Assignment 5: Data Visualization

Logan Loadholtz

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()
library(tidyverse)
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

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

NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemis

NTL_LTER_Lake_Nutrients_PeterPaulGathered_Processed <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrien

NEON_NIWO_Litter_mass_trap_Processed <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed
```

```
#2
class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
```

```
## [1] "character"
```

```
class(NTL_LTER_Lake_Nutrients_PeterPaulGathered_Processed$sampledate)
```

```
## [1] "character"
```

```
class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

```
## [1] "character"
```

```
#All three datasets are NOT reading the dates as "date" instead as "character",
#will need to convert below.


NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate <- as.Date(NTL_LTER_Lake_Chemistry_Nutr
                                                                            format= "%Y-%m-%d")

class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
```

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

```
NTL_LTER_Lake_Nutrients_PeterPaulGathered_Processed$sampledate <- as.Date(NTL_LTER_Lake_Nutrients_Peterl
                                                                          format = "%Y-%m-%d")

class(NTL_LTER_Lake_Nutrients_PeterPaulGathered_Processed$sampledate)
```

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

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

class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

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

Define your theme

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

```
MyDefaultTheme <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "blue"),
        legend.position = "top")

theme_set(MyDefaultTheme)
```

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.

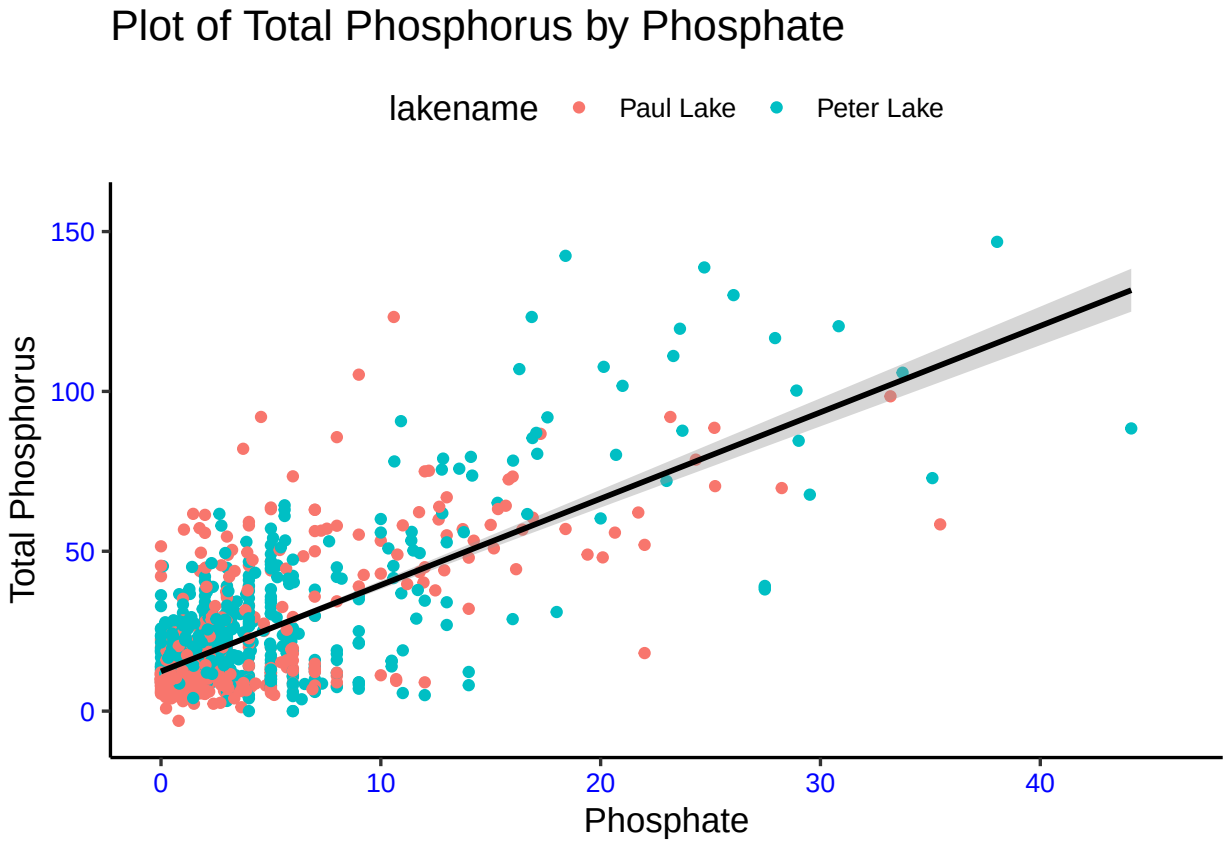
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values.

```
Plot1_Phos <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
                     aes(x= po4, y=tp_ug,color=lakename )) +
  geom_point() +
  geom_smooth(method = lm, color="black") +
  xlim(0,46) +
  labs(title="Plot of Total Phosphorus by Phosphate",
  x="Phosphate",
  y="Total Phosphorus")


print(Plot1_Phos)
```

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

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

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



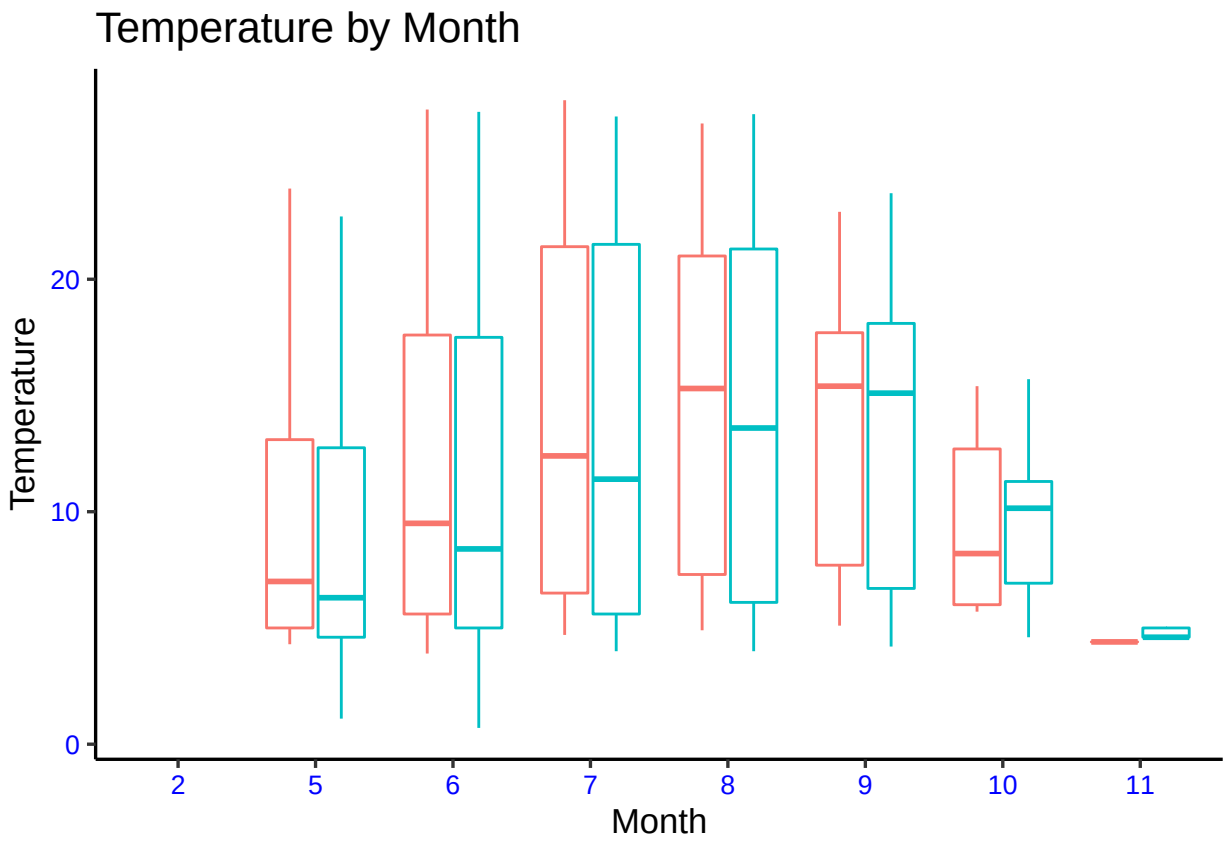
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$month <- as.factor(NTL_LTER_Lake_Chemistry_Nutrien
#Here I changed month to factor so it would show up as separate boxplots

class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$month)
```

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

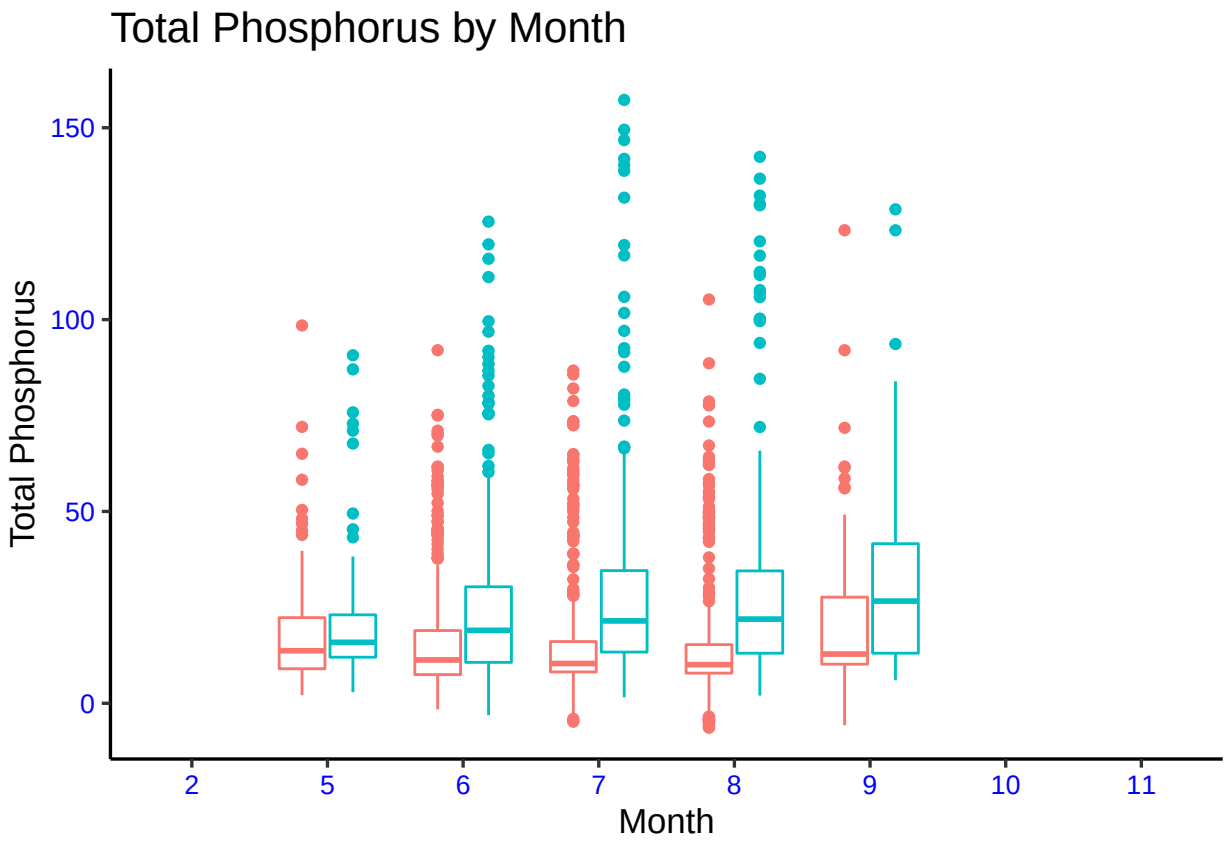
```
Boxplot1_temp <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
                        aes(x= month, y=temperature_C, color=lakename)) +
  geom_boxplot() +
  labs(title="Temperature by Month",
  x="Month",
  y="Temperature")+
  theme(legend.position = "none")

print(Boxplot1_temp)
```

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



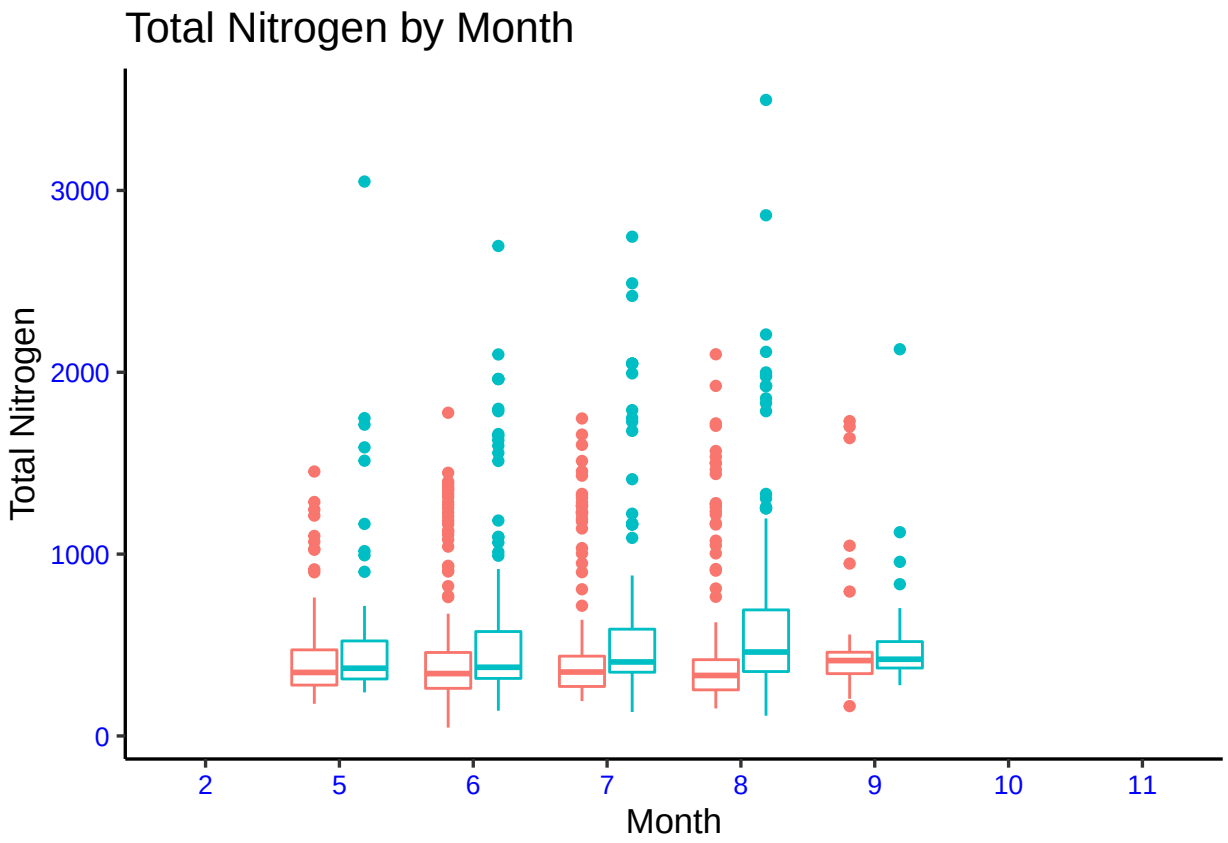

```
Boxplot2_TP <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
                      aes(x= month, y=tp_ug, color=lakename)) +
  geom_boxplot() +
  labs(title="Total Phosphorus by Month",
  x="Month",
  y="Total Phosphorus")+
  theme(legend.position = "none")

print(Boxplot2_TP)
```

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




```
Boxplot3_TN <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
                      aes(x= month, y=tn_ug, color=lakename)) +
  geom_boxplot() +
  labs(title="Total Nitrogen by Month",
  x="Month",
  y="Total Nitrogen")+
  theme(legend.position = "none" )

print(Boxplot3_TN)
```

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



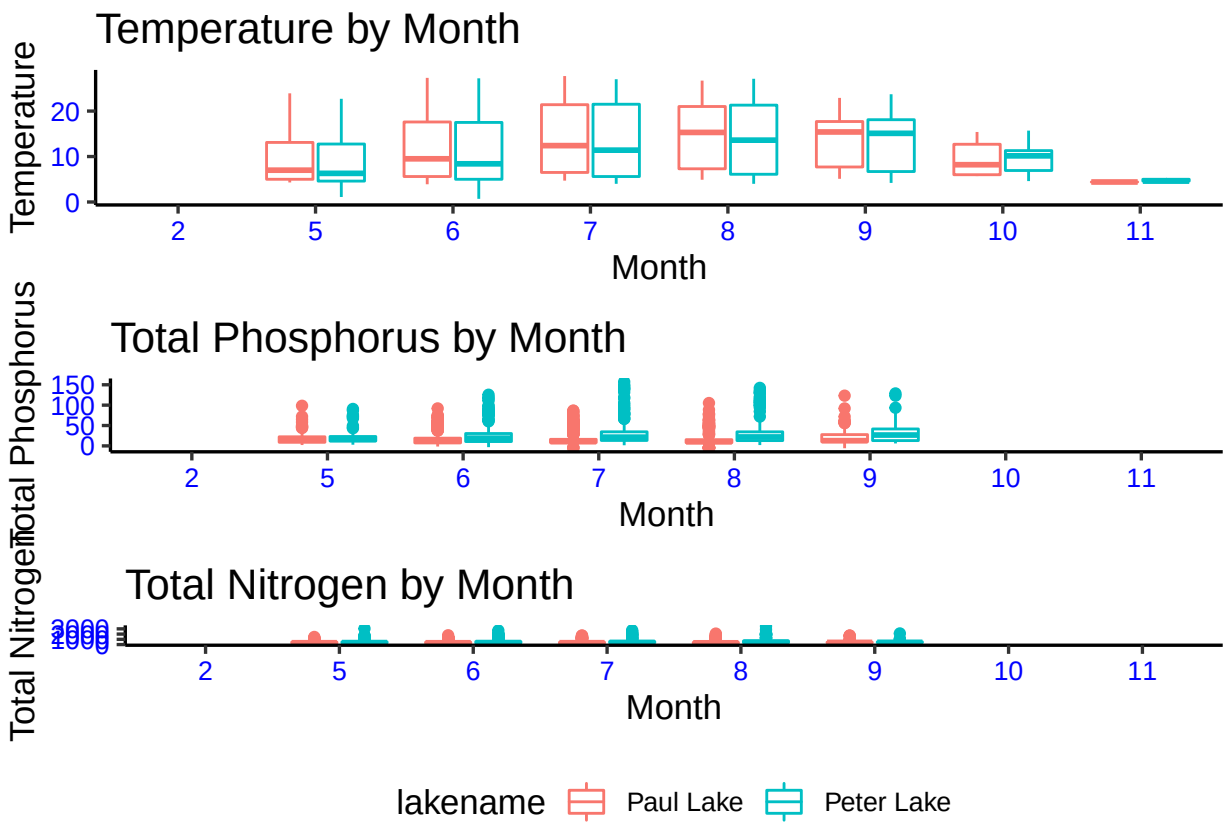

```
Boxplot3_TN_withlegend <- ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed,
                                 aes(x= month, y=tn_ug, color=lakename)) +
  geom_boxplot() +
  labs(title="Total Nitrogen by Month",
  x="Month",
  y="Total Nitrogen")+
  theme(legend.position = "bottom" )

legend_Cowplot <- get_legend(Boxplot3_TN_withlegend)
```

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

```
CowPlot <- plot_grid(Boxplot1_temp, Boxplot2_TP, Boxplot3_TN_withlegend,
                     nrow = 3, align = 'h', rel_heights = c(1.25, 1))
```

```
## 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).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
print(CowPlot)
```



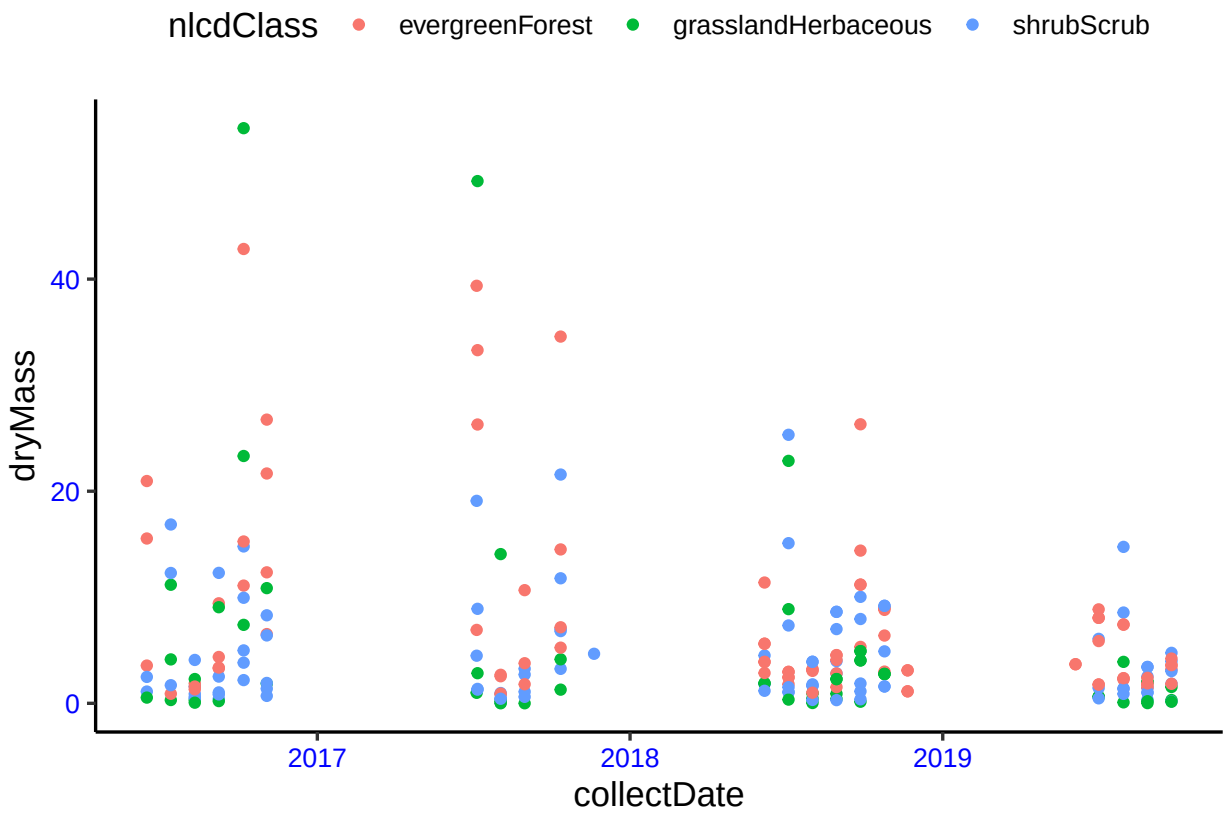
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer:

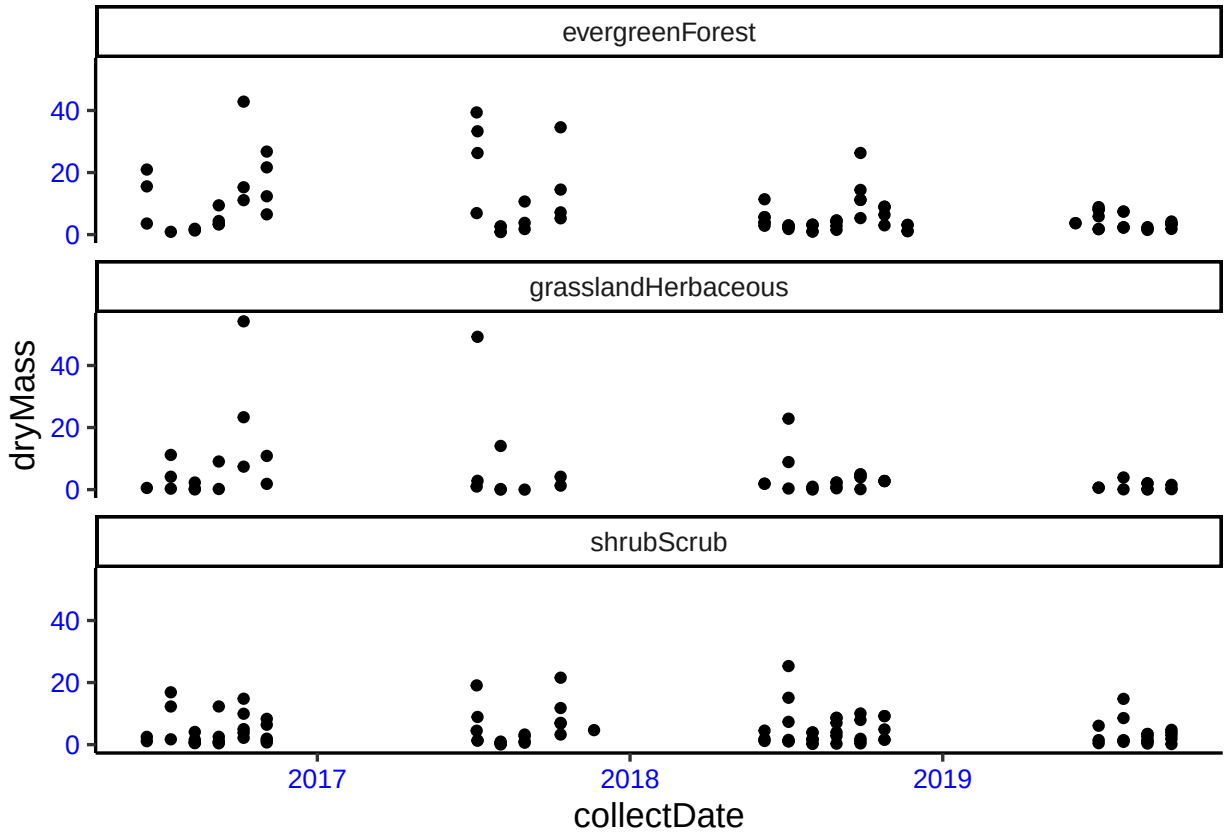
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
Plot6_NiwotRidge <- ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed,
                                  functionalGroup == "Needles")) +
  geom_point(aes(x=collectDate, y=dryMass, color=nlcdClass))

print(Plot6_NiwotRidge)
```




```
Plot7_NiwotRidge <- ggplot(subset(NEON_NIWO_Litter_mass_trap_Processed,
                                  functionalGroup == "Needles")) +
  geom_point(aes(x=collectDate, y=dryMass)) +
  facet_wrap(vars(nlcdClass), nrow=3)

print(Plot7_NiwotRidge)
```



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

Answer: I think plot 7 is more effective becuase it displays the data more clearly in a way that is easier for the reader to comprehend. Plot 6 combines all three classes, and even though they are separated by color, it's still a little difficult to really see what is being displayed, especially if you want to analyze one class. Plot 7 is much more easy to read because the classes are separated, but even though they are separated, the years line up at the bottom for all 3 so you can still compare each one over time. Plot 7 allows the reader to easily analyze each class over time.