Assignment 5: Data Visualization

Kathleen Mason

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/kathleenmason/Documents/DUKE/Data Analytics/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

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

## 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()
```

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

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

```
Nutrient.peterpaul<- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv

ChemandNutrients.peterpaul<- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Proc

Litter<- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
#2
class(Litter$collectDate) #it is a factor
```

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

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

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

```
class(Nutrient.peterpaul$sampledate) #it is a factor
```

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

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

class(ChemandNutrients.peterpaul$sampledate) #factor
```

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

```
ChemandNutrients.peterpaul$sampledate<-as.Date(ChemandNutrients.peterpaul$sampledate, format= "%Y-%m-%d

#Now all confirmed as dates
```

Define your theme

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

```
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
```

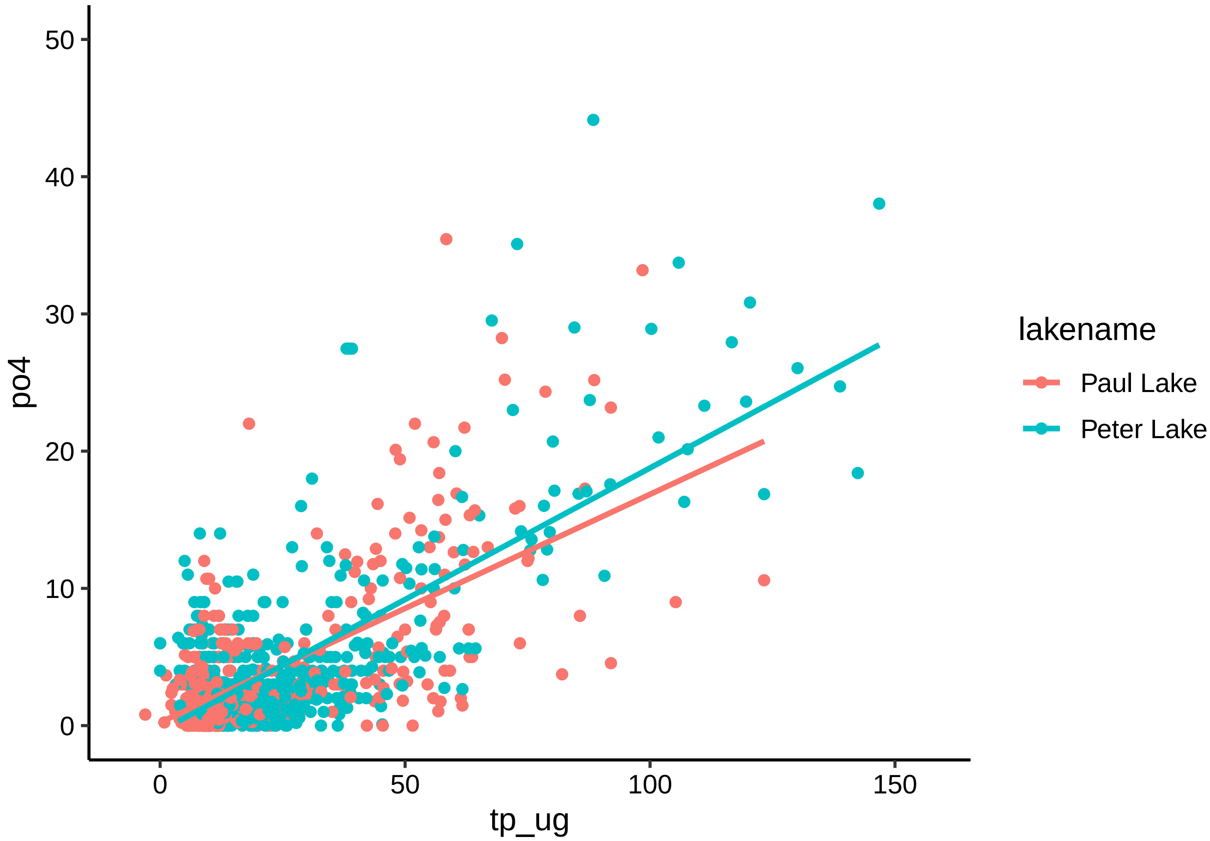
Create graphs

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.

```
totalPbyPhos<-
  ggplot(ChemandNutrients.peterpaul,
         aes(x= tp_ug, y= po4, color= lakename)) +
  geom_point() +
  geom_smooth(method = lm, se=FALSE) +
  ylim(0,50 )+
  mytheme

print(totalPbyPhos)
```

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

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

## Warning: Removed 4 rows containing missing values (geom_smooth).
```



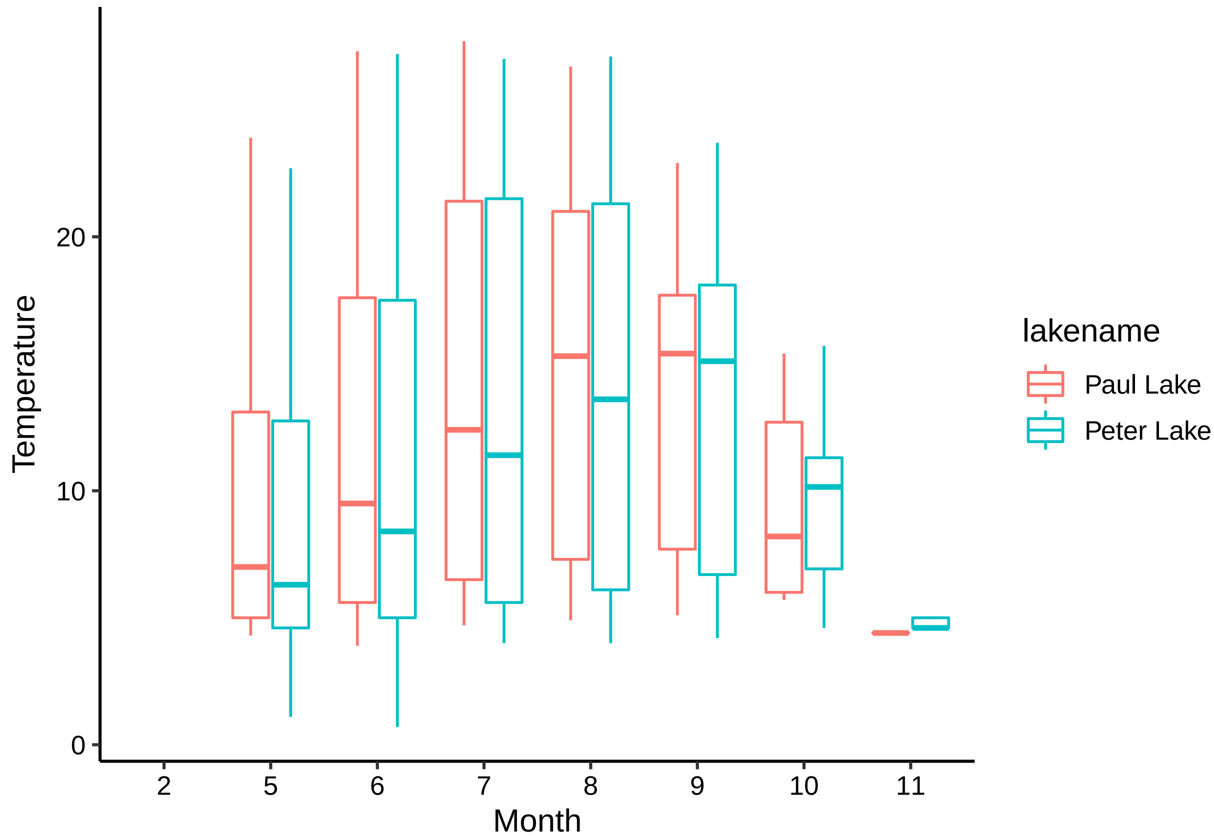
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
class(ChemandNutrients.peterpaul$month)
```

```
## [1] "integer"
```

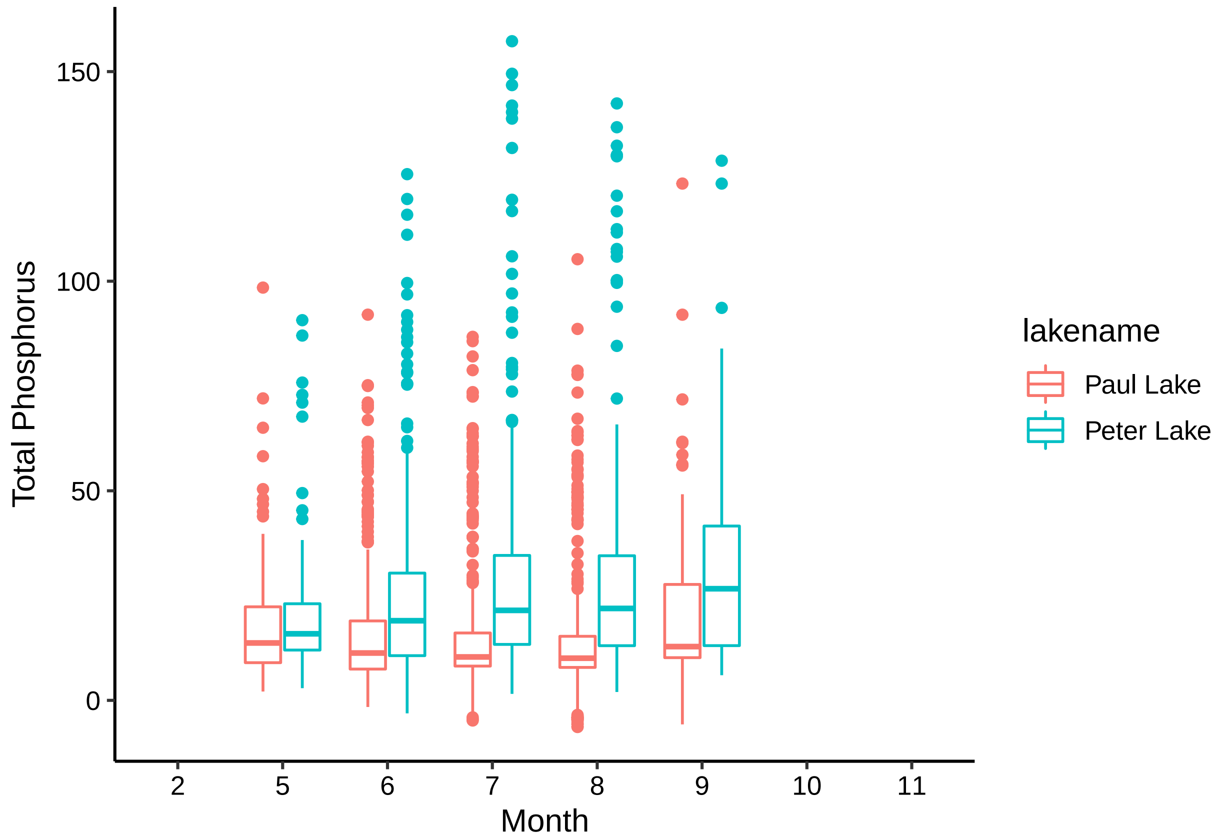
```
ChemandNutrients.peterpaul$month <- factor(ChemandNutrients.peterpaul$month)

TempbyMonth<-
  ggplot(ChemandNutrients.peterpaul, aes(month,temperature_C))+
  geom_boxplot(aes(color=lakename))+
  labs(x=expression("Month"), y=expression("Temperature"))+
  mytheme
print(TempbyMonth)
```

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



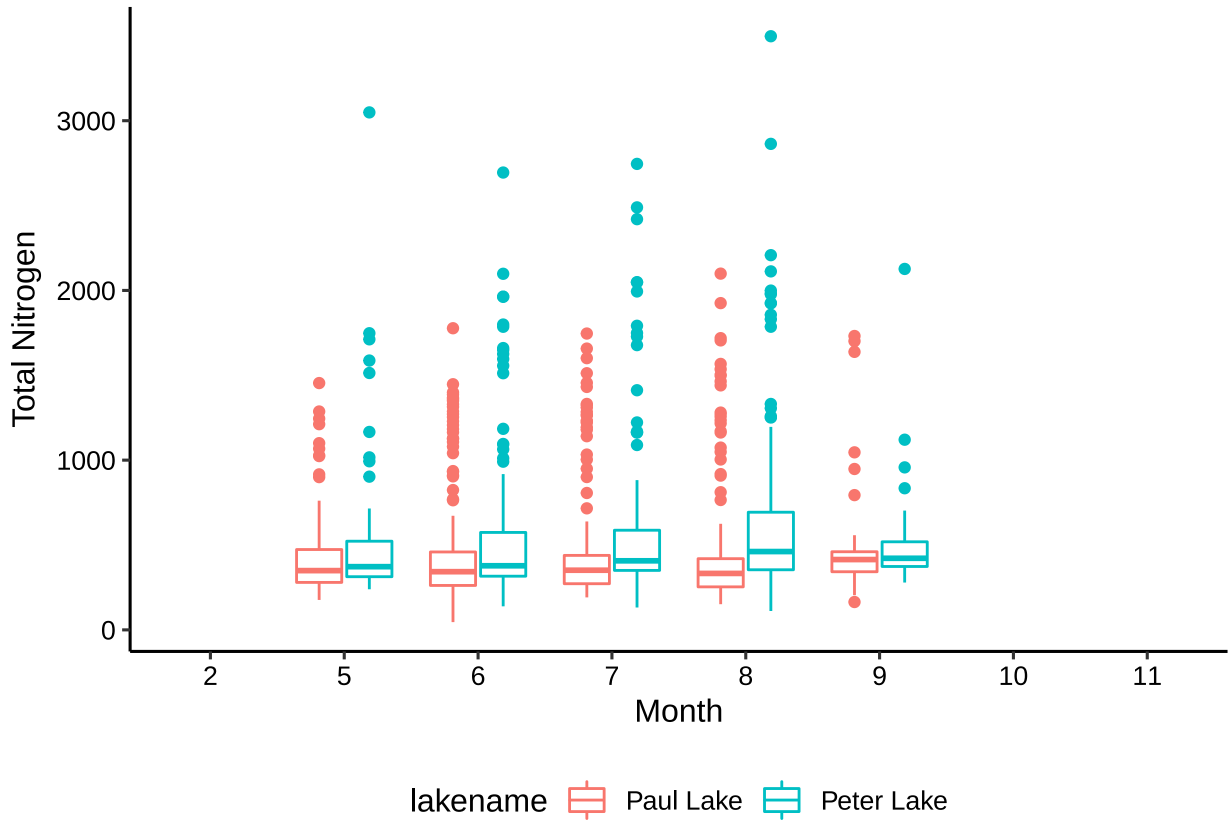

```r
TPbyMonth<-
  ggplot(ChemandNutrients.peterpaul,
         aes(x= month, y= tp_ug))+
  geom_boxplot(aes(color=lakename))+
  labs(x=expression("Month"), y=expression("Total Phosphorus"))+
  mytheme
print(TPbyMonth)
```

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




```
TNbyMonth<-
  ggplot(ChemandNutrients.peterpaul,
         aes(x= month, y= tn_ug))+
  geom_boxplot(aes(color=lakename))+
  labs(x=expression("Month"), y=expression("Total Nitrogen"))+ mytheme +theme(legend.position="bottom")
print(TNbyMonth)
```

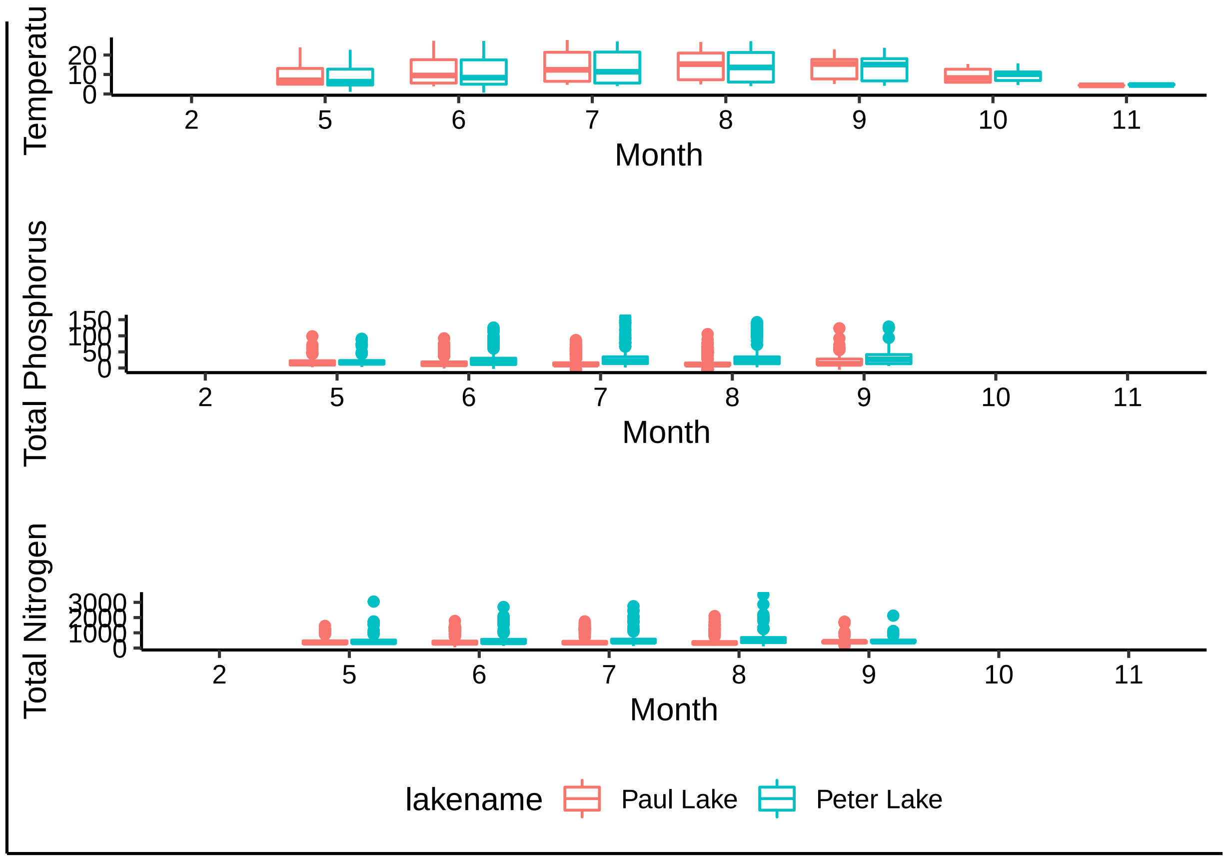
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```




```
#Cowplot
library(cowplot)
MonthbyX<-plot_grid(TempbyMonth+theme(legend.position="none"),
                    TPbyMonth+ theme(legend.position="none"),
                    TNbyMonth,
                    nrow = 3, align = 'h', axis = "bt")+ mytheme
```

```
## 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(MonthbyX)
```




```
#+ theme(legend.position="none") omits a legend, so I omitted two of the graphs legends and kept one
#axis ="bt" aligns the bottom and top of the graphs
```

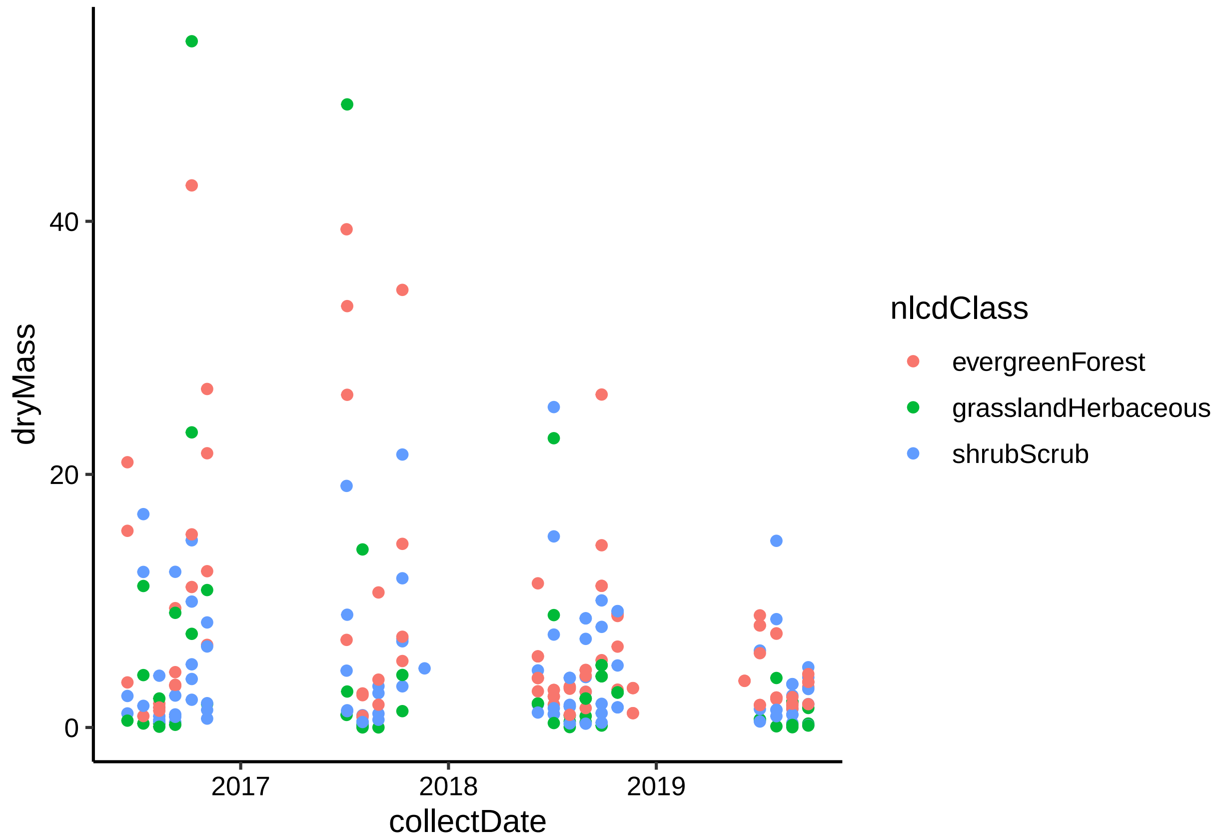
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The two lakes have relatively similar observations among the parameters of temp, TP, and TN. Temperature increases in the summer months, and N and P have more peak concentrations in these same months. P and N are more variable when warmer

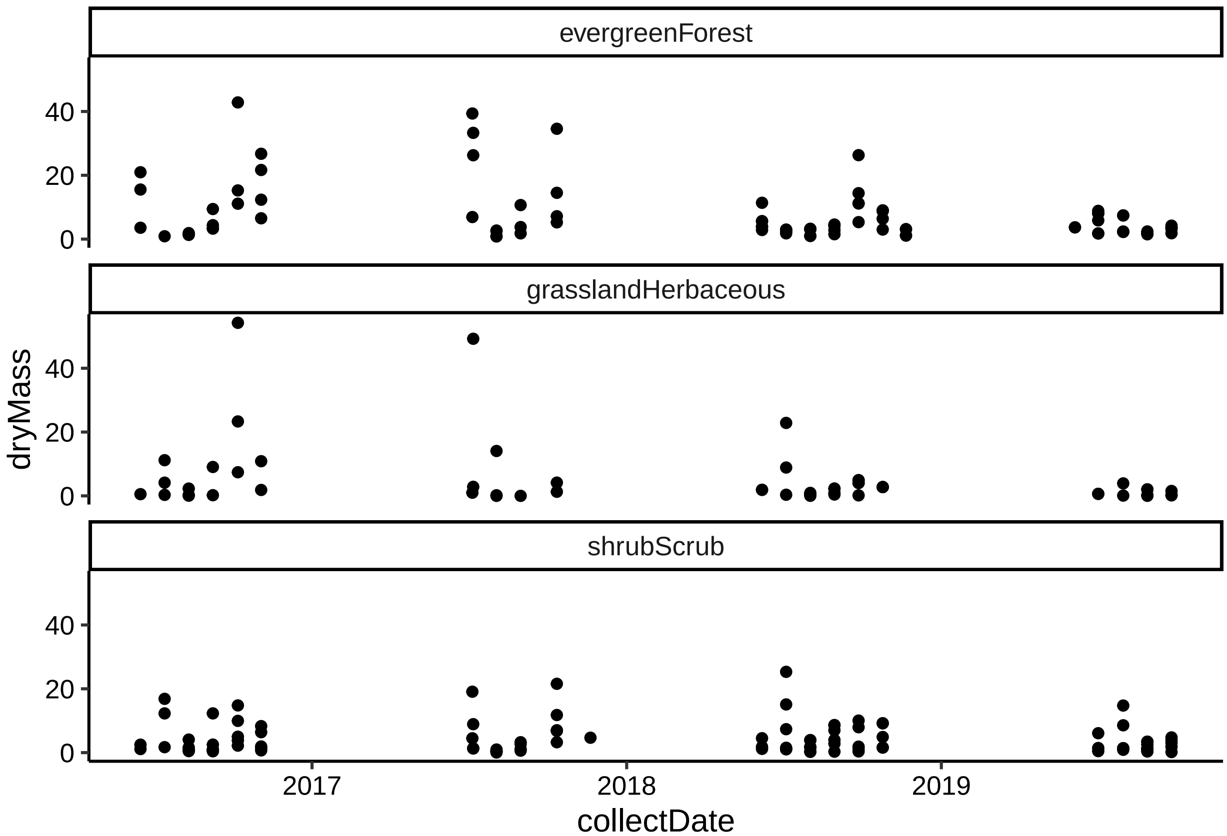
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
Needles.drymass.time<-
  ggplot(subset(Litter, functionalGroup == "Needles" ),
         aes(x=collectDate, y=dryMass))+
  geom_point(aes(color=nlcdClass))+ mytheme+theme(legend.position = "right")
print(Needles.drymass.time)
```




```
#7
Needles.drymass.time<-
  ggplot(subset(Litter, functionalGroup == "Needles" ), aes(x=collectDate, y=dryMass))+
  geom_point()+ facet_wrap(vars(nlcdClass), nrow=3) + mytheme

print(Needles.drymass.time)
```



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

> Answer: I think 7 is a more effective way to see the changes in dry mass over time for each NLCD class, while still comparing them against each other. In figure 6, it is hard to see a specific trend, I think they get lost together.