Assignment 5: Data Visualization

Andrew Brantley

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 Monday, February 14 at 7:00 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 (use the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed.csv`] version).

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

```
#1

getwd()
```

```
## [1] "/Users/AndrewBrantley/Library/CloudStorage/Box-Box/Environmental Data Analytics/GithubRepos/Envi
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

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

PeterPaul.Processed <-
```

```
  read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")

Litter.Processed <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2

#correcting Date format for each date column in both datasets

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

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

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

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

```
# forcing month to be factor not number to have better plotting in #4/#5

PeterPaul.Processed$month <- as.factor(PeterPaul.Processed$month)
```

Define your theme

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

```
#3

# Building theme off dark theme
Andrew.Theme <- theme_dark(base_size = 14) +
  theme(axis.text = element_text(colour = "black", face = "italic"),
        legend.position = "right",
        panel.grid.major.x = element_line(colour = "black", linetype = 3, size = 0.5),
        panel.grid.major.y =  element_line(colour = "black", linetype = 3, size = 0.5))
```

Create graphs

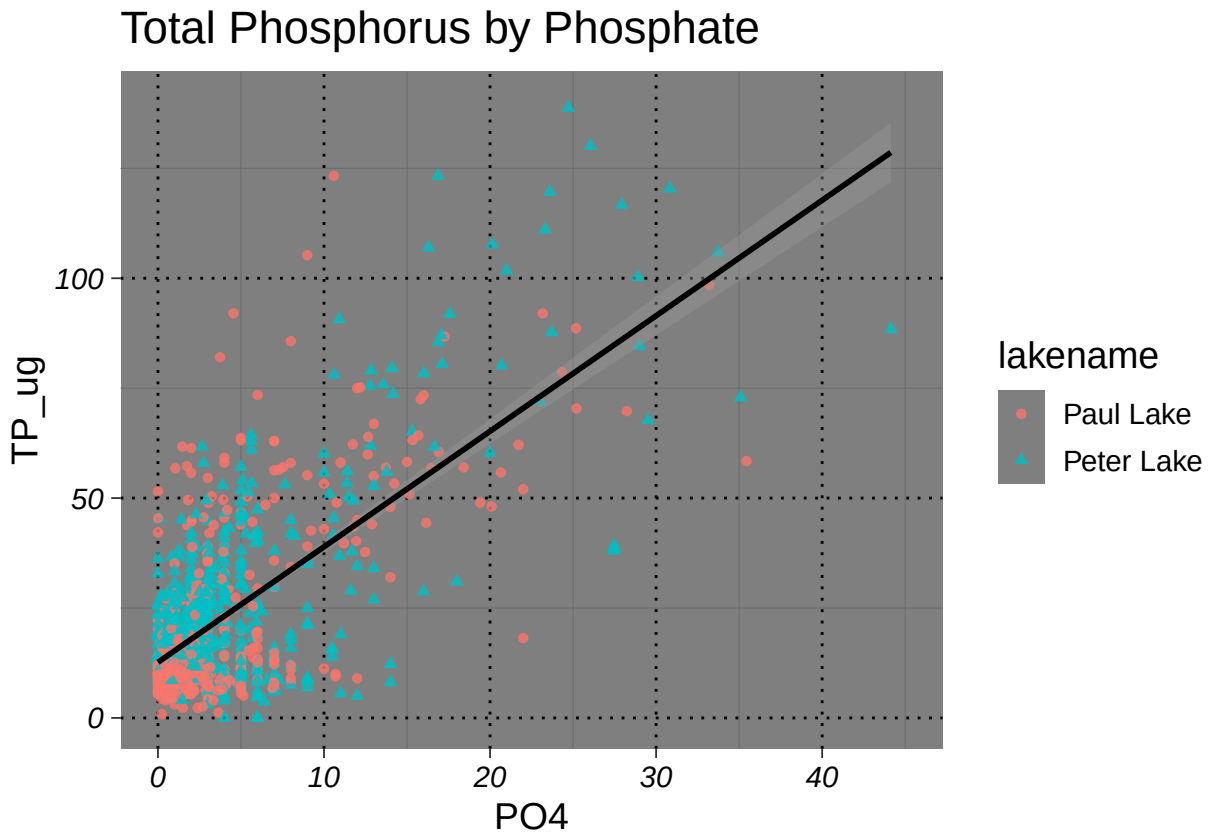
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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 `ylim()`).

```
#4

# Creating Lake total phosphorus by phosphate graphs

Plot1 <- ggplot(PeterPaul.Processed) +
  geom_point(aes(x = po4, y = tp_ug, shape  = lakename, color = lakename),
             alpha = 0.8) + #setting data aesthetics
  xlim(0, 45) +
  ylim(0, 140) +
  ggtitle("Total Phosphorus by Phosphate") + xlab("PO4") + ylab("TP_ug") + #applying names
  geom_smooth(aes(po4, tp_ug), method = "lm", color = "black") + #line of best fit
  Andrew.Theme
```

```
print(Plot1)
```

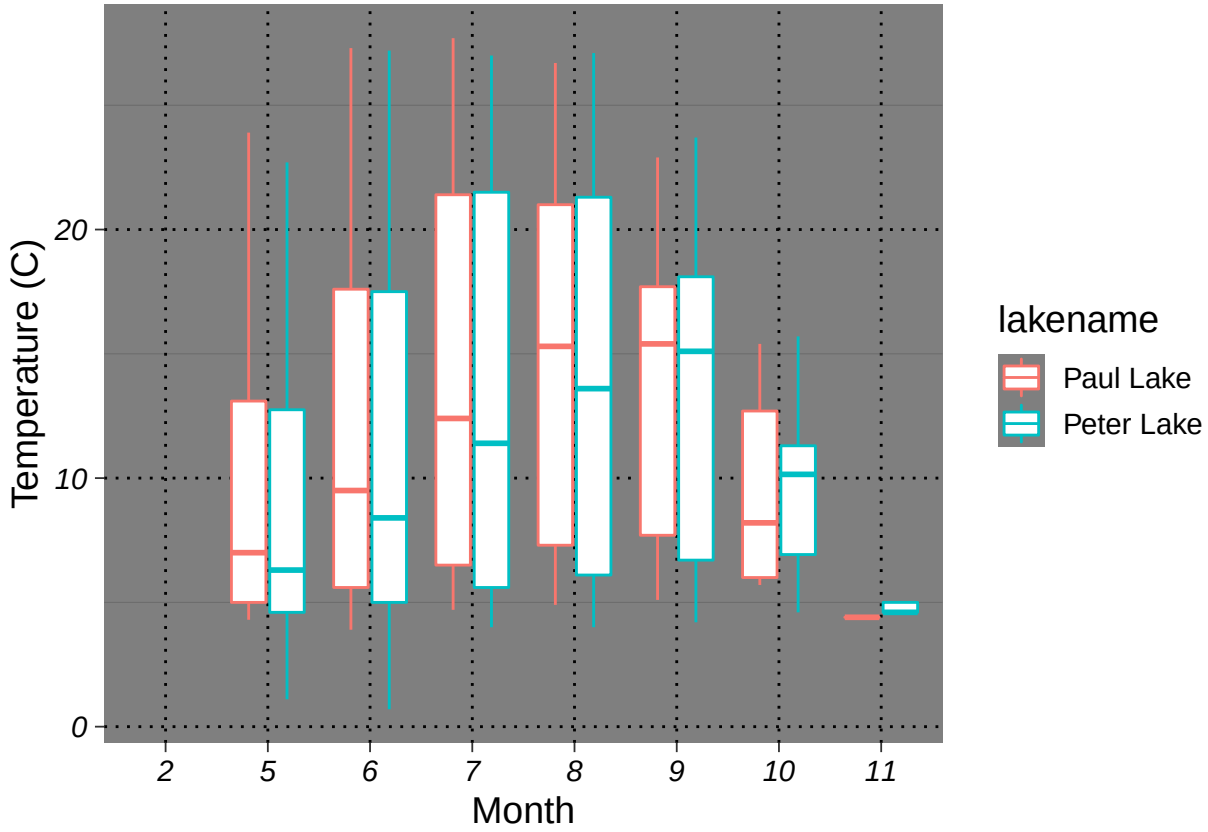
```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 21950 rows containing non-finite values (stat_smooth).
## Warning: Removed 21950 rows containing missing values (geom_point).
```



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.

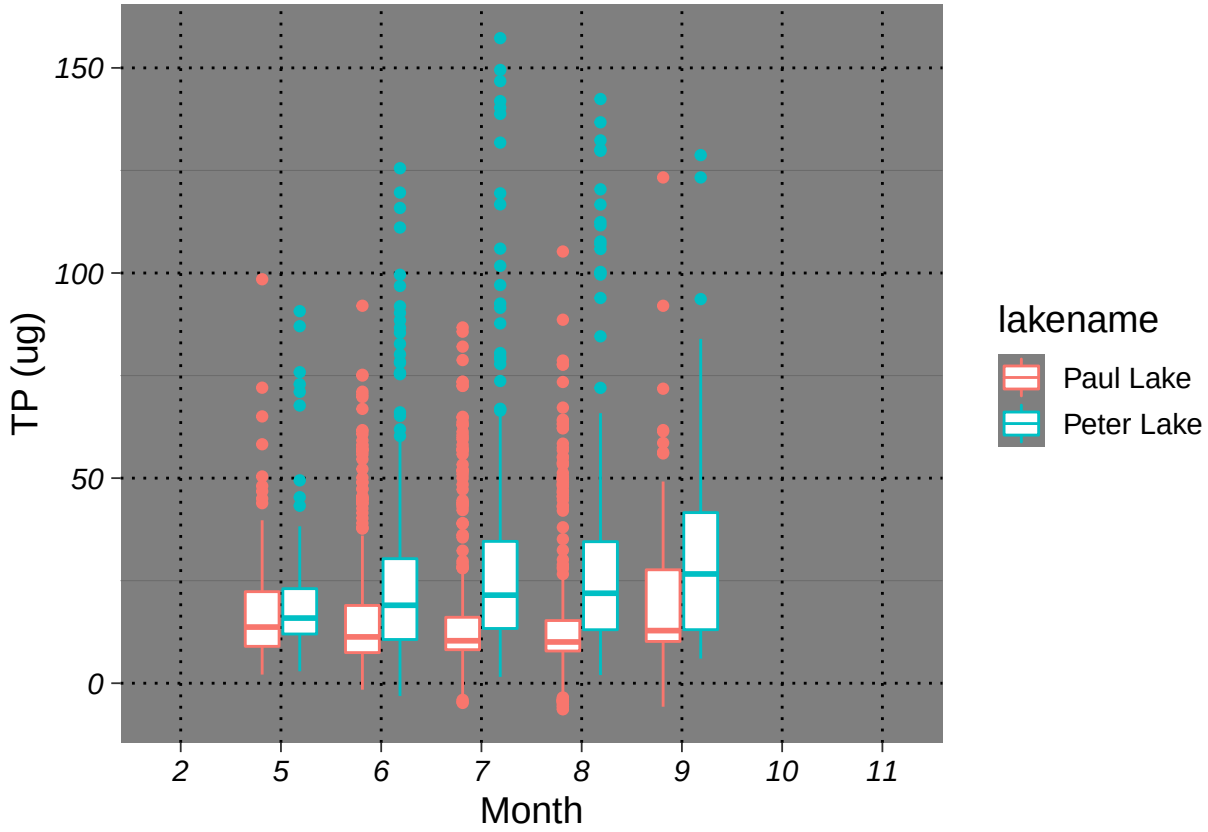
```
#5

# PeterPaul boxplots

TempPlot <- ggplot(PeterPaul.Processed) +
  geom_boxplot(aes(x= month, y = temperature_C, color = lakename)) +
  ylab("Temperature (C)") +
  xlab("Month") +
  Andrew.Theme
print(TempPlot)
```

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



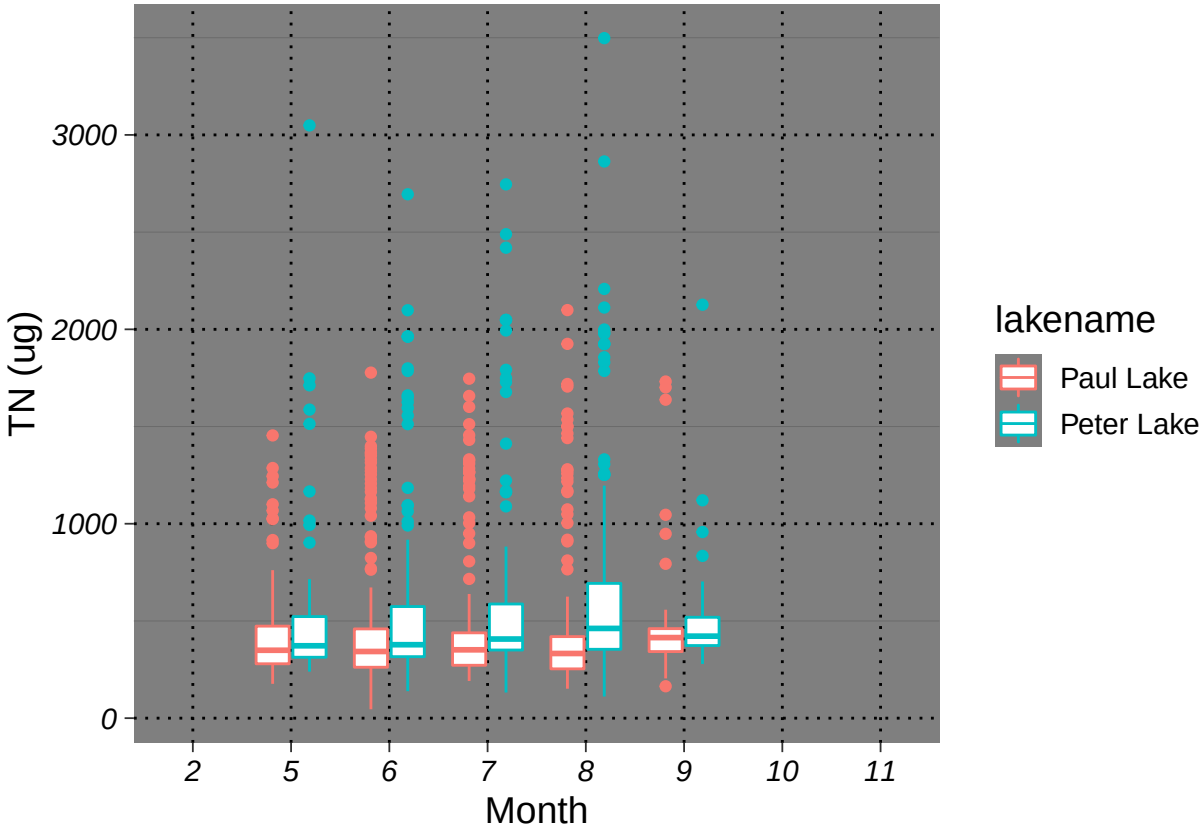

```
TPPlot <- ggplot(PeterPaul.Processed) +
 geom_boxplot(aes(x = month, y = tp_ug, color = lakename)) +
  ylab("TP (ug)") +
  xlab("Month") +
  Andrew.Theme
print(TPPlot)
```

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




```
TNPlot <- ggplot(PeterPaul.Processed) +
 geom_boxplot(aes(x = month, y = tn_ug, color = lakename)) +
  ylab("TN (ug)") +
  xlab("Month") +
  Andrew.Theme
print(TNPlot)
```

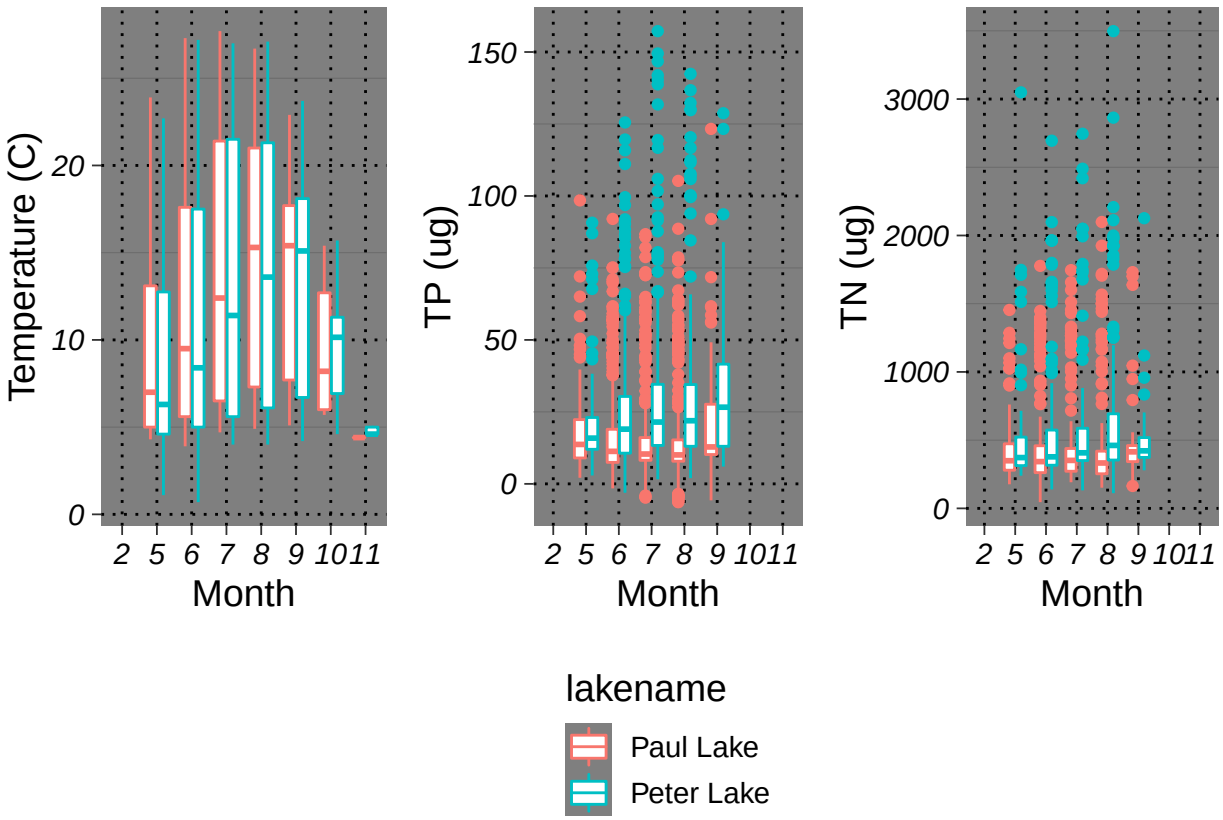
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```




```
# Combining into cowplot

Cowplot <- plot_grid(TempPlot + theme(legend.position = "none"),
                     TPPlot + theme(legend.position = "bottom", legend.direction = "vertical"),
                     TNPlot + theme(legend.position = "none"), ncol = 3, axis = "b", align = "h")
```

```
## 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(Cowplot)
```



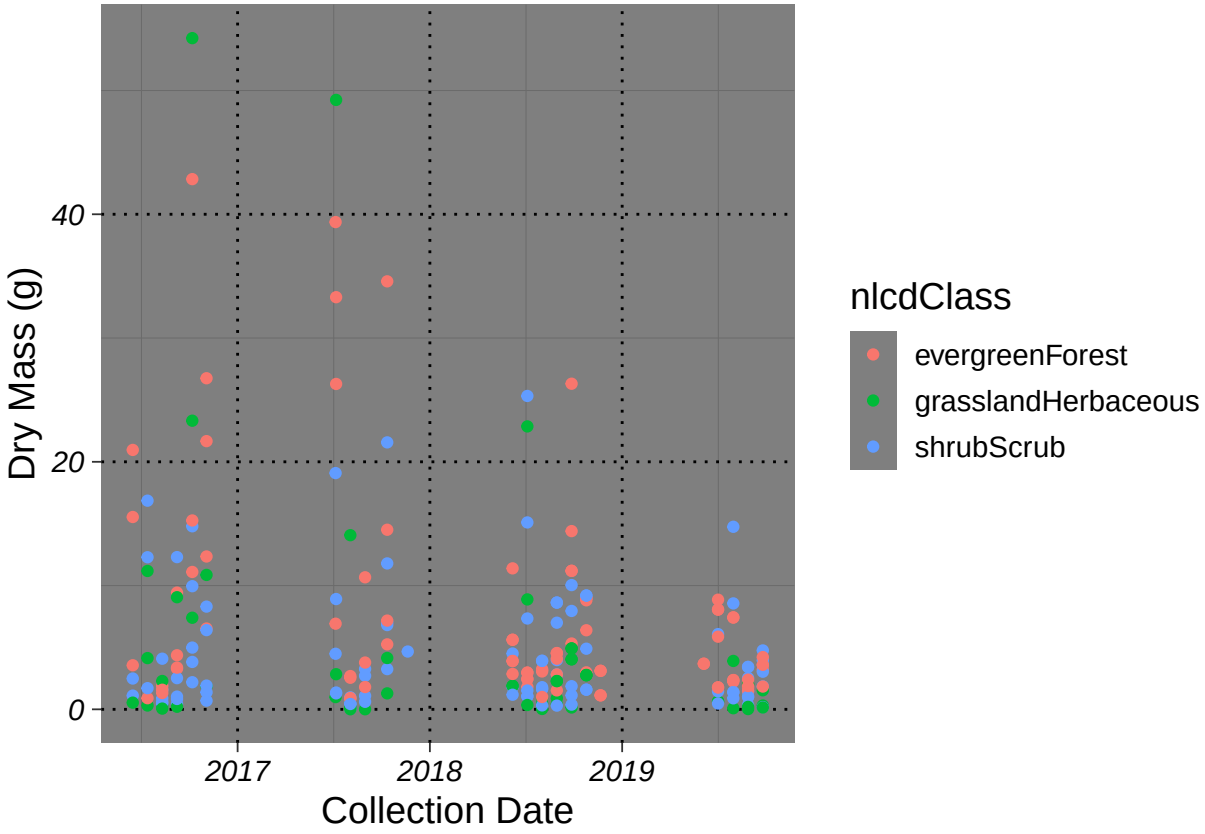
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Overall it seems that there is much larger variation in TP and TN with many outliers in the boxplots. There seems to be a peak in all three variables in the summer months compare to winter months that are generally colder and involve less nutrient flux in the water samples.

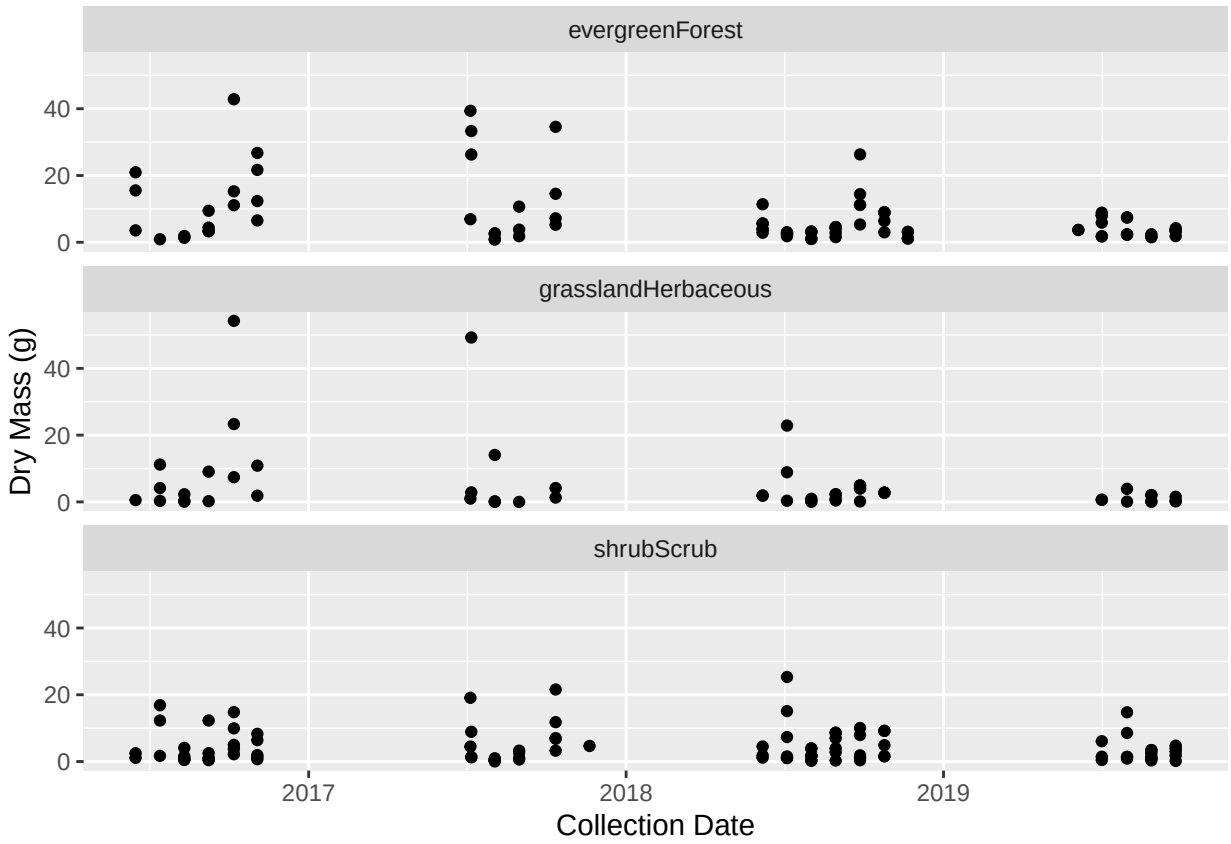
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

# Needles dry mass plotted by nlcd overlapping one another

NeedlesPlot <- ggplot(subset(Litter.Processed, functionalGroup == "Needles")) +
  geom_point(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  ylab("Dry Mass (g)") +
  xlab("Collection Date") +
  Andrew.Theme
print(NeedlesPlot)
```




```
#7

# Needles dry mass faceted by nlcd variable

NeedlesFacetPlot <- ggplot(subset(Litter.Processed, functionalGroup == "Needles")) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  ylab("Dry Mass (g)") +
  xlab("Collection Date") +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(NeedlesFacetPlot)
```



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

Answer: I think the plot for number 7 does a better job of conveying the information that the data is telling us. I think when the NLCD classes are separated it is easier to see trends within each class and within each collection window. It is much easier to see the spread of the different classes in different years when they are not all overlapped like in the graph for number 6.