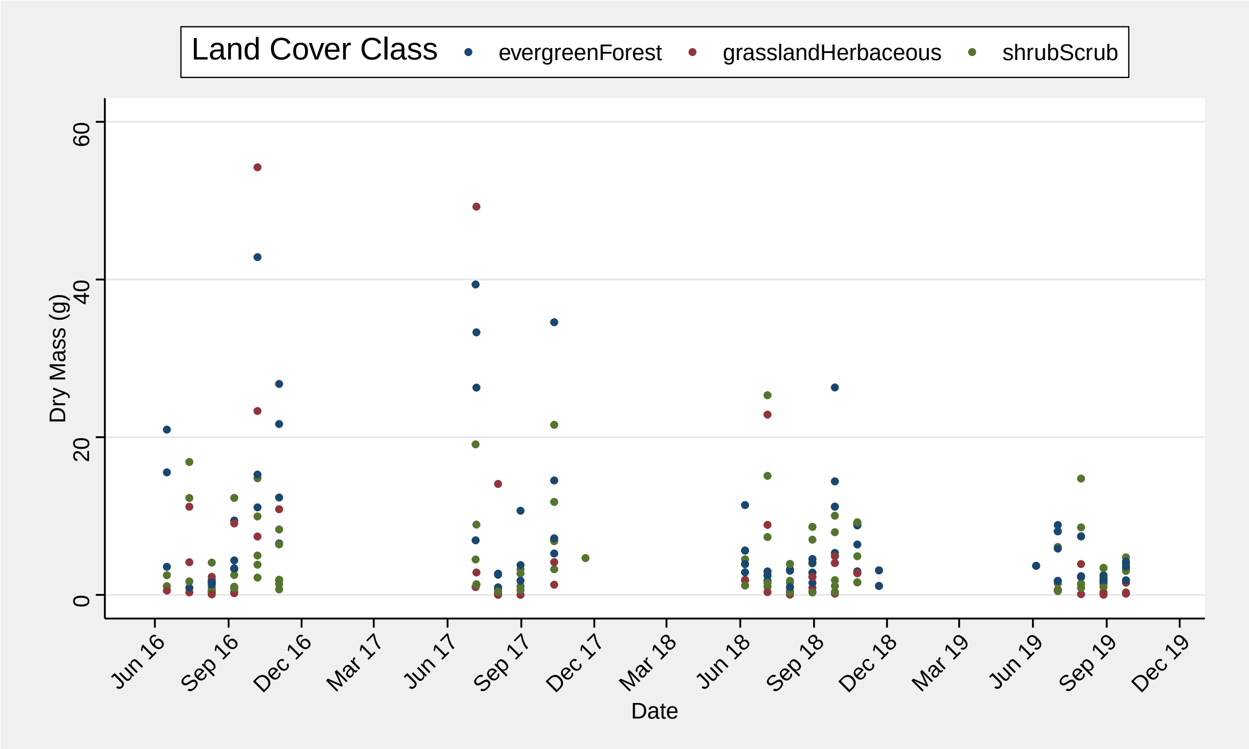


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Throughout the seasons, the temperature between Peter and Paul Lake are very similar and do not differ within the same month. Temperatures have a larger range during the summer months than the fall/winter months. Total phosphorus levels are greater in Peter Lake than Paul Lake, and the phosphorus levels have high concentrations during the summer months relative to the fall months in both lakes. Total nitrogen levels have a greater range in Peter Lake than Paul Lake, and the levels also are higher during the summer months than the fall months for both lakes.

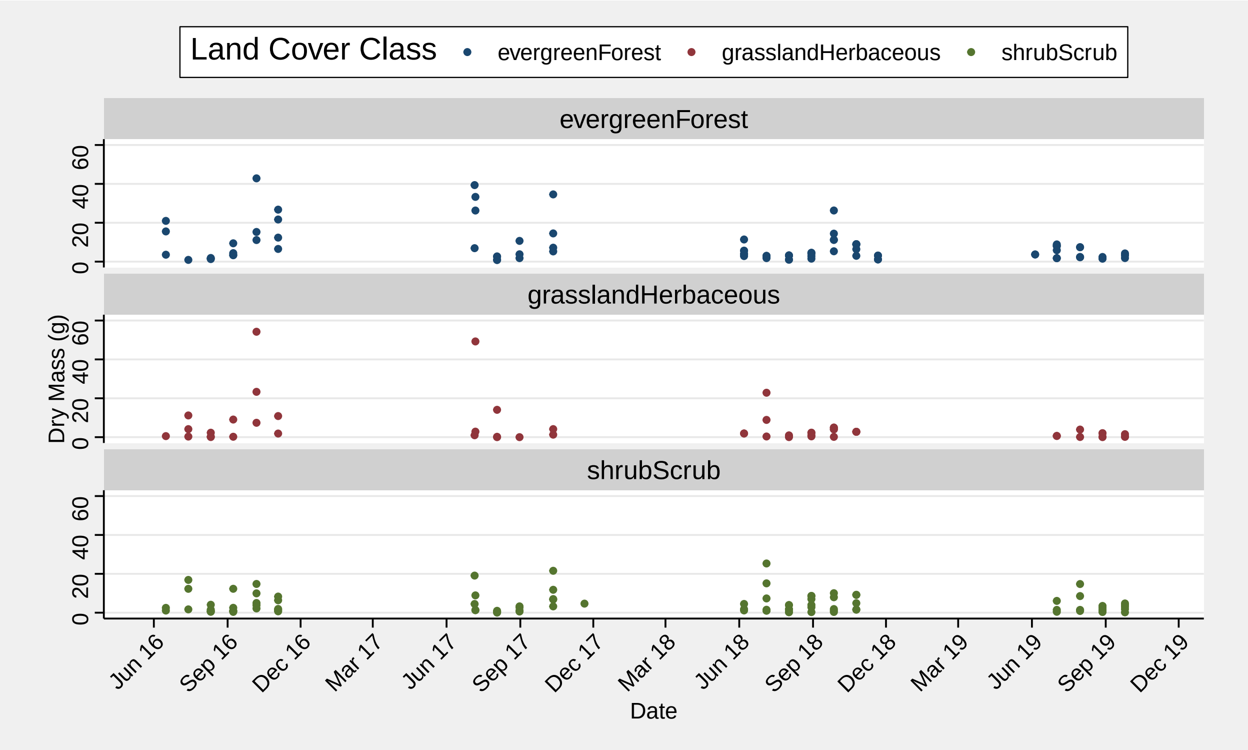
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

```
#first and last date 2016-06-16 UTC--2019-09-25 UTC

needles_graph <-
  ggplot(subset(Neonics, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass)) +
  geom_point(aes(color = nlcdClass)) +
  labs(x = "Date", y = "Dry Mass (g)", color = "Land Cover Class") +
  scale_color_stata("s2color") +
  ylim(0, 60) +
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-10-31")),
        date_breaks = "3 months", date_labels = "%b %y") +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))

print(needles_graph)
```



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

```r
needles_faceted <-
  ggplot(subset(Neonics, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass)) +
  geom_point(aes(color = nlcdClass)) +
  labs(x = "Date", y = "Dry Mass (g)", color = "Land Cover Class") +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  scale_color_stata("s2color") +
  ylim(0, 60) +
  scale_x_date(limits = as.Date(c("2016-06-01", "2019-10-31")),
        date_breaks = "3 months", date_labels = "%b %y") +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))

print(needles_faceted)
```



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

Answer: I believe the facet wrap plot from #7 is more effective because we can clearly see each land cover class and the associated dry mass from needles for each of collection dates. The data points from the plot in #6 were on top of each other and it is hard to tell what the dry mass amount goes to what land cover class. With the facet wrap each of the classes and dry mass amounts are easy comparable.