Assignment 5: Data Visualization

Ardath Dixon

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., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) 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() # confirm working directory. use setwd() to change if needed.
```

```
## [1] "C:/Users/ardat/OneDrive/Documents/DataAnalytics/Environmental_Data_Analytics_2021"
```

```
library(tidyverse) # load tidyverse package
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

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

```
library(cowplot) # load cowplot package

# Upload data files of interest
Nutr_PeterPaul <- read.csv(
  './Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv')
```

```
Nutr_PeterPaul_tidy <- read.csv(
  './Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv')

#2 Change date columns to date format, and confirm changes
Nutr_PeterPaul$sampledate <- as.Date(Nutr_PeterPaul$sampledate, format = "%m/%d/%Y")
class(Nutr_PeterPaul$sampledate)
```

```
## [1] "Date"
```

```
Nutr_PeterPaul_tidy$sampledate <- as.Date(Nutr_PeterPaul_tidy$sampledate, format = "%Y-%m-%d")
class(Nutr_PeterPaul_tidy$sampledate)
```

```
## [1] "Date"
```

Define your theme

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

```
mytheme <- theme_bw(base_size = 9)+ theme(axis.text = element_text(color="dark gray"))
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.

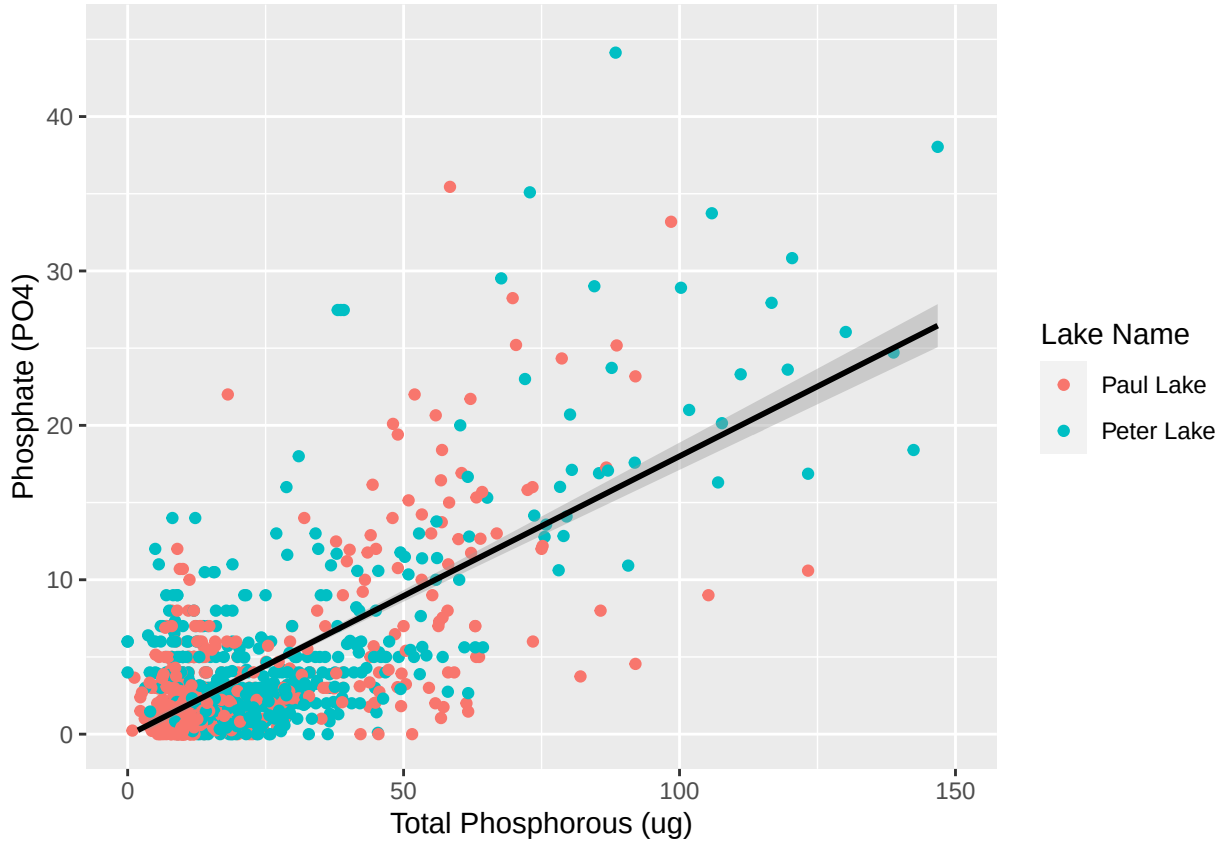
```
TPug_PO4_plot <- ggplot(Nutr_PeterPaul, aes(x=tp_ug, y=po4, color = lakename))+
  geom_point()+
  geom_smooth(method=lm, color = "black")+
  theme_gray()+
  xlim(0,150)+
  ylim(0,45)+
  labs(color = "Lake Name", x= "Total Phosphorous (ug)", y = "Phosphate (PO4)")
print(TPug_PO4_plot)
```

```
## `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).
```

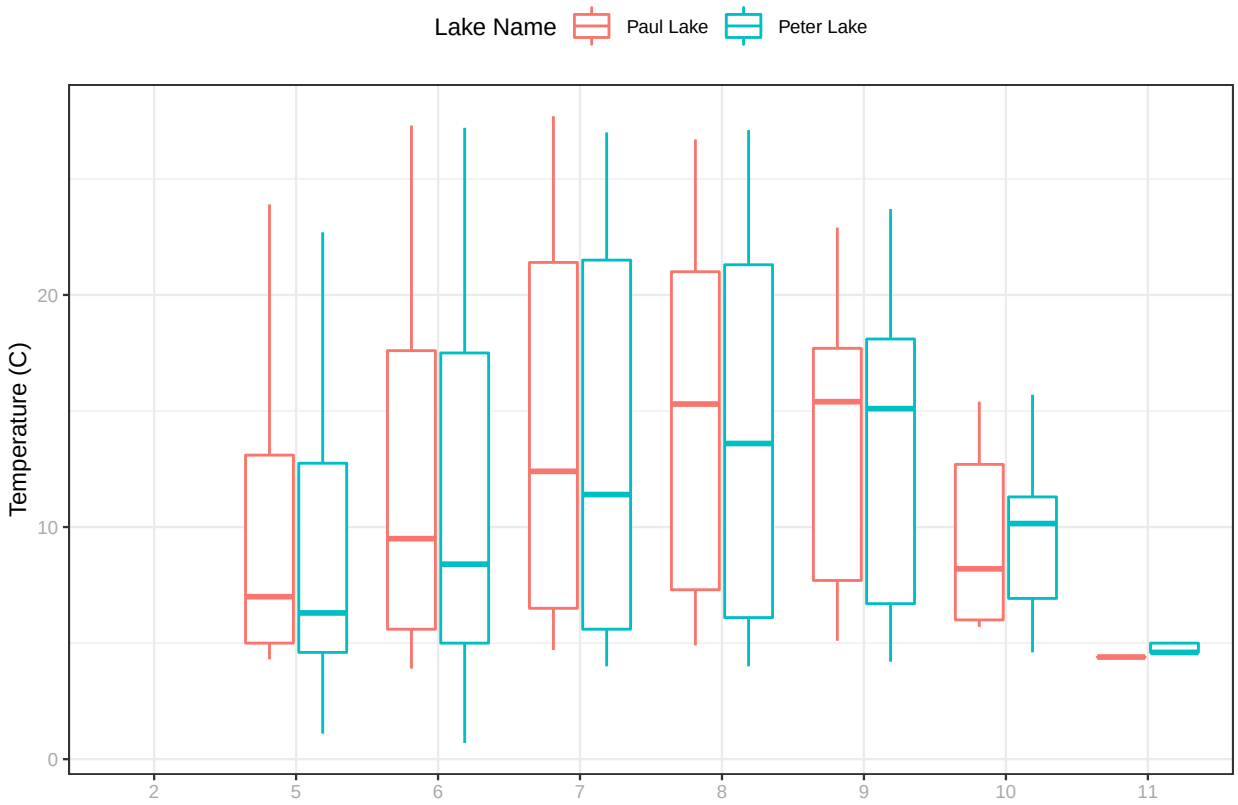
```
## Warning: Removed 1 rows containing missing values (geom_smooth).
```



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.

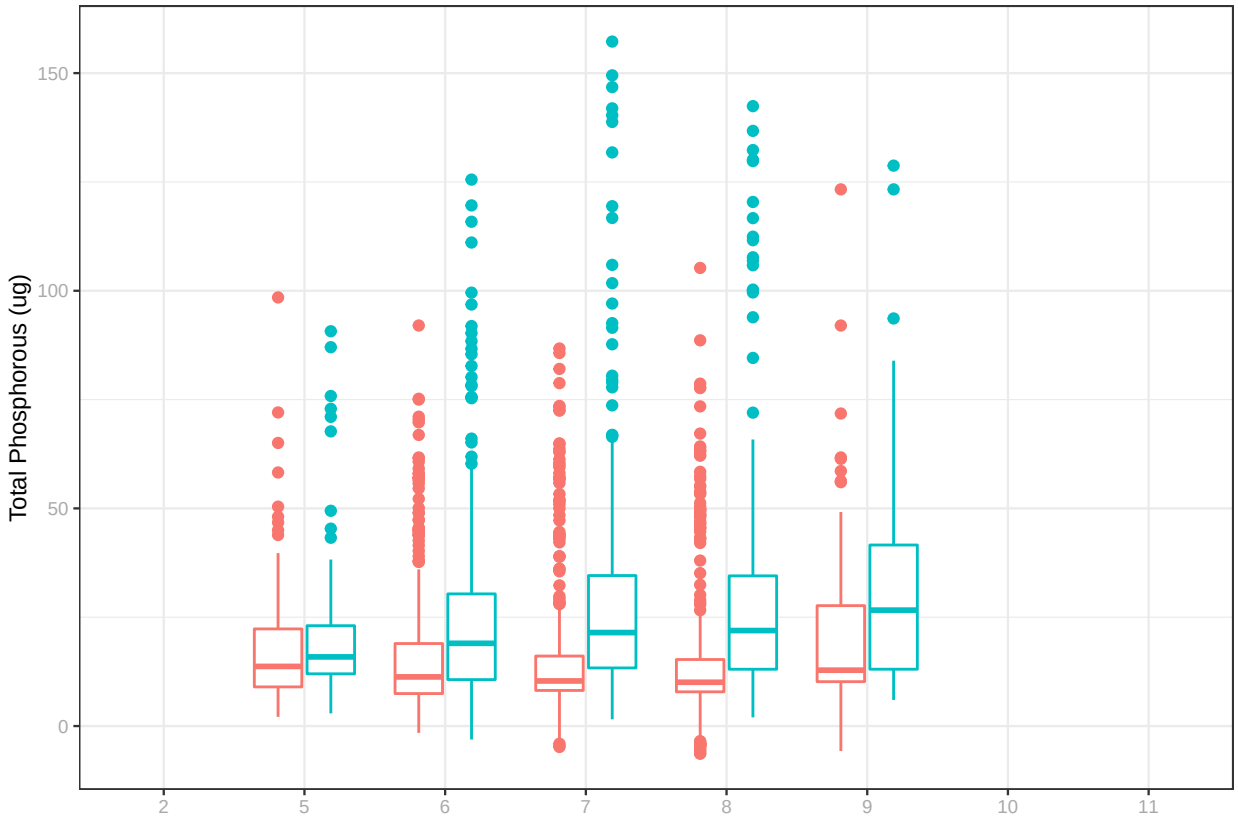
```
Nutr_PeterPaul$month <- as.factor(Nutr_PeterPaul$month) # change date to date format
Nutr_PeterPaul_tidy$month <- as.factor(Nutr_PeterPaul_tidy$month) # change date to date format

TempMonth_plot <- ggplot(Nutr_PeterPaul, aes(x=month, y=temperature_C, color=lakename))+
  geom_boxplot()+
  theme(legend.position = "top")+
  labs(color = "Lake Name", x = "", y = "Temperature (C)")
print(TempMonth_plot)
```

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



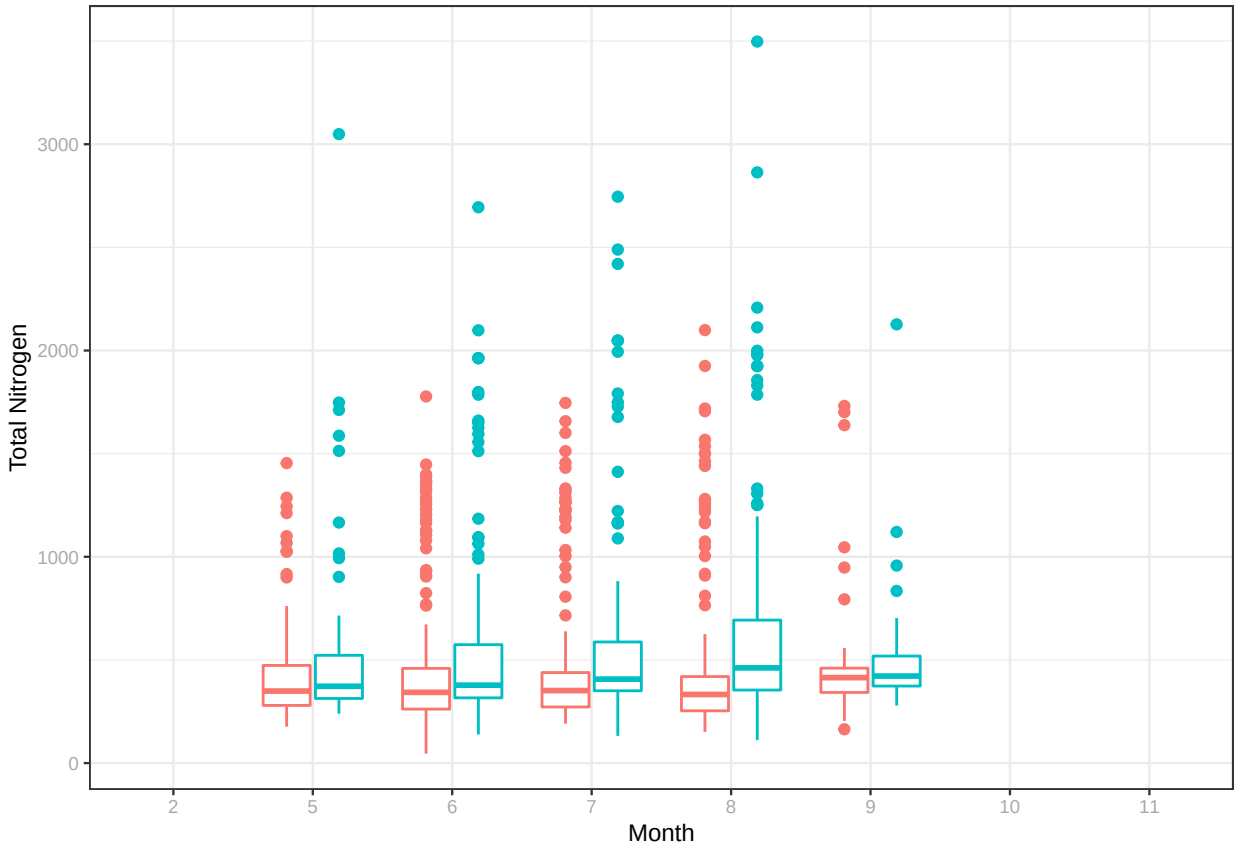

```
TPmonth_plot <- ggplot(Nutr_PeterPaul, aes(x=month, y=tp_ug, color = lakename))+
  geom_boxplot()+
  theme(legend.position = "none")+
  labs(x="", y = "Total Phosphorous (ug)")
print(TPmonth_plot)
```

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




```
TNmonth_plot <- ggplot(Nutr_PeterPaul, aes(x=month, y=tn_ug, color = lakename))+
  geom_boxplot()+
  theme(legend.position = "none")+
  labs(x="Month", y = "Total Nitrogen")
print(TNmonth_plot)
```

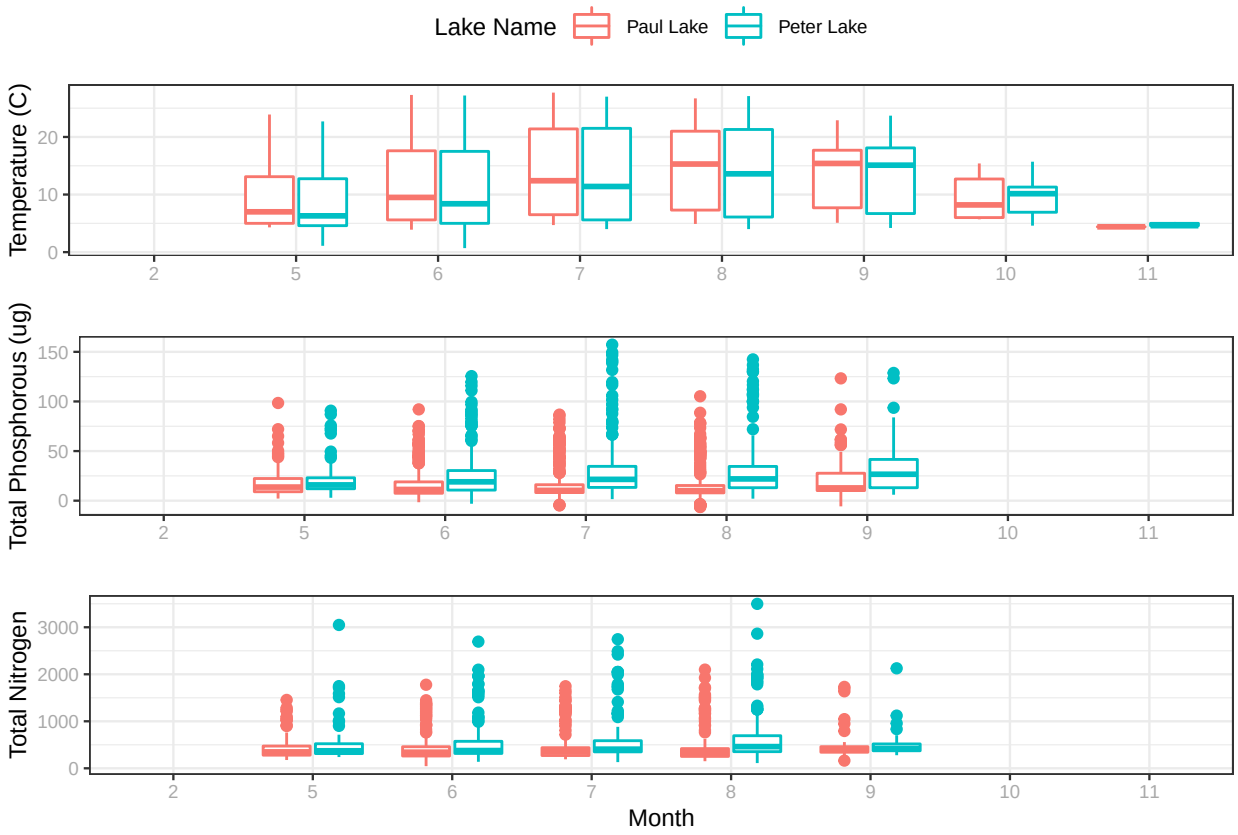
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```




```
Temp_TP_TN_month <- plot_grid(TempMonth_plot, TPmonth_plot,
                              TNmonth_plot, nrow=3, rel_heights = c(4,3,3))
```

```
## 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(Temp_TP_TN_month)
```



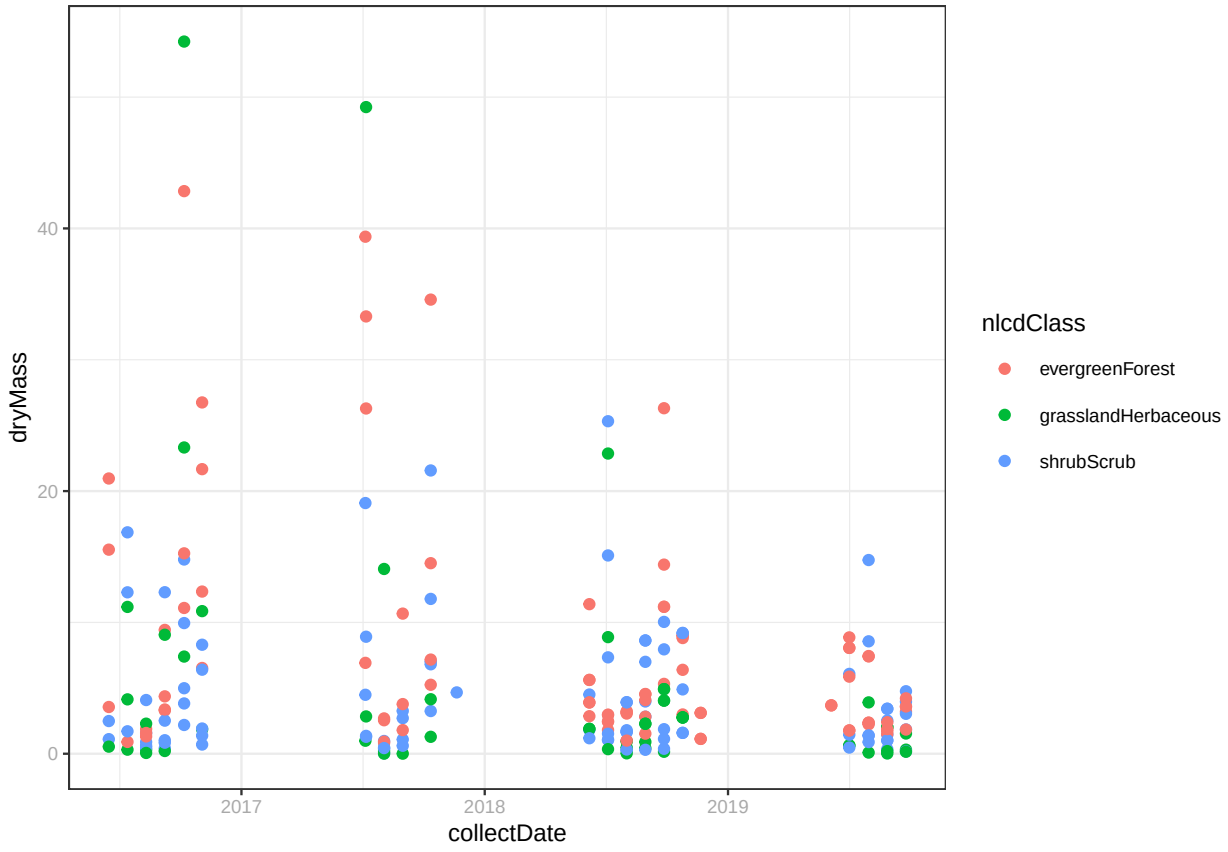
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Both lakes' temperature ranges stayed fairly contained with no outliers for any month, gradually increasing for the summer and decreasing respectively into the fall. The two lakes measured similar temperatures to each other across months. August shows reliably high measurements across all three categories: temperature, phosphorous, and nitrogen compared to other months.For Phosphorous and Nitrogen, both lakes showed outliers, yet Peter Lake's outliers gave much wider distributions. Peter Lake's measurements (medians, interquartile ranges) were consistently higher than Paul Lake's for Phosphorous and Nitrogen.

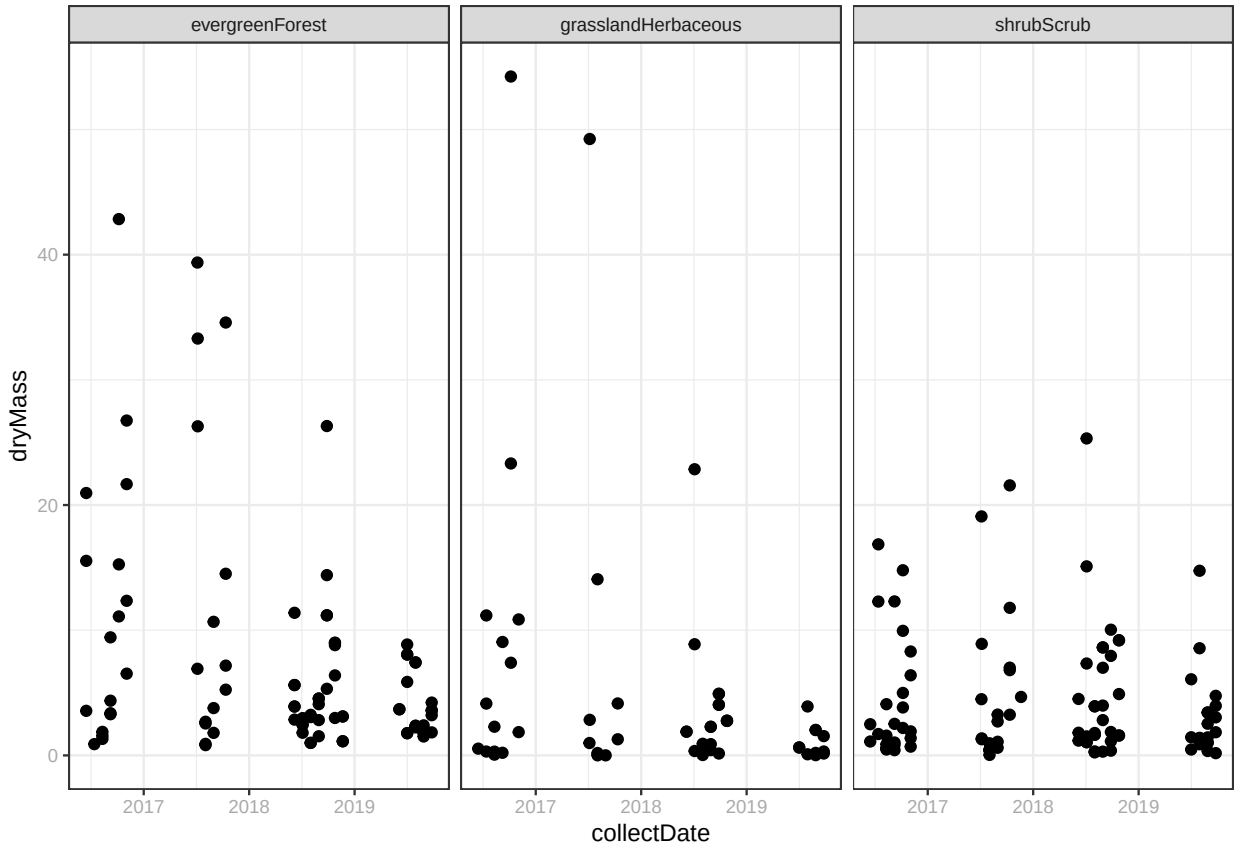
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#import litter dataset and change date format
Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
Litter$collectDate <- as.Date(Litter$collectDate, format = "%Y-%m-%d")

# Plot dry mass of needle litter by date, with NLCDs separated by colors
Needles <- ggplot(subset(Litter, functionalGroup == "Needles"),
                  aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point()
print(Needles)
```




```
# Plot dry mass of needle litter by date, with NLCDs separated into 3 facets
Needles_NLCD <- ggplot(subset(Litter, functionalGroup == "Needles"),
                       aes(x=collectDate, y=dryMass))+
  facet_wrap(vars(nlcdClass))+
  geom_point()
print(Needles_NLCD)
```



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

> Answer: I believe plot 7 is more effective, since it enables an easier comparison of needle mass changes over time, and of needle mass changes between land cover type. Plot 6 is a bit harder to read since both of those comparisons are combined into one visualization. Plot 6 emphasizes more the gaps in data collection months, which is not the intended purpose of the display.