Assignment 5: Data Visualization

Erin Ansbro

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 #getting set up
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 C:/Users/eka19/OneDrive - Duke University/Documents/872/EDE_Fall2023
```

```
library(ggplot2)
#install.packages("cowplot")
library(cowplot)
```

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

```
getwd()
```

```
## [1] "C:/Users/eka19/OneDrive - Duke University/Documents/872/EDE_Fall2023"
```

```
#reading in the data
LakeChemistry <- read.csv(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")

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

#2 changing to dates
class((LakeChemistry$sampledate))
```

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

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

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

```
class(Litter$collectDate)
```

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

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

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

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 Building my theme
ErinTheme <- theme_light()+
  theme(legend.position="top",plot.background = element_rect(fill = "light gray"),
        axis.title = element_text(face="bold"))

theme_set(ErinTheme)
```

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()`).

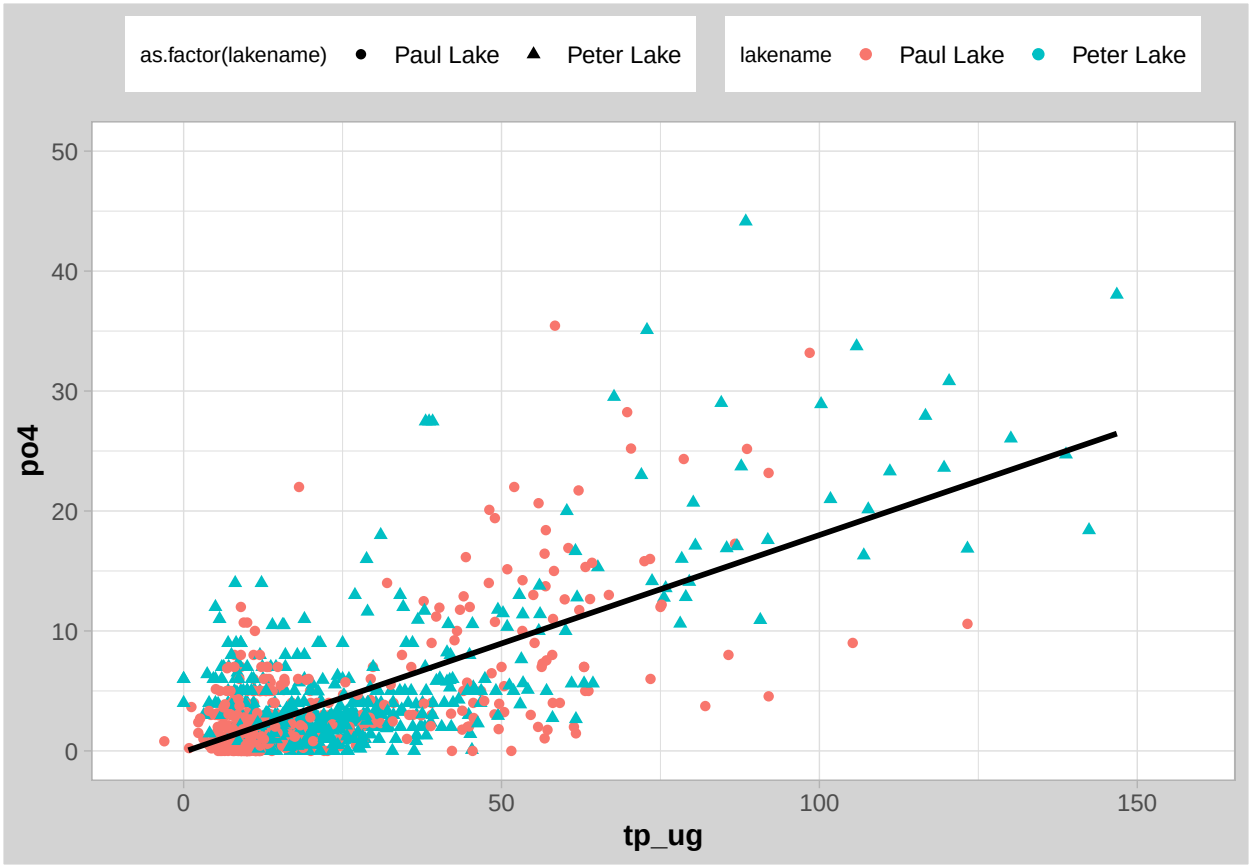
```
#4  Building dot plot for comparing tp and po4
PvsPo4 <- ggplot(LakeChemistry, aes(x=tp_ug, y=po4))+
  geom_point(aes(shape=as.factor(lakename), color=lakename))+
  ylim(0,50)+
  geom_smooth(aes(x=tp_ug, y=po4),method = lm, se = FALSE, color="black")+
   theme(legend.title = element_text(size=8))
print(PvsPo4)
```

```
## `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()`).

## 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 creating month column
LakeChemistry <-  LakeChemistry %>%
  mutate(Month=month(sampledate))

class(LakeChemistry$Month)
```

```
## [1] "numeric"
```

```
unique(LakeChemistry$Month)
```

```
## [1]  5  6  7  8  9 10 11  2
```

```
#changing month to factor, adding 12 levels, and labelling the 12 months
LakeChemistry$Month <-factor(
  LakeChemistry$Month, levels = c(1,2,3,4,5,6,7,8,9,10,11,12),labels =c(
    "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
    "Oct", "Nov", "Dec") )
class(LakeChemistry$Month)
```

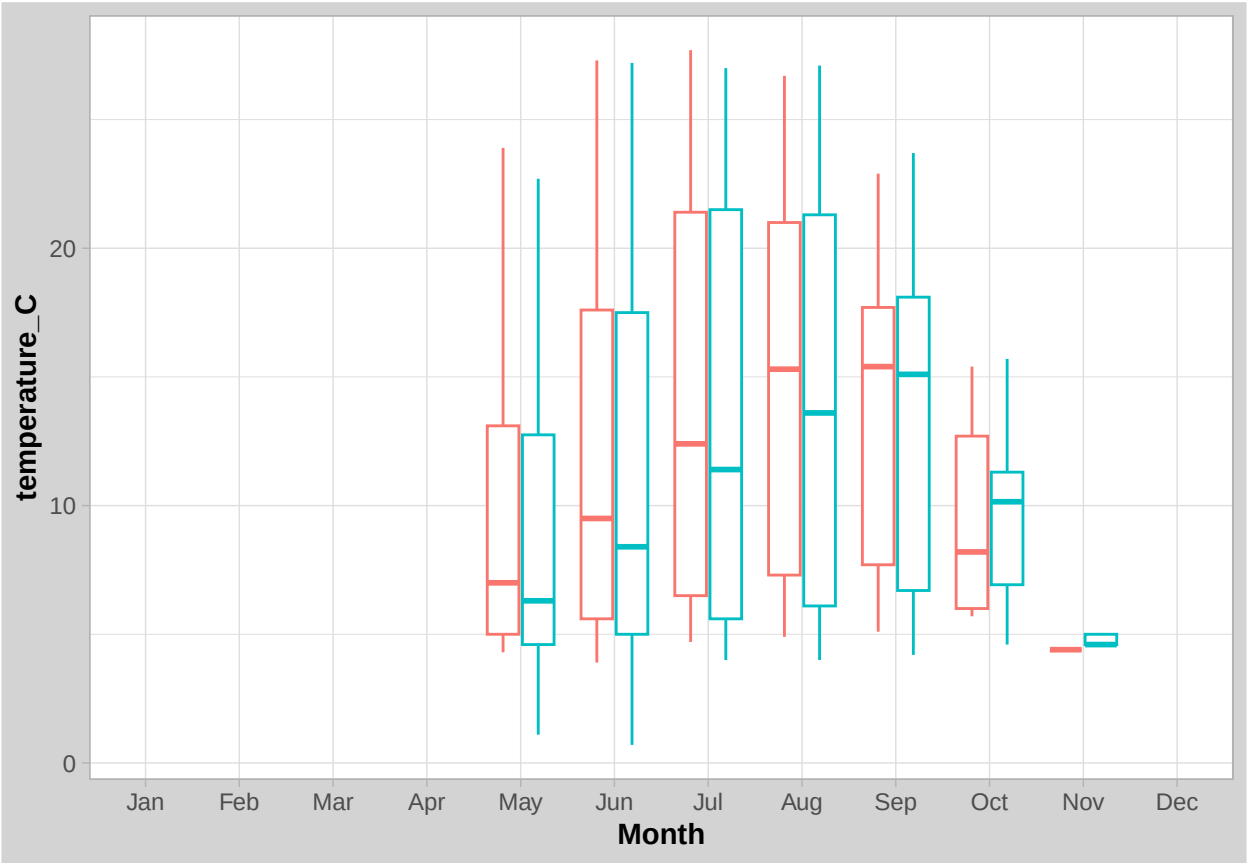
```
## [1] "factor"
```

```
unique(LakeChemistry$Month)
```

```
## [1] May Jun Jul Aug Sep Oct Nov Feb
## Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

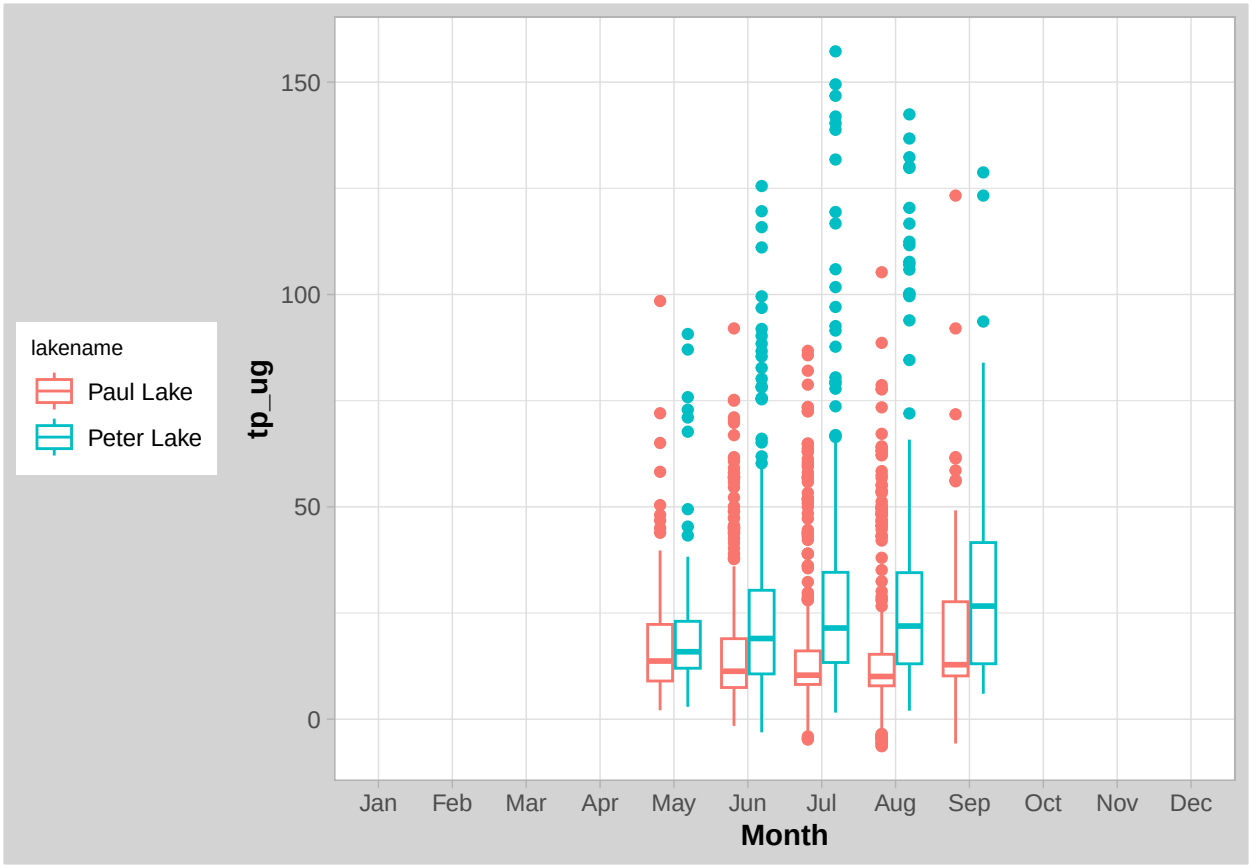
```
#creating three boxplots for temp, tn and tn
BoxplotTemp <- ggplot(LakeChemistry, aes(x=Month, y=temperature_C))+
  geom_boxplot(aes(color=lakename))+
    theme(legend.position="none")+
  scale_x_discrete(drop=FALSE)
print(BoxplotTemp)
```

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



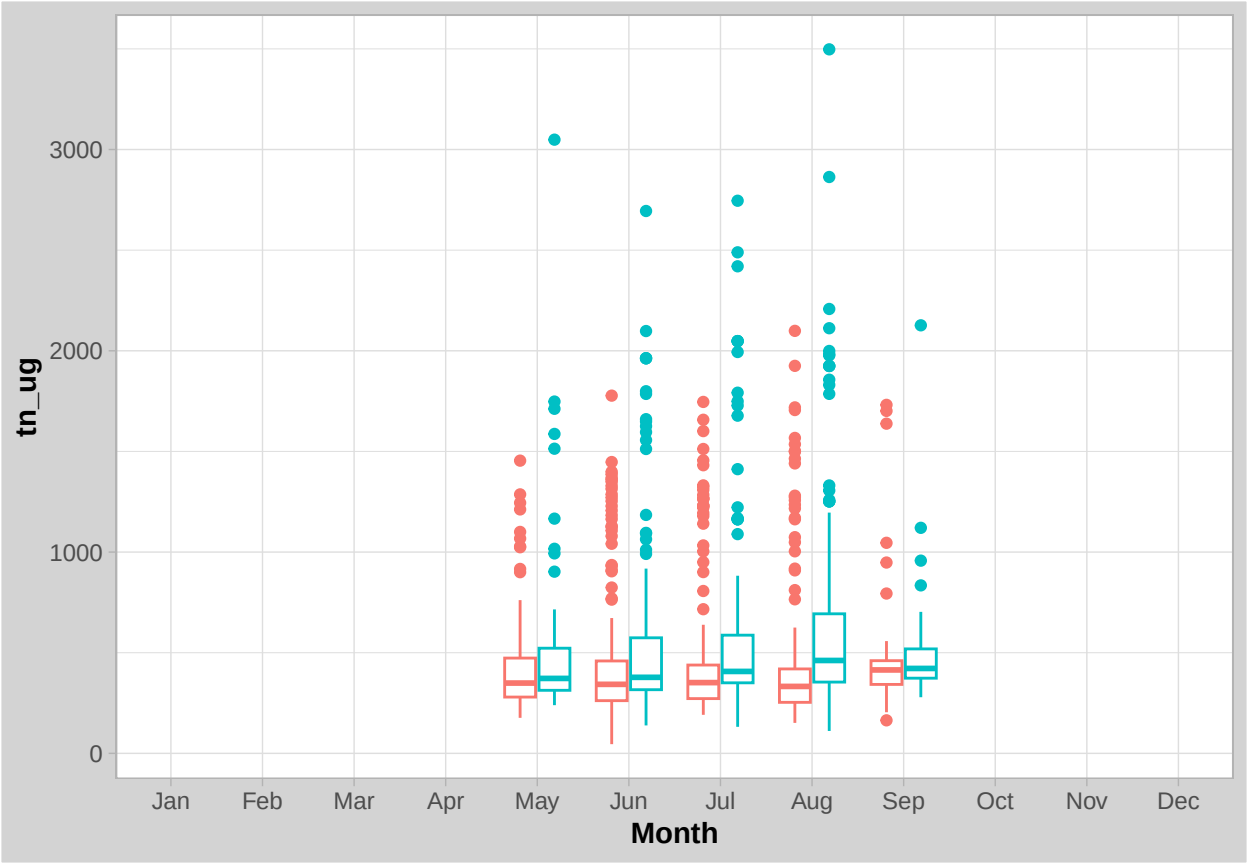

```
BoxplotTP <- ggplot(LakeChemistry, aes(x=Month, y=tp_ug))+
  geom_boxplot(aes(color=lakename))+
 theme(legend.position = "left", legend.title = element_text(size=8))+
  scale_x_discrete(drop=FALSE)
print(BoxplotTP)
```

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




```
BoxplotTN <- ggplot(LakeChemistry, aes(x=Month, y=tn_ug))+
  geom_boxplot(aes(color=lakename))+
    theme(legend.position="none")+
  scale_x_discrete(drop=FALSE)
print(BoxplotTN)
```

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



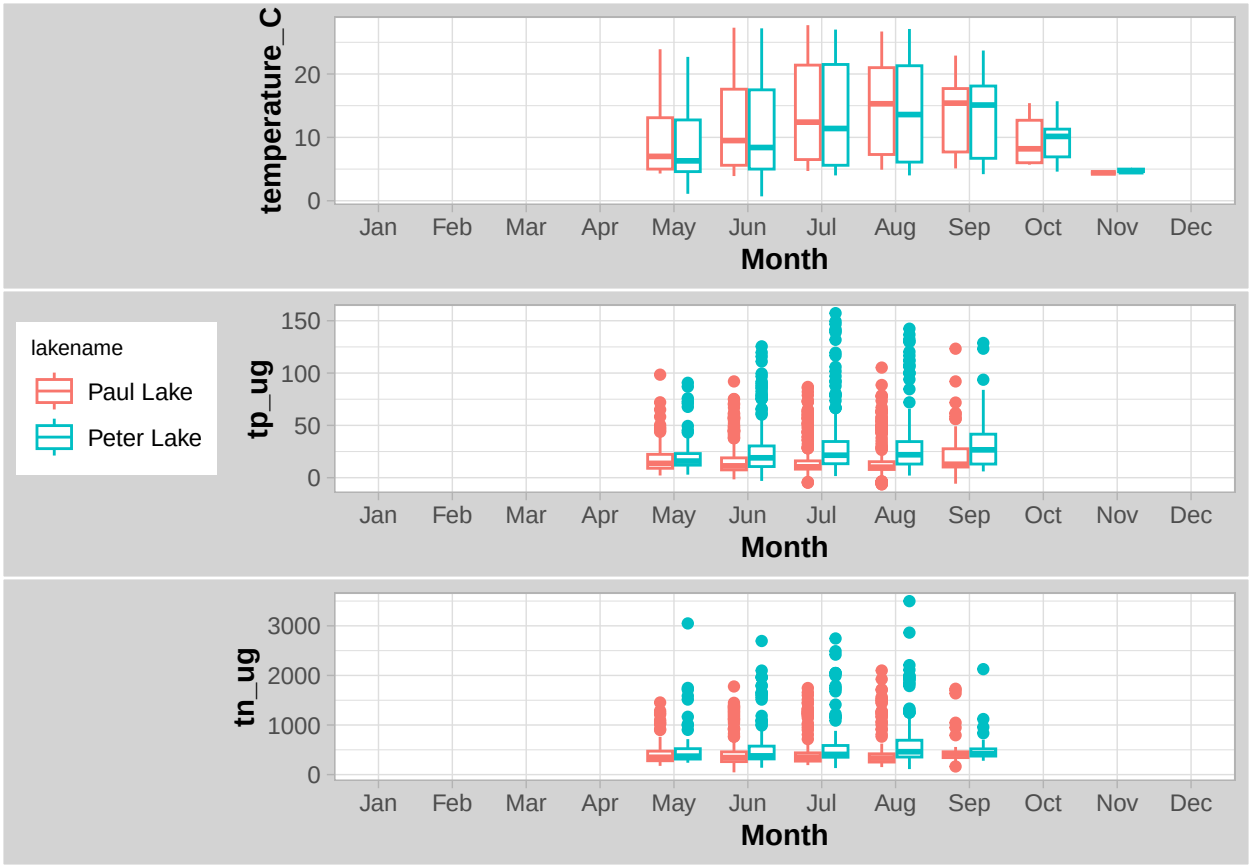

```
#combining those three boxplots to be in the same visual
Plotgrid <- plot_grid(BoxplotTemp, BoxplotTP, BoxplotTN, nrow=3, axis='tblr', align= "vh")
```

```
## 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(Plotgrid)
```



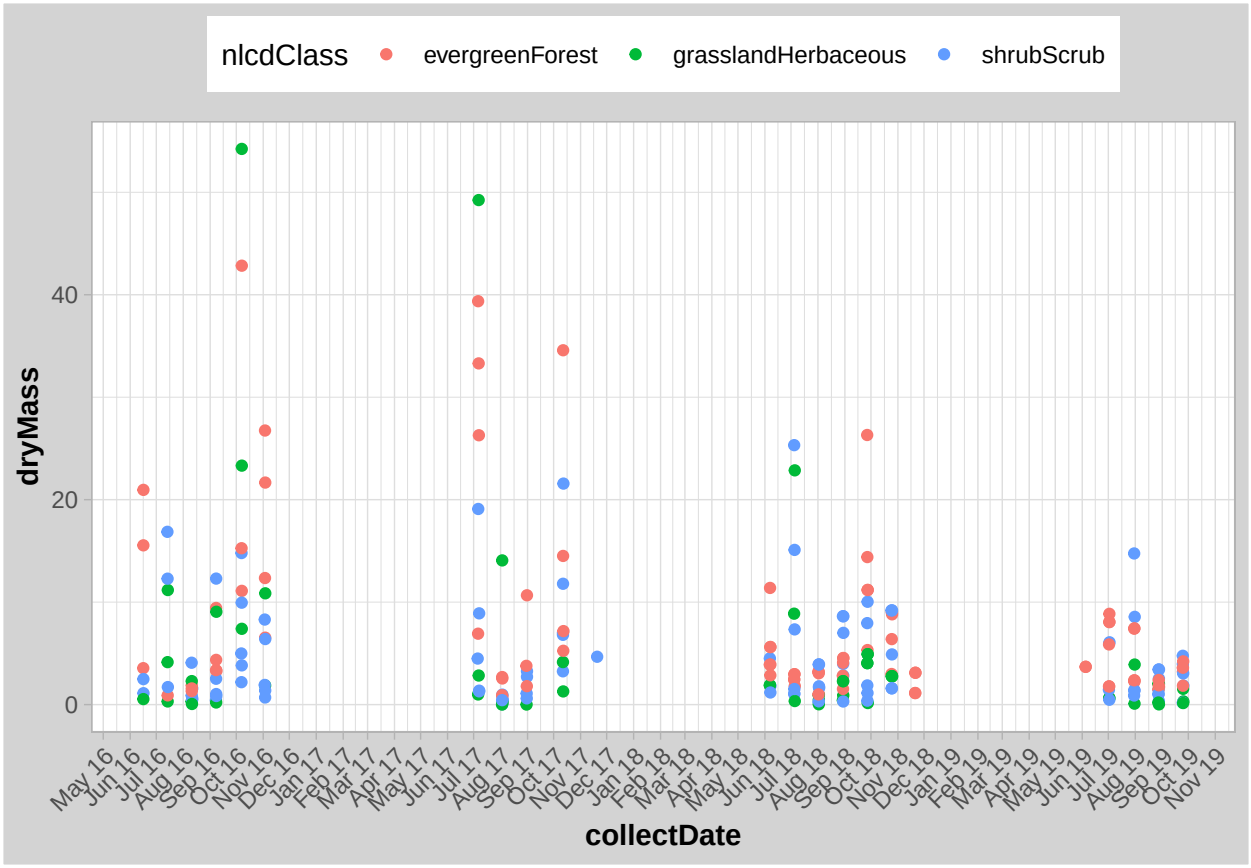
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: There's only data for warmer months. There is no data for tp_ug or tn_ug for the months of October and November.There are outliers for tp_ug and tn_ug for higher micrograms.

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 Filtering for needles
LitterFilter <- filter(Litter, functionalGroup == "Needles")
LitterPlot <- ggplot(LitterFilter) + #create plot for needle mass by date
  geom_jitter(aes(x = collectDate, y = dryMass, color = nlcdClass))+ #color coding by class
  scale_x_date(date_labels="%b %y",date_breaks  ="1 month")+
  theme(axis.text.x=element_text(angle=45, hjust = 1))
print(LitterPlot)
```

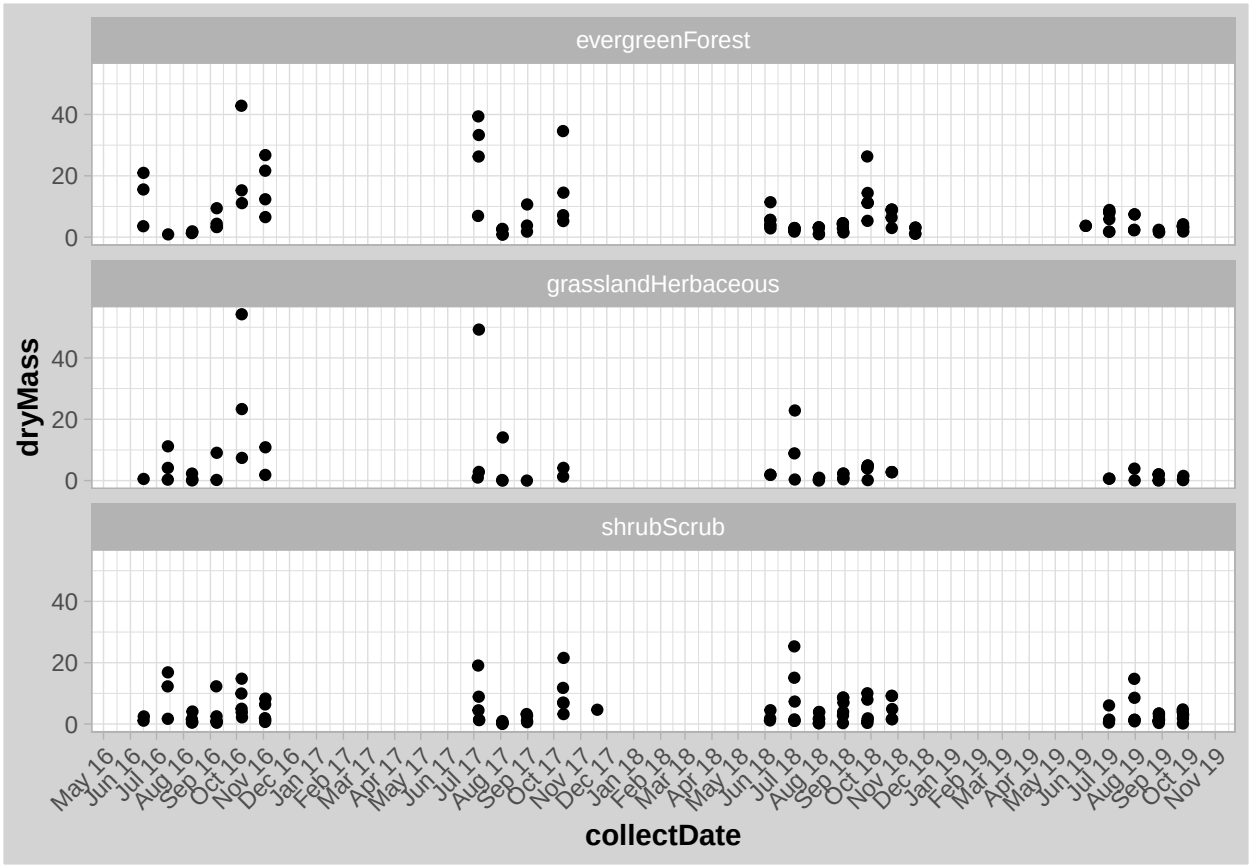



```
#7
unique(LitterFilter$nlcdClass)
```

```
## [1] "shrubScrub"          "grasslandHerbaceous" "evergreenForest"
```

```
#creating three separate plots for the class instead of color coding
Litterplot2 <- ggplot(LitterFilter)+
  geom_jitter(aes(x=collectDate, y=dryMass))+
  facet_wrap(vars(nlcdClass), nrow=3)+
  scale_x_date(date_labels="%b %y",date_breaks  ="1 month")+
  theme(axis.text.x=element_text(angle=45, hjust = 1))

print(Litterplot2)
```



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

Answer: I think it is easier to see the facet wrap plot, this is because the data is somewhat similar, and the colors are hard to distingush if they are all all on top of one another. If you break the plot into three separate ones, it's easier to see the points and thusly, the similarities and differences, for the different classes. For example, in July 2018, you can easily see that there was a lot less dryMass for evergeen forest in the facet_wrap. I do not pick up on that difference as easily when I look at the color plot.