Assignment 5: Data Visualization

Samantha Pace

Fall 2023

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

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

```
#1

#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("here")
#install.packages("cowplot")
# they are all now comments so it will knit.

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/R/EDE_Fall2023
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
# These code chunks are loading the packages.

getwd() # verifying working directory
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```
setwd("/home/guest/R/EDE_Fall2023") #setting wd to EDE_Fall2023
getwd() # checking wd again
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```
# the two code chunks below are reading in the processed datasets.

Peter.Paul.chem.nutrients <-
  read.csv(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)

Neon.Niwo.litter <-
  read.csv("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
           stringsAsFactors = TRUE)

#2
# fixing dates to be date format
Peter.Paul.chem.nutrients$sampledate <- ymd(Peter.Paul.chem.nutrients$sampledate)
Neon.Niwo.litter$collectDate <- ymd(Neon.Niwo.litter$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```
#3
# creating a theme based on the bw theme; larger text; modified background plot
# and modified title.

mytheme <-
```

```
  theme_classic(base_size = 14)+
    theme(panel.grid.major = element_line(colour = "grey80"))+
  theme(plot.title = element_text(size = rel(1)))

# the following code sets this theme as the default theme.
theme_set(mytheme)
```

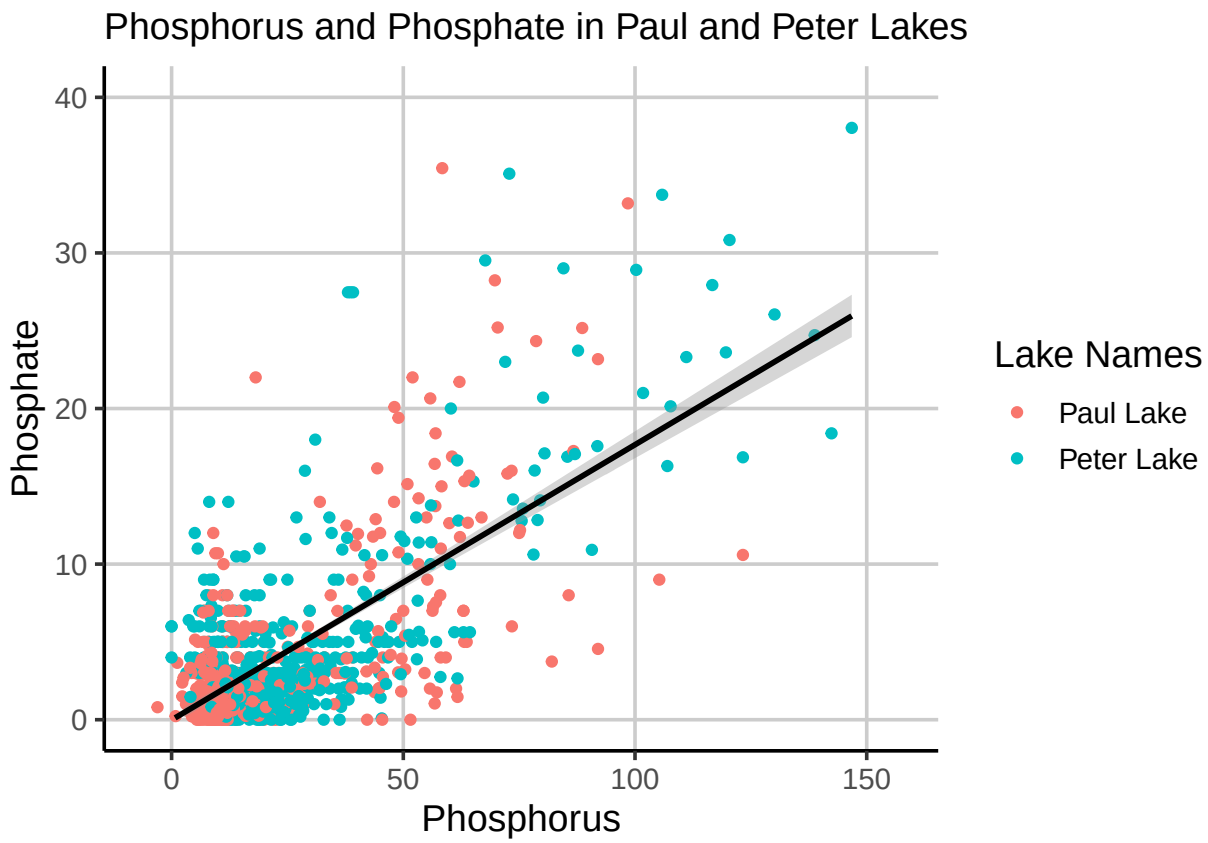
Create graphs

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 (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
# this code chunk makes a scatter plot of
# phosphorus by phosphate with two different colors for each lake.
# Extreme values are hidden and a black line of best fit is in black.

Lakes.phosphorus.phosphate <-
  ggplot(Peter.Paul.chem.nutrients, aes(x = tp_ug, y = po4, color = lakename))+
  geom_point()+
  ylim(0, 40)+
  geom_smooth(method=lm, color = 'black')+
  xlab("Phosphorus")+
  ylab("Phosphate")+
labs(color = "Lake Names")+
ggtitle("Phosphorus and Phosphate in Paul and Peter Lakes")
print(Lakes.phosphorus.phosphate)
```

```
## `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 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.

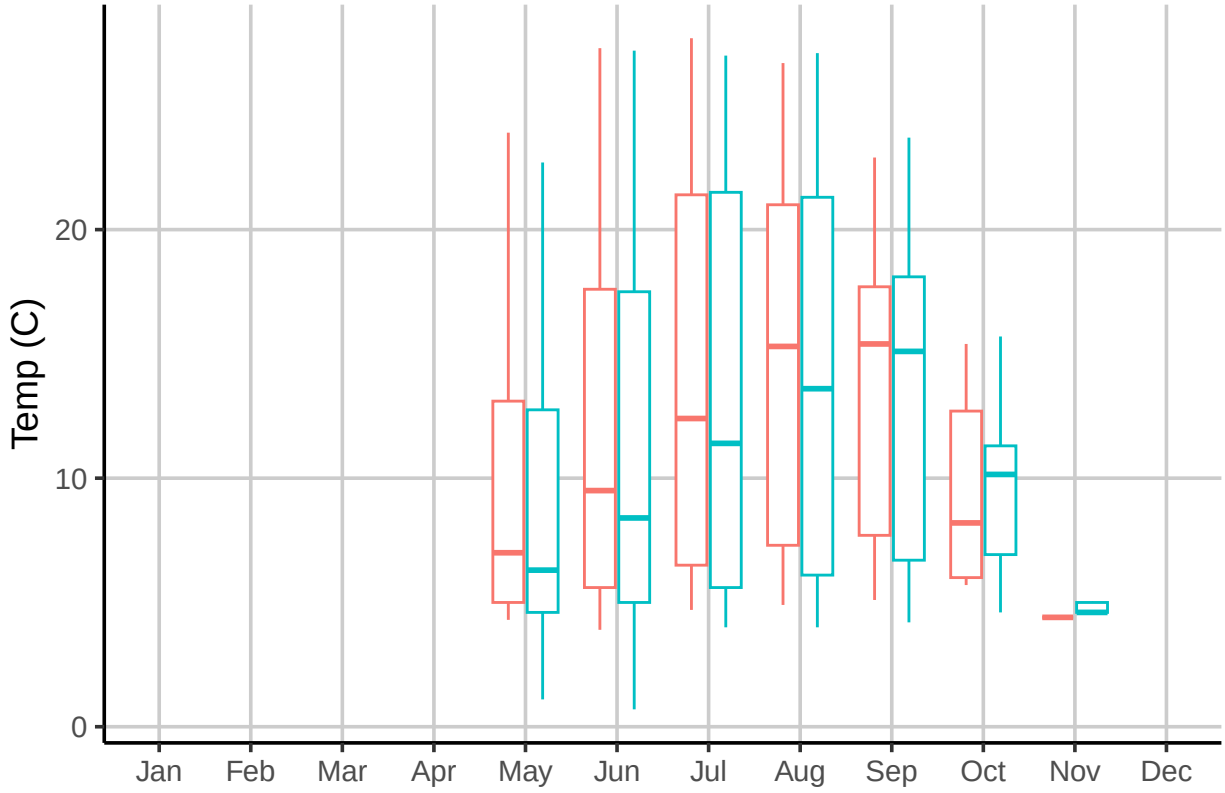
Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
# Lakes.plot2 is a boxplot of temperature for each lake with improved readability.
Lakes.plot2 <-
  ggplot(Peter.Paul.chem.nutrients,
         aes(x = factor(month,
                        levels = 1:12,
                        labels=month.abb),
                         y = temperature_C))+
  geom_boxplot(aes(color = lakename))+
  xlab("")+
  ylab("Temp (C)")+
  labs(color = "Lake Names")+
  theme(legend.position = "none")+
  scale_x_date(date_breaks = "1 months", date_labels = "%b %y")+
  scale_x_discrete(drop=FALSE)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
print(Lakes.plot2) #this has no month label for the purposes of the
```

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

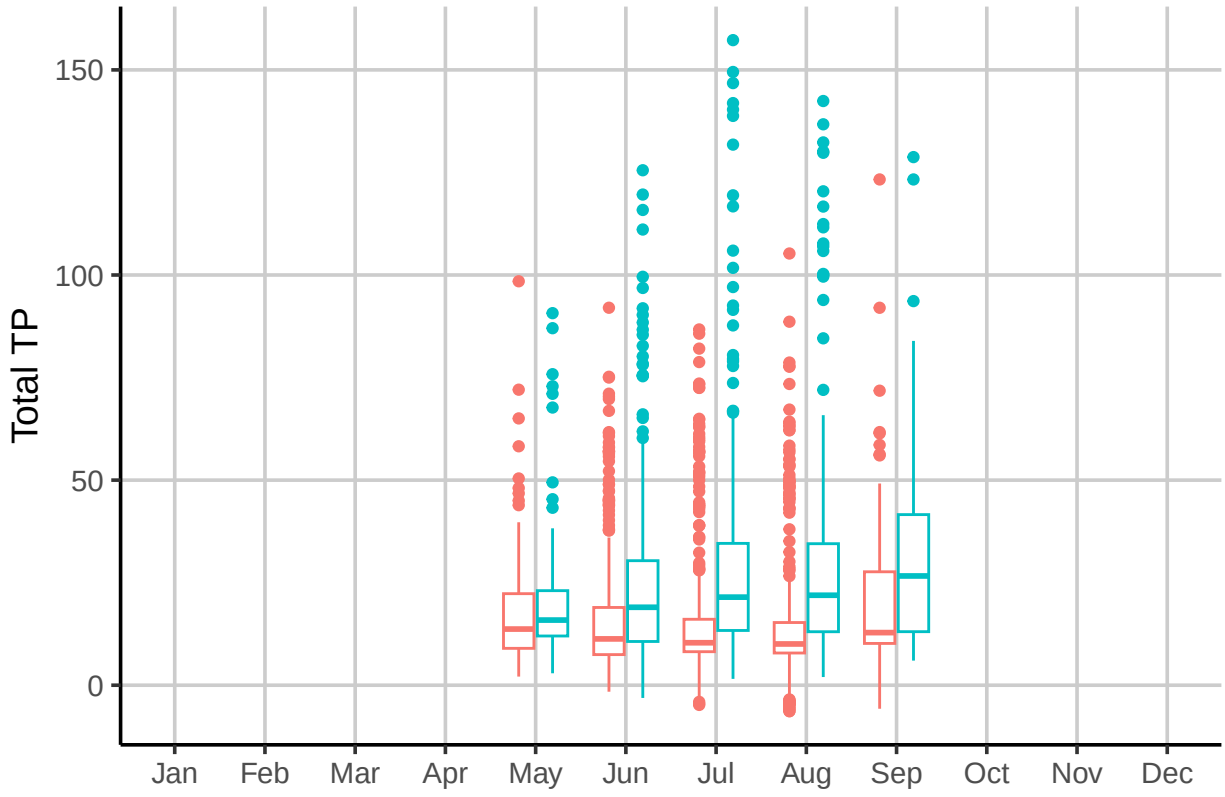



```
# combined plot.
# Lakes.plot3 is an object that has the box plot by month for Total TP.
Lakes.plot3 <-
  ggplot(Peter.Paul.chem.nutrients,
         aes(x = factor(month,
                        levels = 1:12,
                        labels=month.abb),
                         y = tp_ug)) +
  geom_boxplot(aes(color = lakename))+
   xlab("")+
  ylab("Total TP")+
labs(color = "Lake Names")+
  theme(legend.position = "none")+
  scale_x_date(date_breaks = "1 months", date_labels = "%b %y")+
  scale_x_discrete(drop=FALSE)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
print(Lakes.plot3) #no month label on x axis for the purpose of combining later
```

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

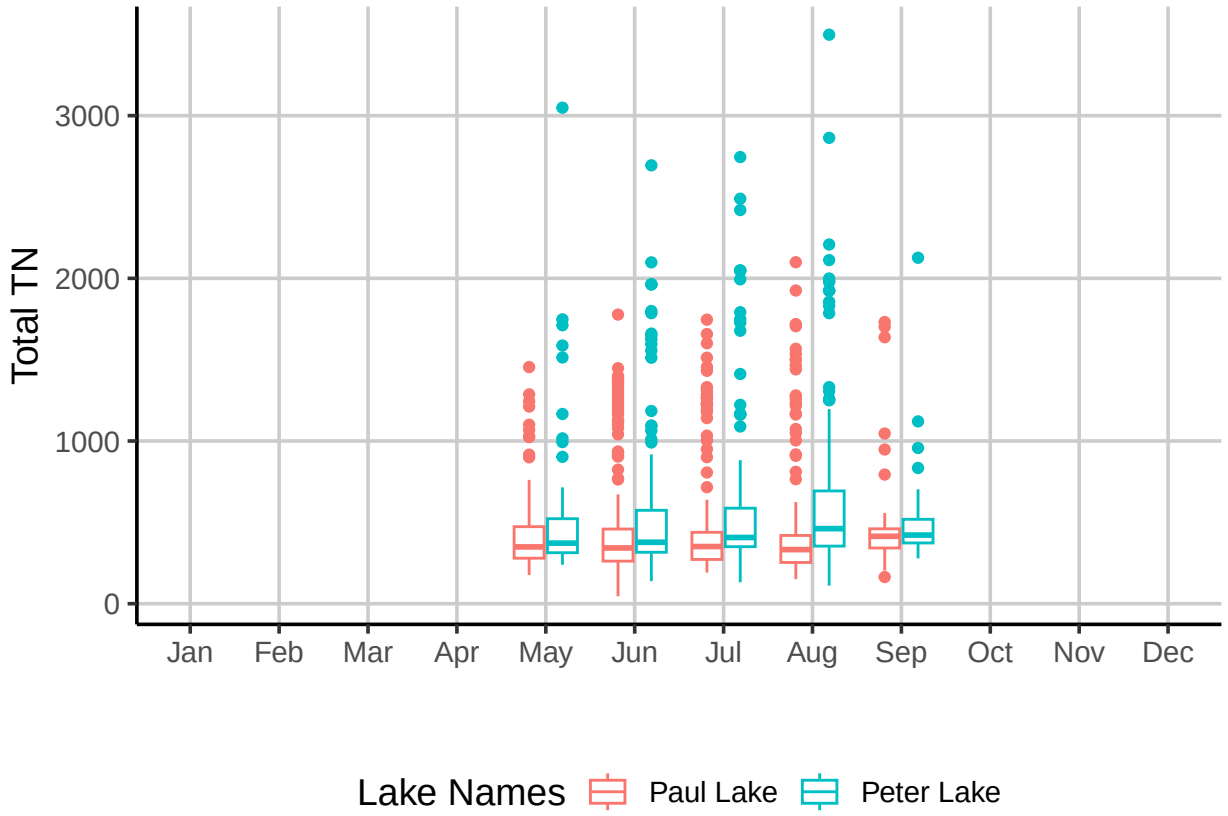



```
# Lakes.plot4 is an object holding the boxplot for the lakes for Total TN
Lakes.plot4 <-
  ggplot(Peter.Paul.chem.nutrients,
         aes(x = factor(month,
                        levels = 1:12,
                        labels=month.abb),
                         y = tn_ug)) +
  geom_boxplot(aes(color = lakename))+
   xlab("")+
  ylab("Total TN")+
labs(color = "Lake Names")+
    scale_x_date(date_breaks = "1 months", date_labels = "%b %y")+
  scale_x_discrete(drop=FALSE)+
  theme(legend.position = "bottom")
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

```
print(Lakes.plot4)
```

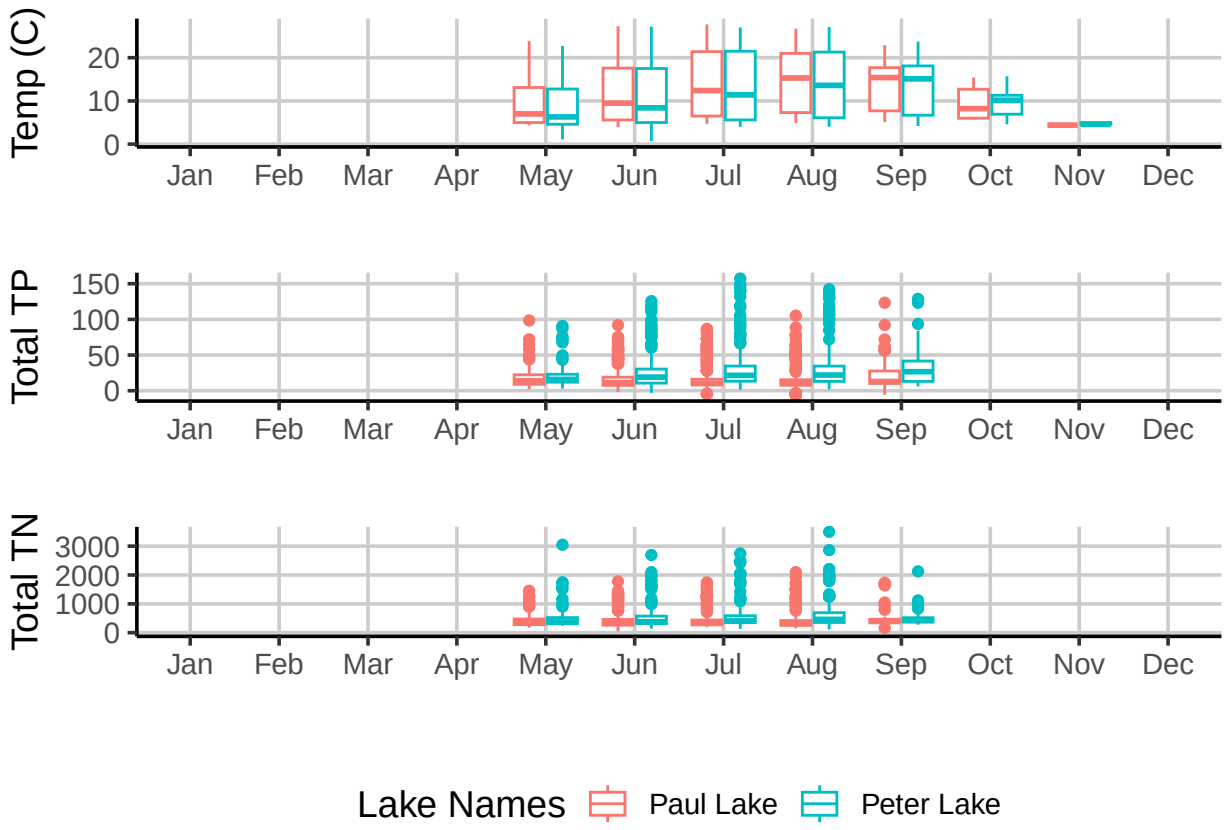
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```




```
# this code chunk is plotting the three graphs together.
combined.plot <-
  plot_grid(Lakes.plot2, Lakes.plot3, Lakes.plot4,
            nrow =3,
            align = 'v',
           # axis = "bt",
            rel_heights = c(1.25,1.25,1.75)
           )+
  ggtitle("Temp, TP, TN")
```

```
## 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(combined.plot)
```



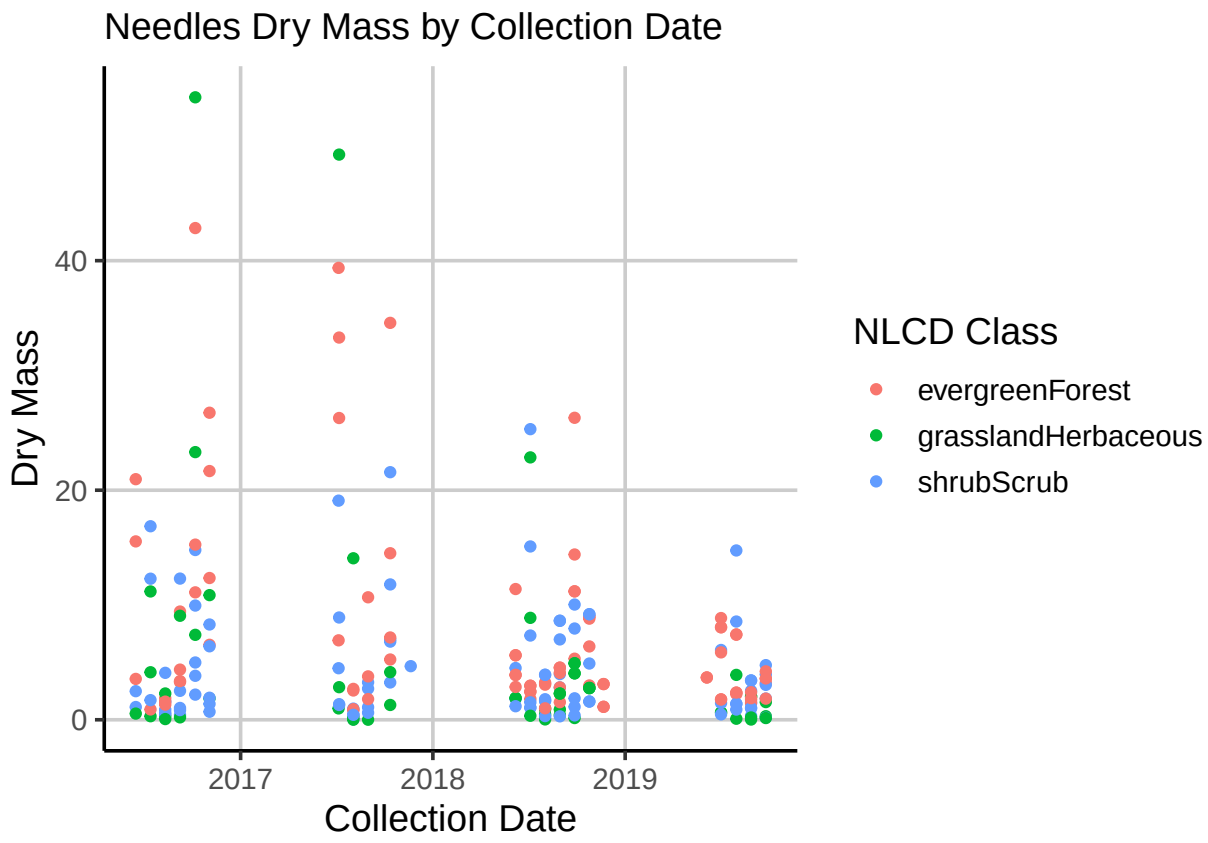
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: There are no data collected from December to April; it is primarily collected May through September, with some temperature data collected in October and November. There are a significant number of outliers for Total TP and Total TN. Peter Lake generally has higher outliers and a wider spread than Paul Lake.

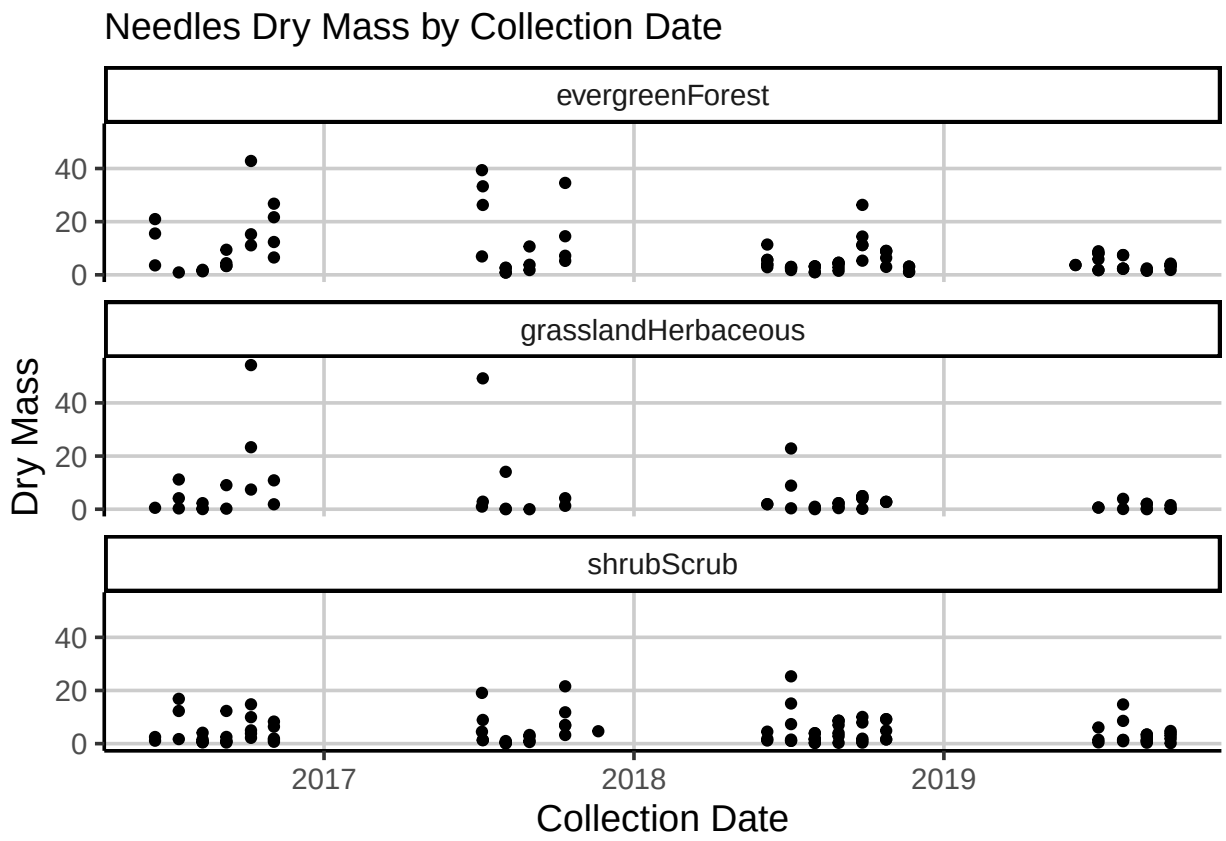
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
# This code chunk is making a new object for the scatterplot of the Needles subset
# of Neon.Niwo.litter.
Litter.Needles.plot1 <-
  ggplot(subset(Neon.Niwo.litter, functionalGroup == "Needles"),
         aes(x= collectDate, y = dryMass, color = nlcdClass)) +
  geom_point()+
labs(color = "NLCD Class")+
xlab("Collection Date")+
ylab("Dry Mass")+
  ggtitle("Needles Dry Mass by Collection Date")
print(Litter.Needles.plot1)
```




```
#7
# plotting the needles subset with a facet_wrap.
Litter.Needles.plot2 <-
  ggplot(subset(Neon.Niwo.litter, functionalGroup == "Needles"),
         aes(x= collectDate, y = dryMass)) +
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)+
xlab("Collection Date")+
ylab("Dry Mass")+
  ggtitle("Needles Dry Mass by Collection Date")
print(Litter.Needles.plot2)
```



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

Answer: I think that the faceted plot in Question 7 is more effective because within a single NLCD Class we are able to see the trends across the years, and looking down the columns we can visually compare each of the NLCD classes to each other. In the plot in Question 6, we can see some general trends through the years, but it is visually difficult to compare each of the NLCD classes to each other.