Assignment 5: Data Visualization

Jaimie Wargo

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.

```r
#1
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(here)
```

```
## Warning: package 'here' was built under R version 4.2.3
```

```
## here() starts at C:/Users/purec/Documents/Duke/Fall_2023/EDA/EDE_Fall2023
```

```r
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.2.3
```

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

```r
here()
```

```
## [1] "C:/Users/purec/Documents/Duke/Fall_2023/EDA/EDE_Fall2023"
```

```r
PeterPaul <- read.csv(file = here('Data','Processed_KEY',
                                  'NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
                      stringsAsFactors = T)

NiwotLitter <- read.csv(file = here('Data','Processed_KEY',
                                    'NEON_NIWO_Litter_mass_trap_Processed.csv'),
                      stringsAsFactors = T)
#2
class(PeterPaul$sampledate)
```

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

```r
class(NiwotLitter$collectDate)
```

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

```
#Dates are read in as factors, going to change to date
#Both data frames are formatted in year-month-day

PeterPaul$sampledate <- ymd(PeterPaul$sampledate)
NiwotLitter$collectDate <- ymd(NiwotLitter$collectDate)

class(PeterPaul$sampledate)
```

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

```
class(NiwotLitter$collectDate)
```

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

```
#Both are now listed as dates
```

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
mytheme <- theme_gray()+
  theme(plot.title = element_text(size = 16, hjust= 0),
        axis.title = element_text(size = 13),
        legend.position = 'right')

theme_set(mytheme)
```

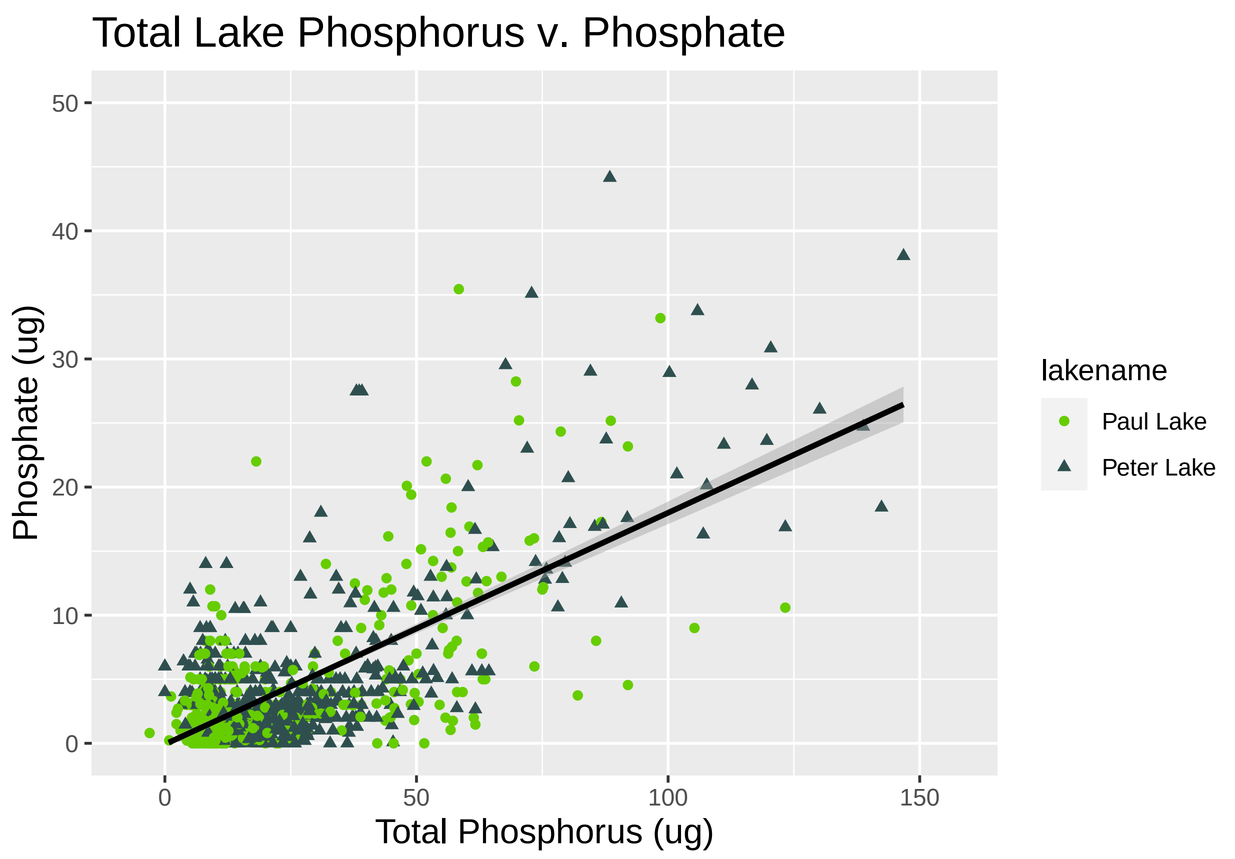
Create graphs

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
Lake_Phosphorus <- ggplot(PeterPaul, aes(x = tp_ug, y = po4))+
  geom_point(aes(color= lakename, shape = lakename))+
  geom_smooth(method=lm, color='black')+
  scale_color_manual(values = c("chartreuse3", "darkslategray"))+
  labs(title = "Total Lake Phosphorus v. Phosphate",
       x = "Total Phosphorus (ug)",
       y = "Phosphate (ug)")+
  ylim(0,50)

print(Lake_Phosphorus)
```

```
## `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).
```



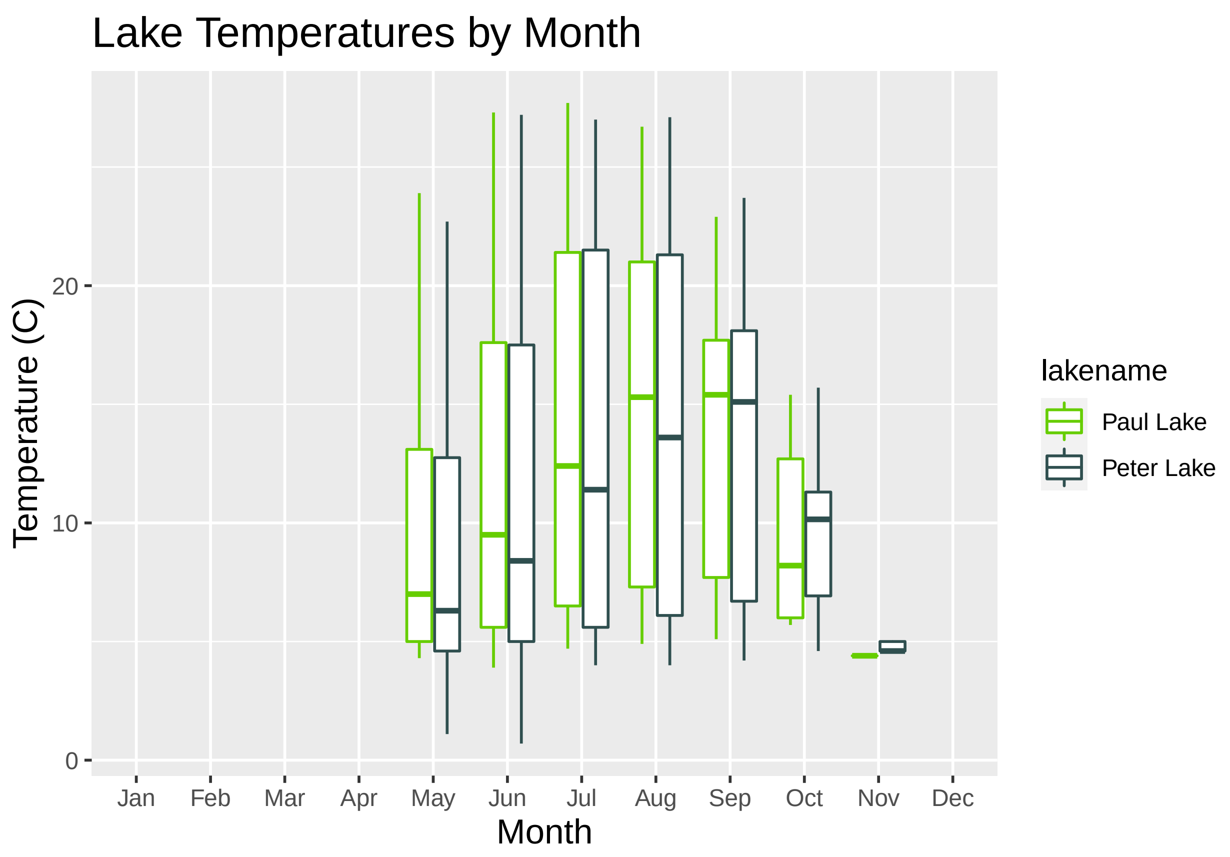
```
#I added the ylim to show from 0 to 50, as there was an outlier above 50.
#I specified the aesthetics within the geom_point so that the geom_smooth would
  #create one line apply to both lakes and not produce two lines for each lake.
```

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

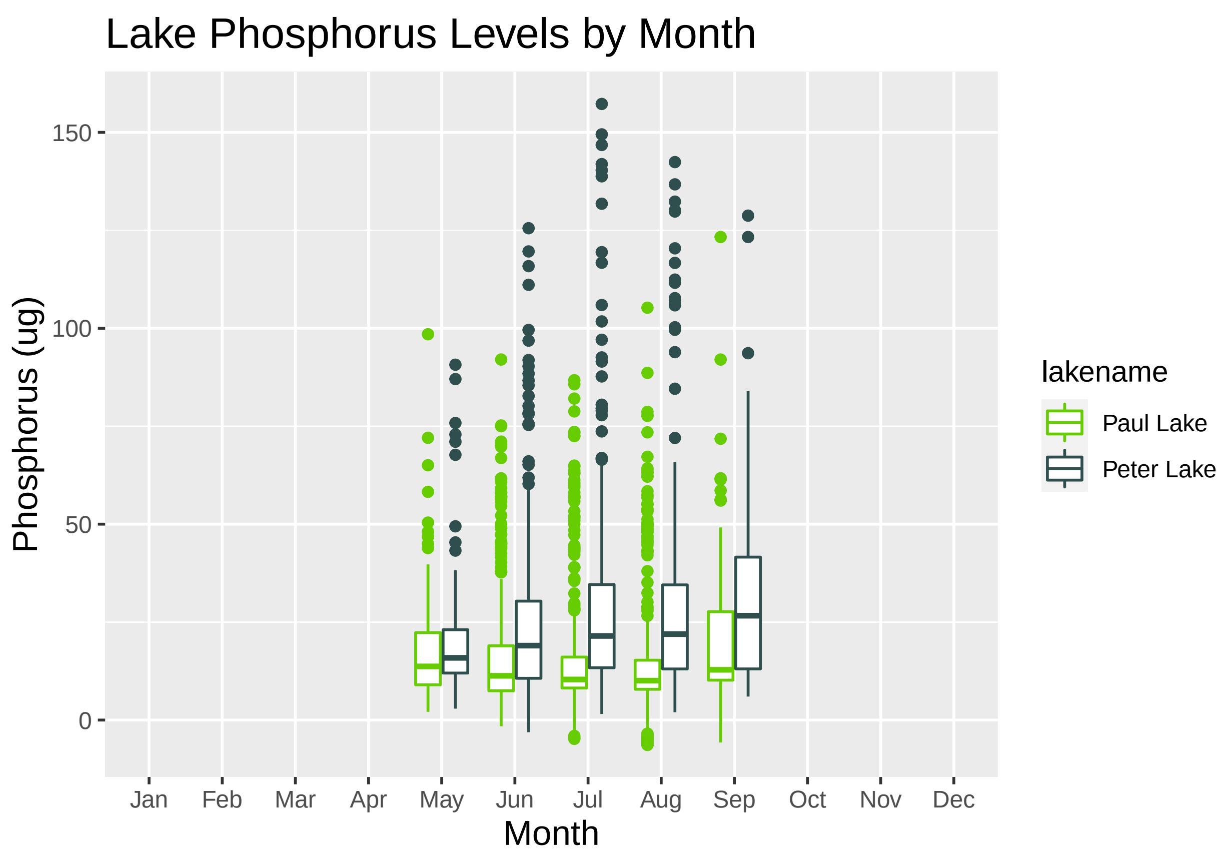
```r
#5
PeterPaul$month <- factor(PeterPaul$month,
  levels=c(1:12),labels=month.abb)
#I am changing the month column to a factor, specifying the levels and what the
  #labels should be; this was simple using the month.abb function

TempBox <- ggplot(PeterPaul, aes(x=month, y=temperature_C))+
  geom_boxplot(aes(color=lakename))+
  scale_color_manual(values = c("chartreuse3", "darkslategray"))+
  labs(title = "Lake Temperatures by Month",
       x = "Month",
       y = "Temperature (C)")+
  scale_x_discrete(drop=F)
print(TempBox)
```

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



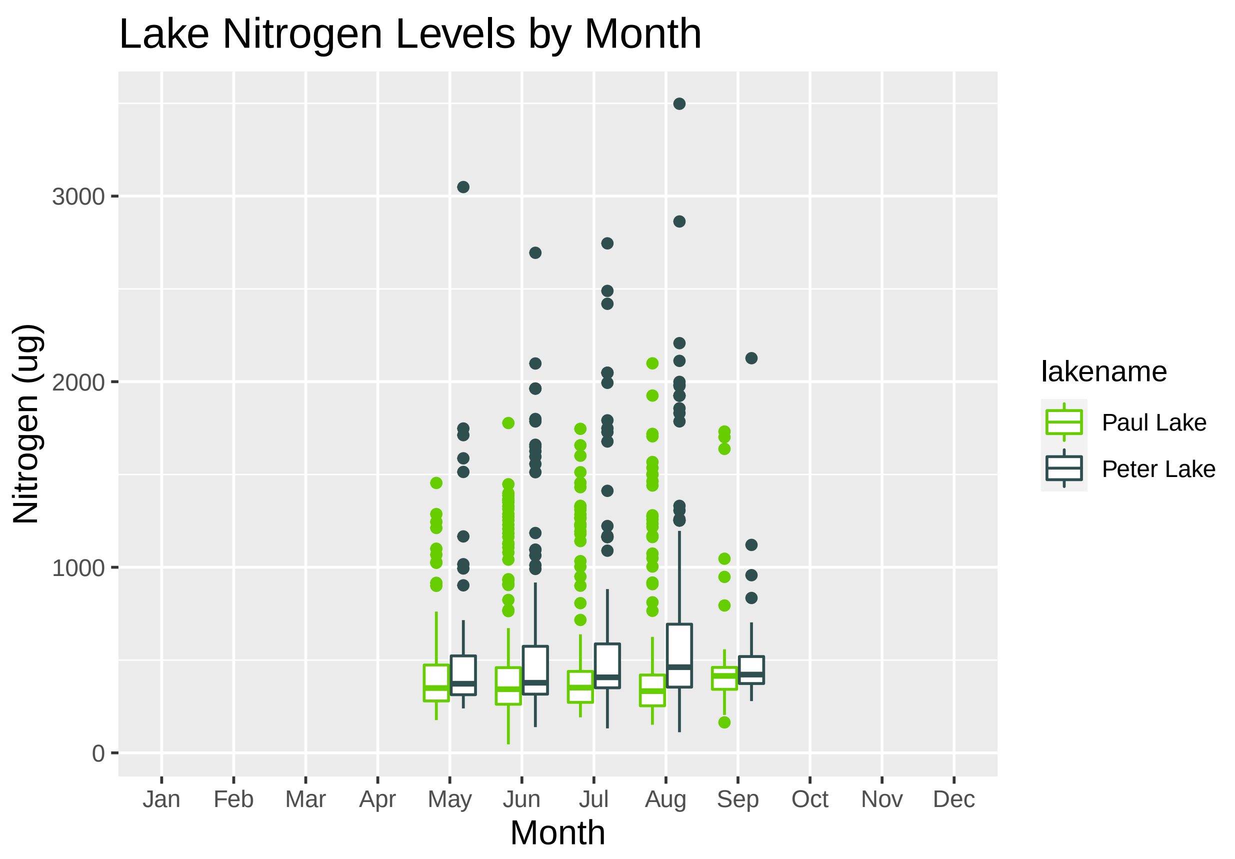
```r
TPBox <- ggplot(PeterPaul, aes(x=month, y=tp_ug))+
  geom_boxplot(aes(color=lakename))+
  scale_color_manual(values = c("chartreuse3", "darkslategray"))+
  labs(title = "Lake Phosphorus Levels by Month",
       x = "Month",
       y = "Phosphorus (ug)")+
  scale_x_discrete(drop=F)
print(TPBox)
```

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



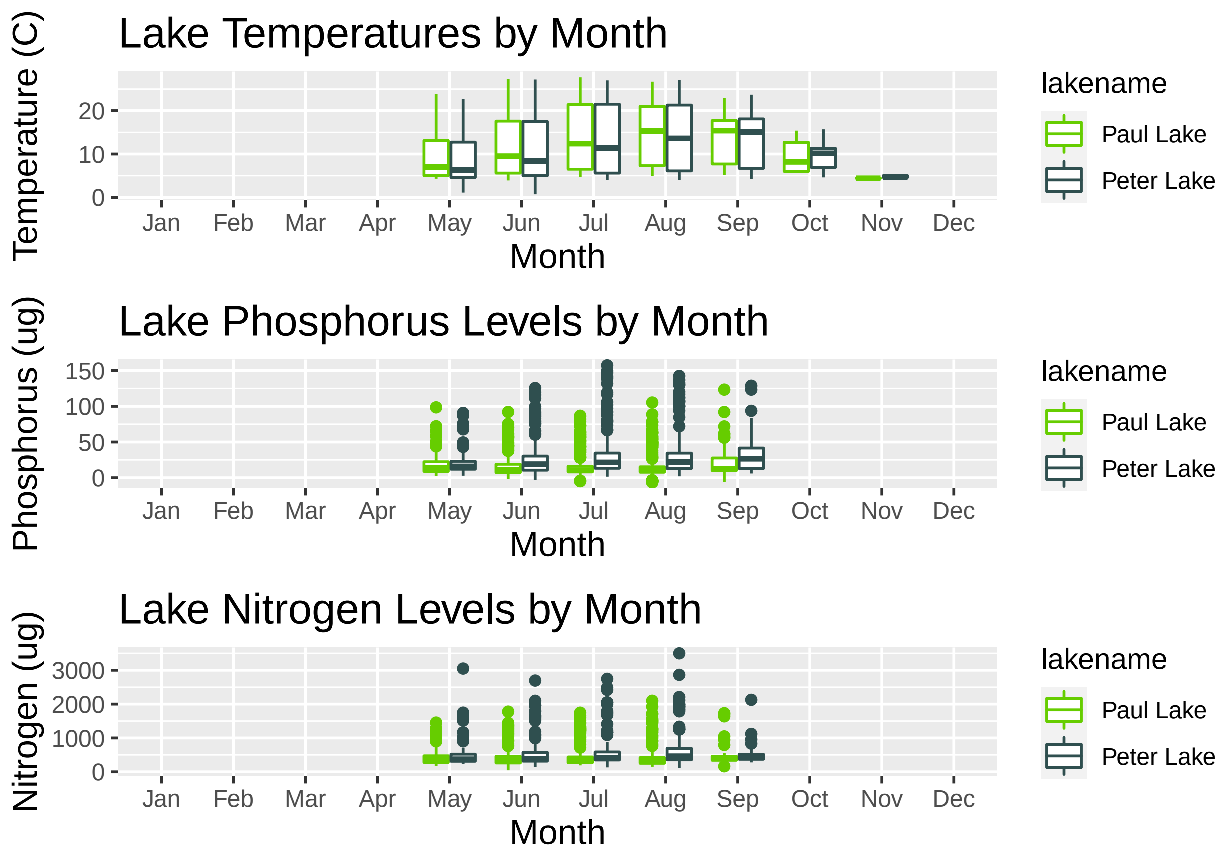
```r
TNBox <- ggplot(PeterPaul, aes(x=month, y=tn_ug))+
  geom_boxplot(aes(color=lakename))+
  scale_color_manual(values = c("chartreuse3", "darkslategray"))+
  labs(title = "Lake Nitrogen Levels by Month",
       x = "Month",
       y = "Nitrogen (ug)")+
  scale_x_discrete(drop=F)
print(TNBox)
```

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



```
#To show all months, I used scale_x_discrete with the argument drop=False so that
  #even factors with no observations were not dropped.

plot_grid(TempBox, TPBox, TNBox, nrow = 3, align="v")
```

```
## 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).
```



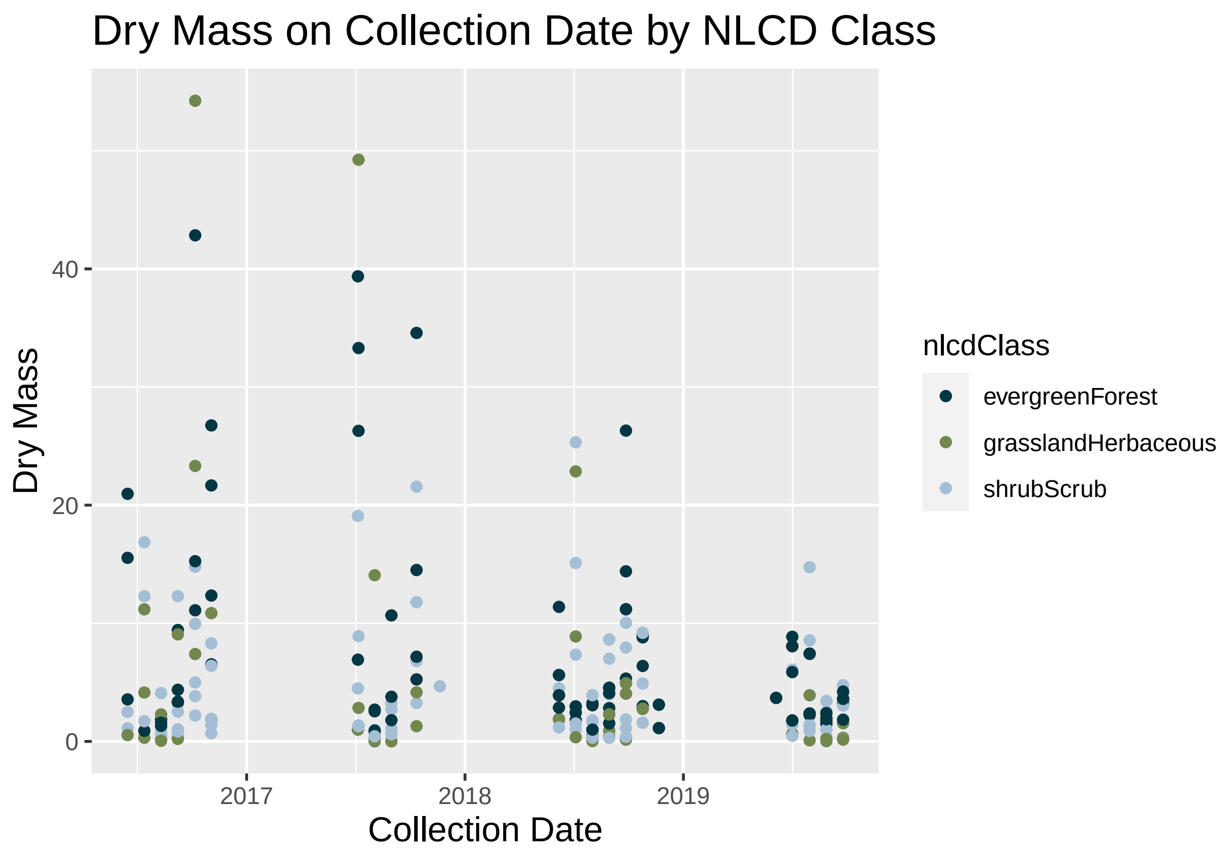
```r
#This function from cowplot combines the three plots in one image
```

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

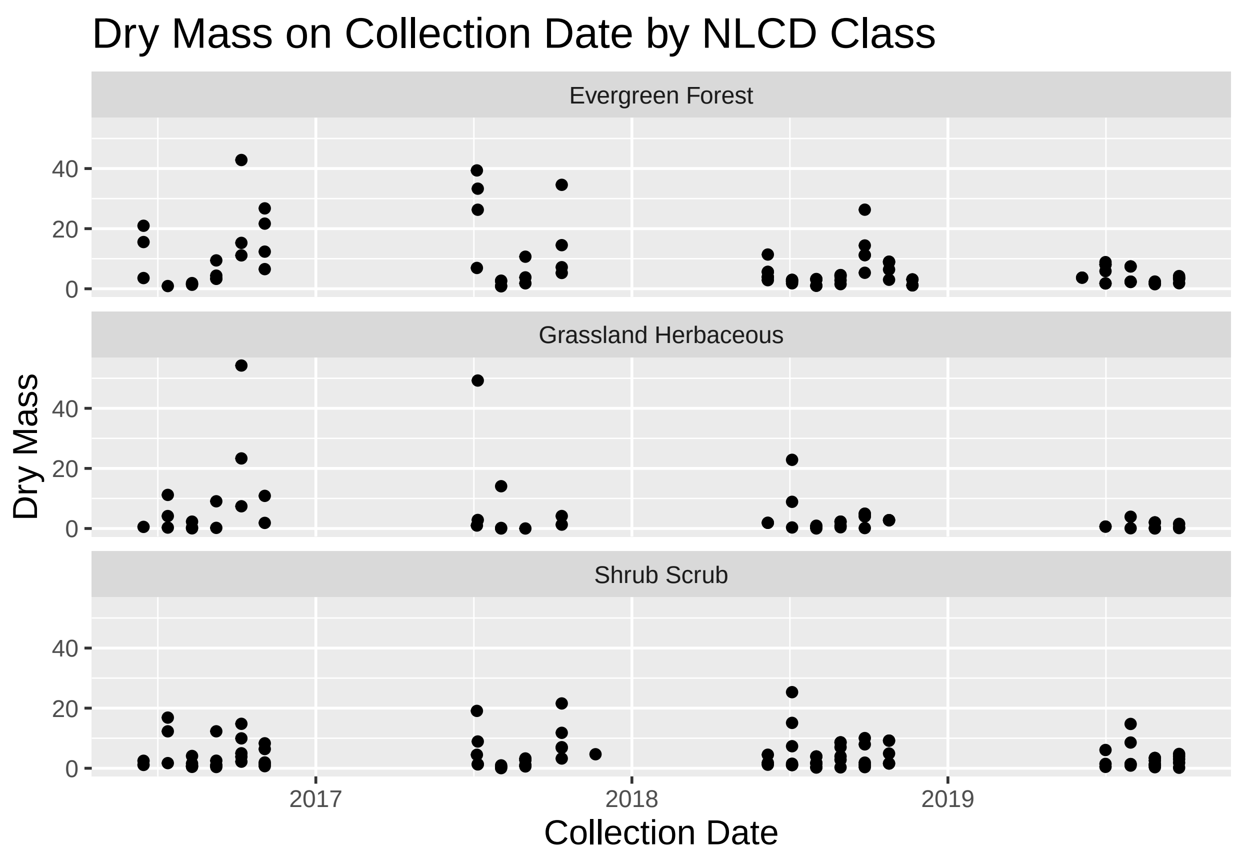
> Answer: The phosphorus and nitrogen concentration values have many high outliers. Peter Lake typically has higher nutrient concentrations than Paul Lake, excluding the observations of phosphorus in May. Temperatures are very comparable between the two lakes, and they follow the expected patterns of increasing temperatures in the summer. Phosphorus concentrations also seem to increase in the summer, whereas nitrogen does not have an obvious seasonal pattern.

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.

```r
#6
NeedlesPlot <- ggplot(subset(NiwotLitter,
                             NiwotLitter$functionalGroup == 'Needles'),
                      aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point()+
  labs(x="Collection Date",
       y="Dry Mass",
       title="Dry Mass on Collection Date by NLCD Class")+
  scale_color_manual(values = c("#023743", "#72874E", "#A4BED5"))
print(NeedlesPlot)
```



```
#7
nlcdLabels <- c("Evergreen Forest", "Grassland Herbaceous", "Shrub Scrub")
names(nlcdLabels) <- c("evergreenForest", "grasslandHerbaceous", "shrubScrub")
#I included this because I wanted to change the labels on the facet grids.
  #I found how to do this on Stack Overflow.

NeedlesFacetPlot <- ggplot(subset(NiwotLitter,
                             NiwotLitter$functionalGroup == 'Needles'),
                      aes(x=collectDate, y=dryMass))+
  geom_point()+
  labs(x="Collection Date",
       y="Dry Mass",
       title="Dry Mass on Collection Date by NLCD Class")+
  facet_wrap(vars(nlcdClass), nrow=3,
             labeller = labeller(nlcdClass = nlcdLabels))
print(NeedlesFacetPlot)
```



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

> Answer: I think 7 is much more effective. It is much clearer to see the data separated out between the NLCD classes rather than trying to interpret the data when it is all overlapping, even if they are separated aesthetically with colors or shapes.