Assignment 5: Data Visualization

Meg Manning

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 Set up workspace
getwd()
```

```
## [1] "/Users/margaretmanning/Desktop/GitHub/ENVIRON 872/EDE_Fall2023"
```

```r
#load packages
library("tidyverse"); library("lubridate");
```

```
## -- 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("here"); library("cowplot")
```

```
## here() starts at /Users/margaretmanning/Desktop/GitHub/ENVIRON 872/EDE_Fall2023
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

```
## [1] "/Users/margaretmanning/Desktop/GitHub/ENVIRON 872/EDE_Fall2023"
```

```
#read in data
NTL_LTER_processed <- read.csv(
  "./Data/Processed_Key/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)

Niwot_Ridge_processed <- read.csv(
  "./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

#2 check date
class(NTL_LTER_processed$sampledate)
```

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

```
class(Niwot_Ridge_processed$collectDate)
```

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

```
#change to date
NTL_LTER_processed$sampledate <- as.Date(NTL_LTER_processed$sampledate,
                                         format = "%Y-%m-%d")
Niwot_Ridge_processed$collectDate <- as.Date(Niwot_Ridge_processed$collectDate,
                                             format = "%Y-%m-%d")
#Check class
class(NTL_LTER_processed$sampledate)
```

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

```
class(Niwot_Ridge_processed$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 build a theme
my_theme <- theme(
    plot.title = element_text(
      color = "blue"),
    plot.background = element_rect(
      fill = "white"),
    legend.title = element_rect(
      color = "blue",
      fill = "white"),
    legend.position = "right")
```

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.

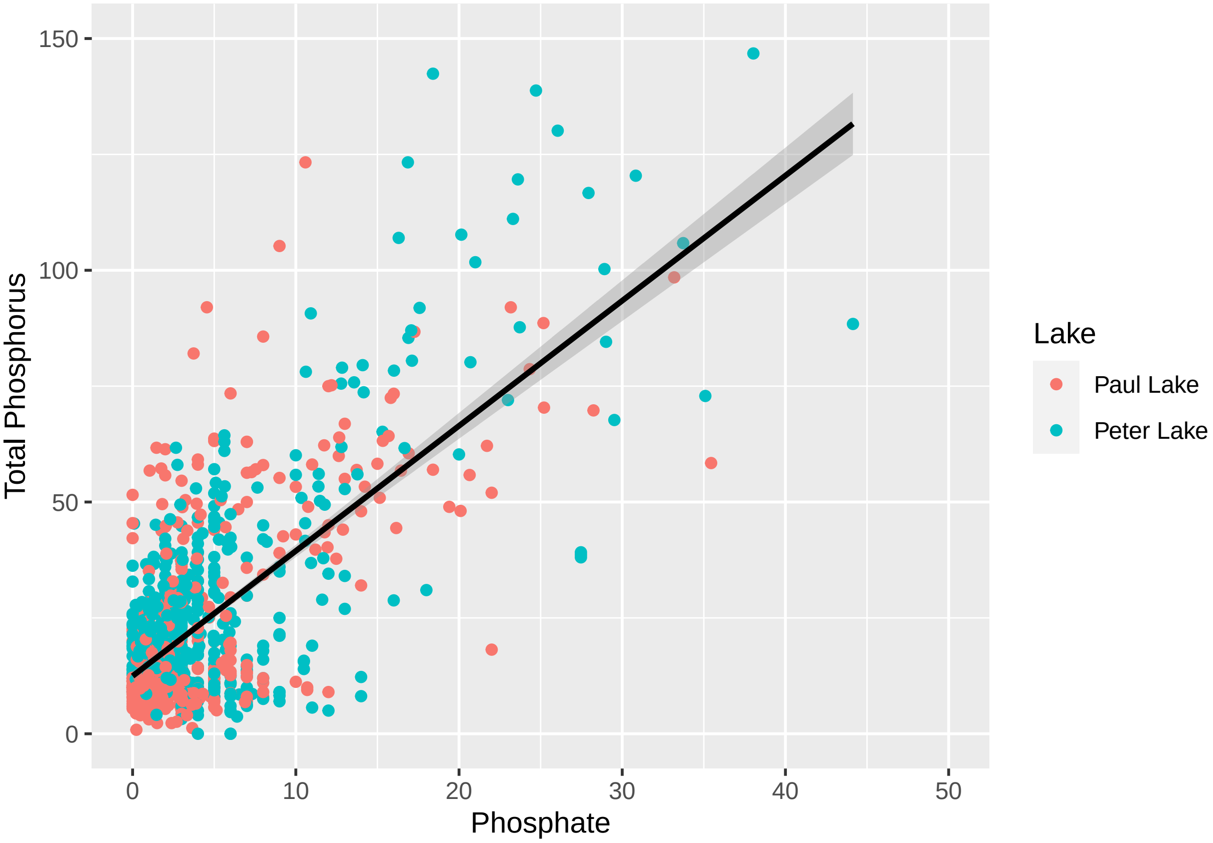
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4 Total phosphorus (tp_ug) by phosphate (po4)
ggplot(NTL_LTER_processed, aes(x = po4,
           y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  labs(y = "Total Phosphorus",
       x = "Phosphate",
       color = "Lake")+
  xlim(0, 50) +
  ylim(0, 150)
```

```
## `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()`).
```

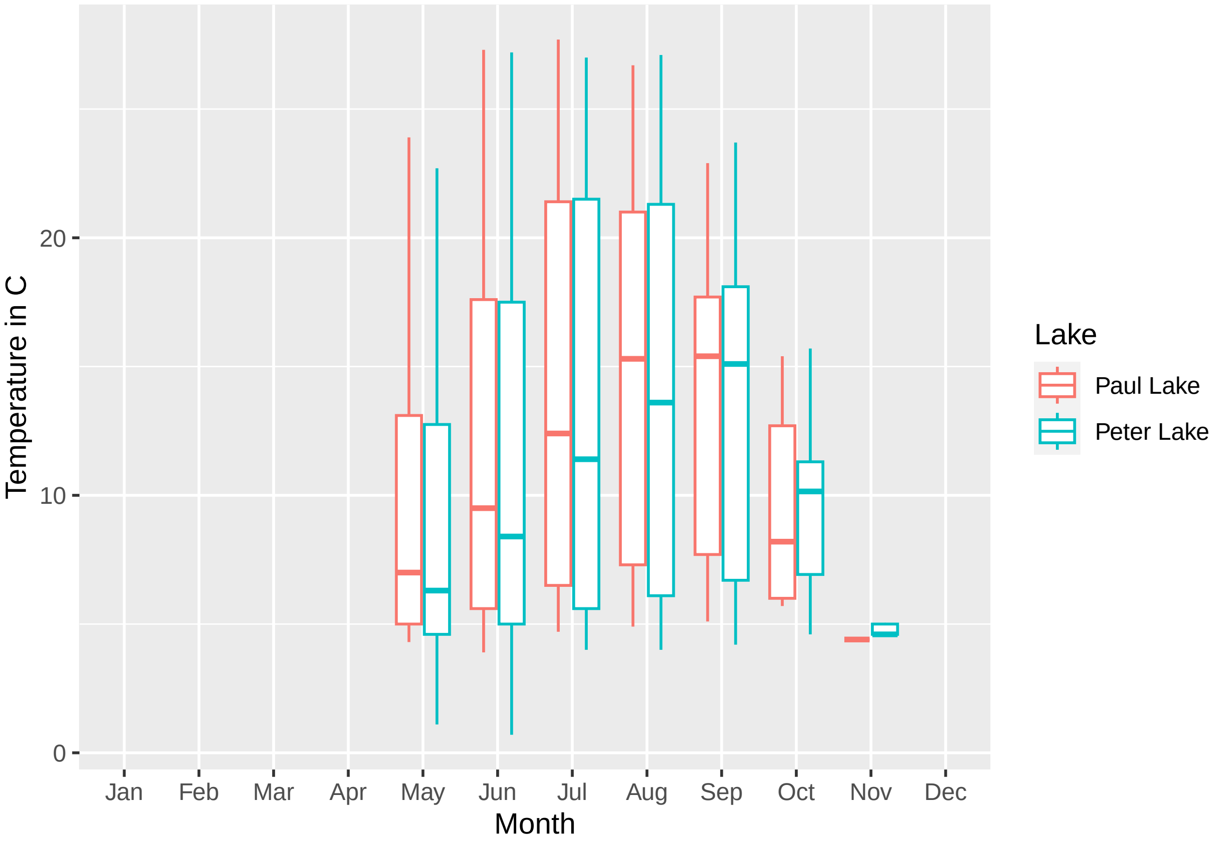


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

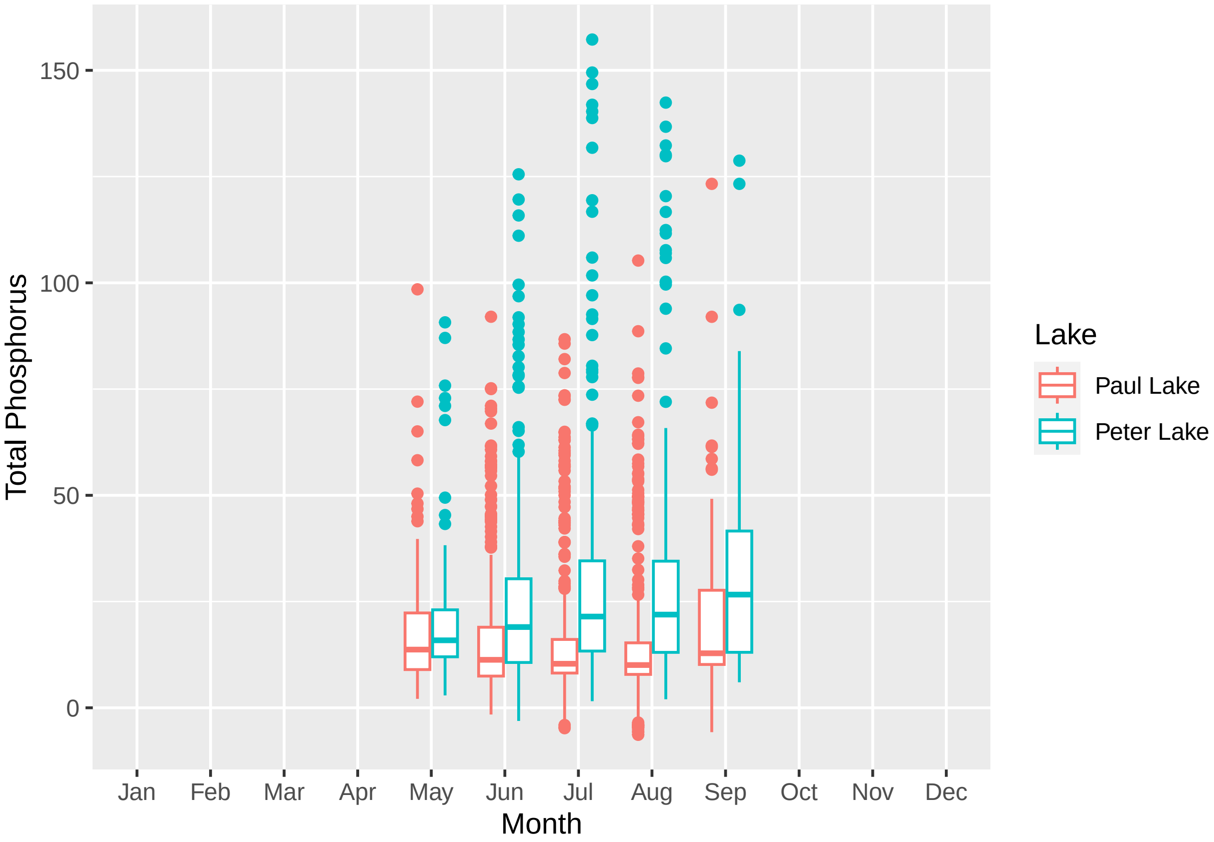
```
#5 3 separate boxplots of temperature, TP, and TN
temperature <- ggplot(NTL_LTER_processed,
       aes(x=factor(month,
                    levels = 1:12,
                    labels = month.abb),
           y = temperature_C,
           color = lakename)) +
  geom_boxplot()+
  scale_x_discrete(name = "Month", drop = FALSE) +
  labs(y = "Temperature in C",
       color = "Lake")
print(temperature)
```

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



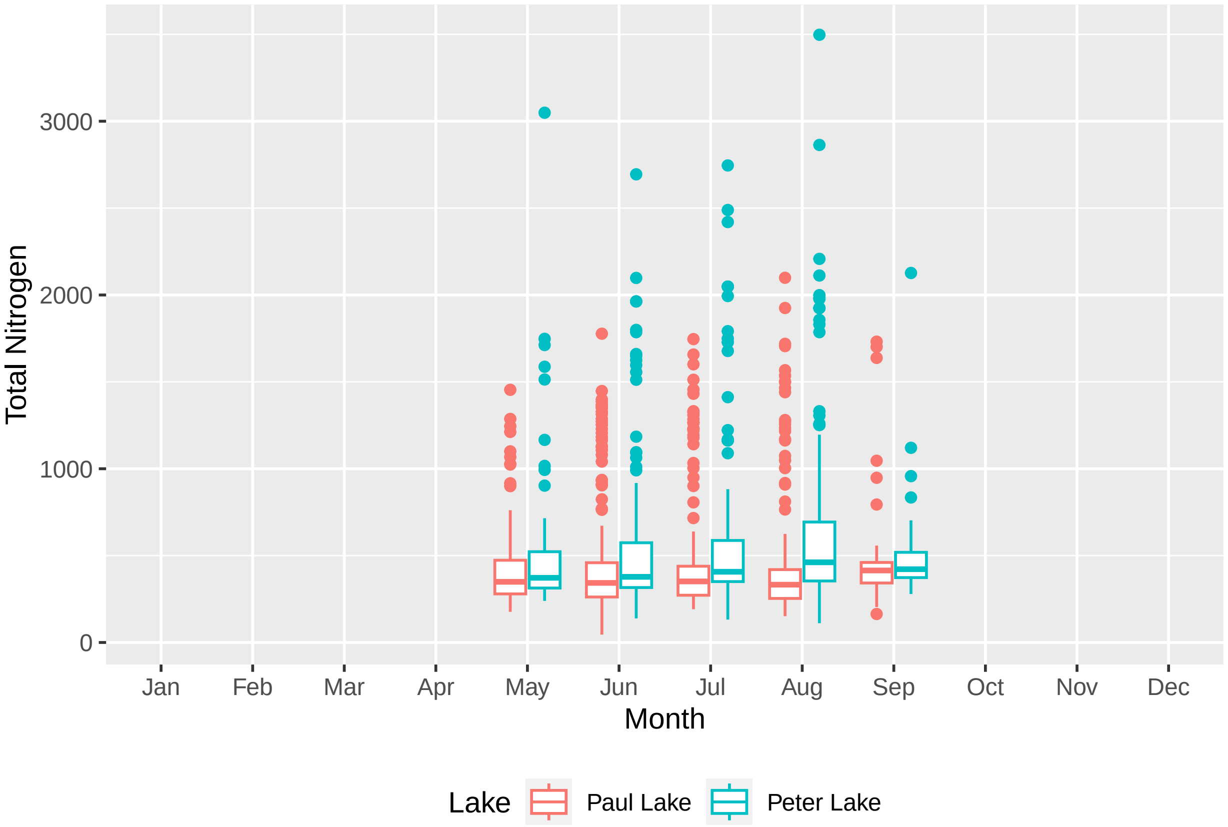
```
TP <- ggplot(NTL_LTER_processed,
       aes(x=factor(month,
                    levels = 1:12,
                    labels = month.abb),
           y = tp_ug, color = lakename)) +
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop = FALSE) +
  labs(y = "Total Phosphorus",
       color = "Lake")
print(TP)
```

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



```
TN <- ggplot(NTL_LTER_processed,
       aes(factor(month,
                    levels = 1:12,
                    labels = month.abb),
           y = tn_ug, color = lakename))+
  geom_boxplot() +
  scale_x_discrete(name = "Month", drop = FALSE) +
  labs(y = "Total Nitrogen",
       color = "Lake")+
  theme(legend.position = "bottom")
print(TN)
```

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



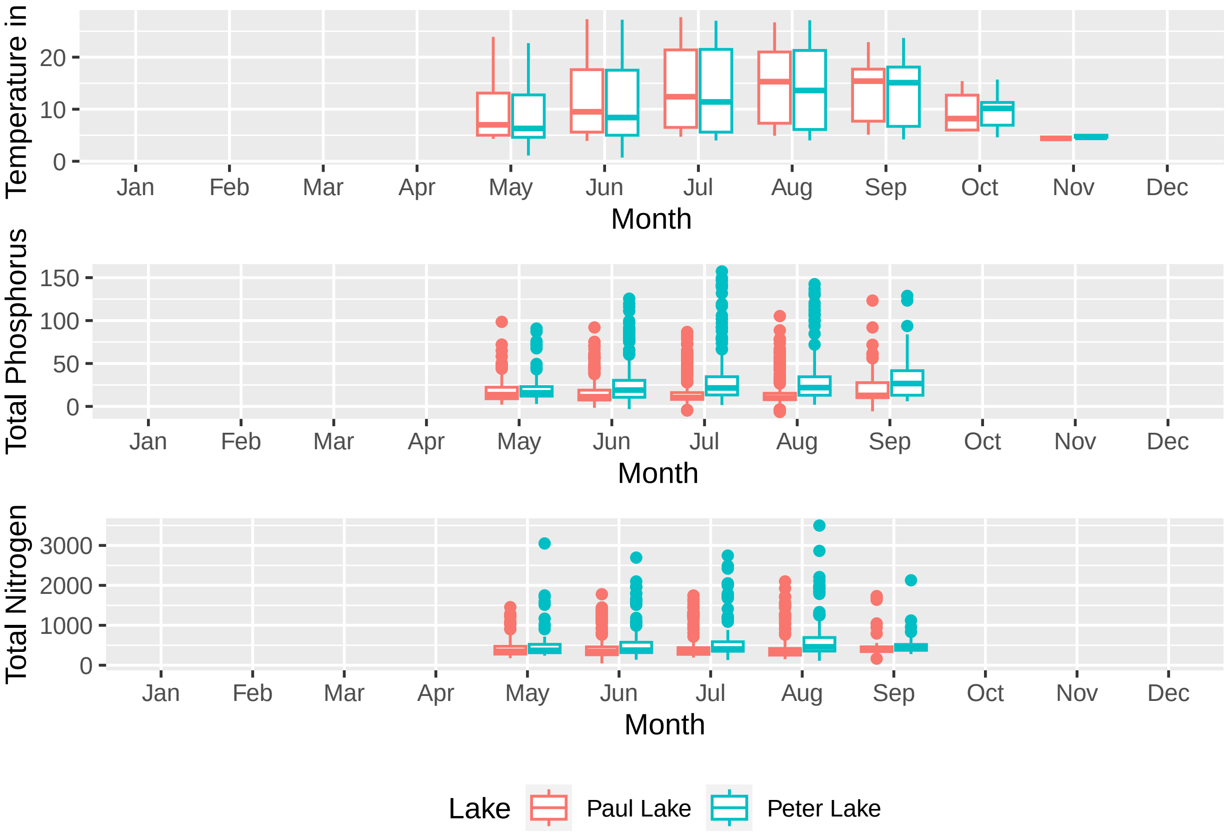
```r
#create a cowplot of all three graphs
plot_grid(temperature +
            theme(legend.position = "none"),
          TP +
            theme(legend.position = "none"),
          TN,
          nrow = 3,
          align = 'h',
          rel_heights = c(1, 1, 1.4))
```

```
## 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.
```

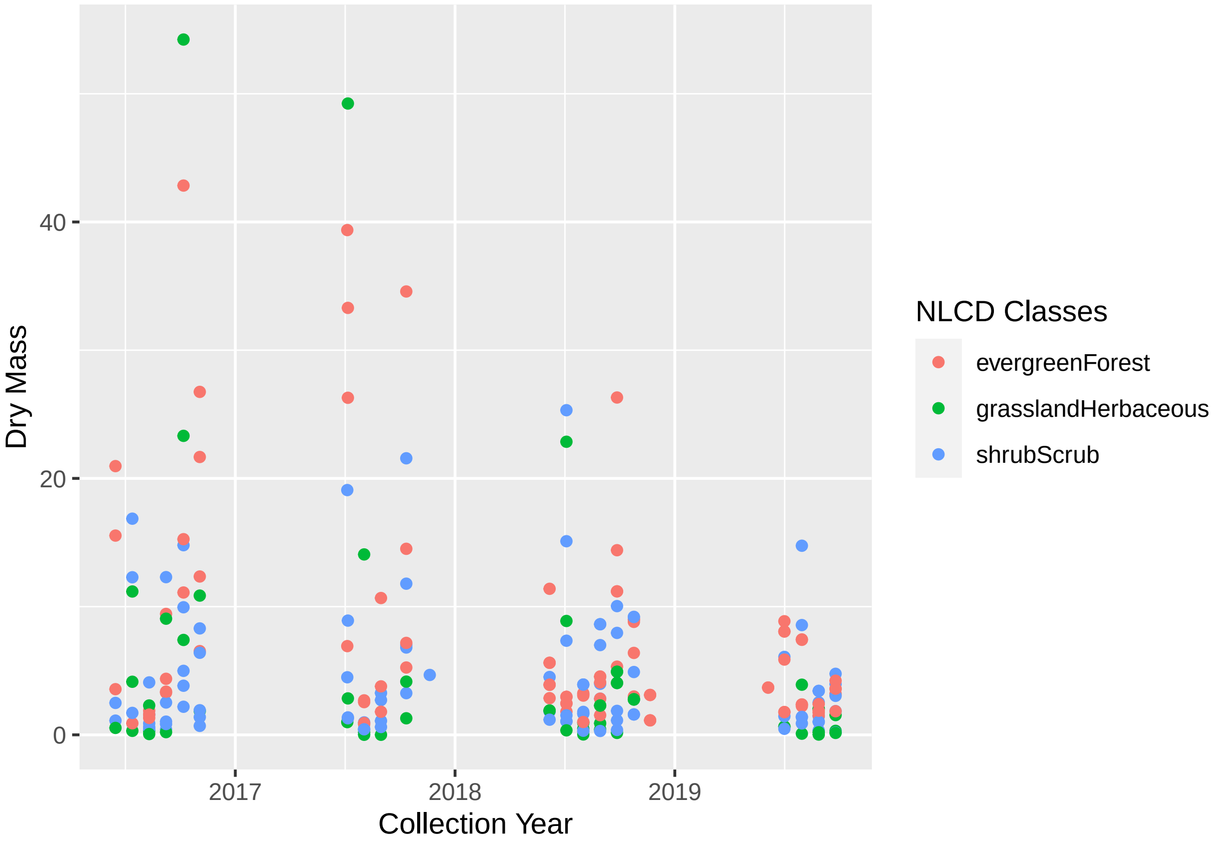


Question: What do you observe about the variables of interest over seasons and between lakes?

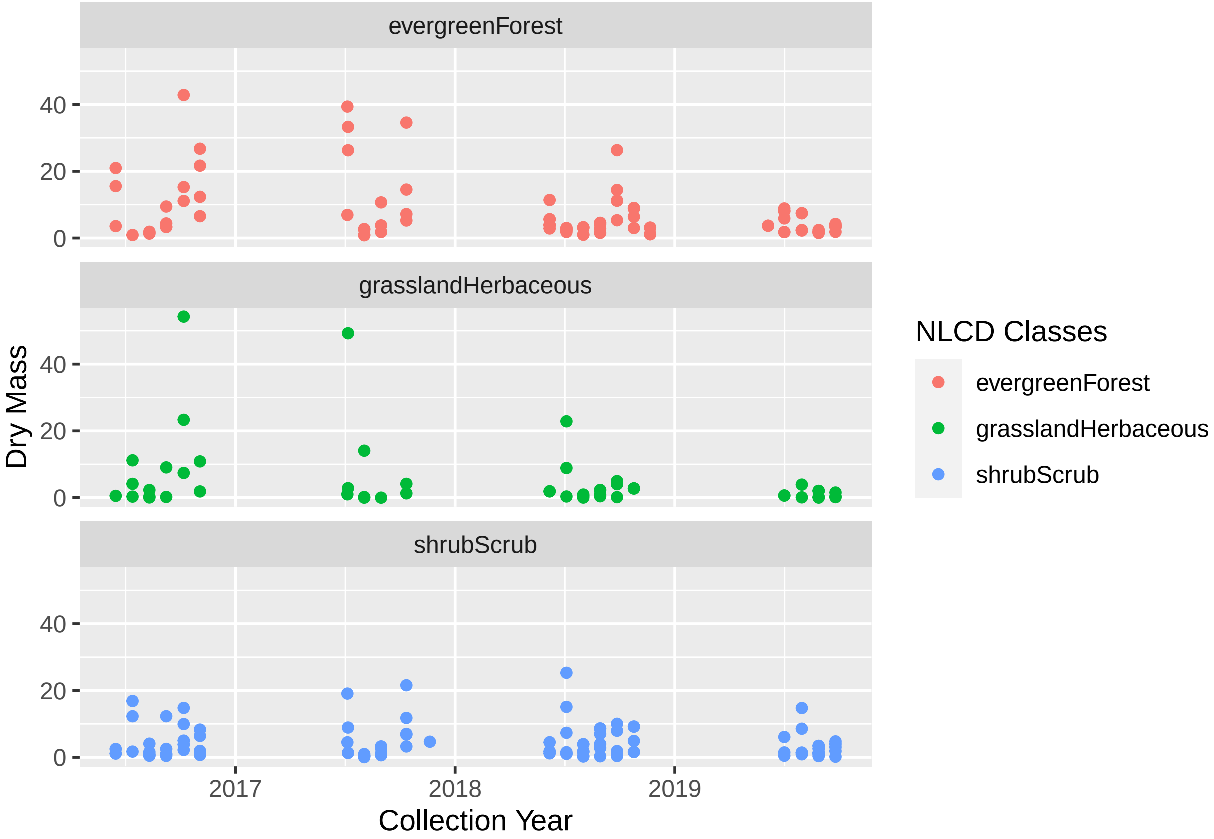
> Answer: Based on the graph, it appears that the mean for temperature, phosphorus and nitrogen all seem to be highest or higher in August-September than in other months. July-September is peak growing season so it would make sense that the most phosphorus and nitrogen are seen in these months. This is also summer, so it would make sense too that the highest temperatures are recorded in these months. Values tend to be at the lowest at the beginning of the May.

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 filter Niwot for Needles functional group
ggplot(filter(Niwot_Ridge_processed,
              functionalGroup == "Needles"),
       aes(x = collectDate, y = dryMass,
           color = nlcdClass)) +
  geom_point()+
  labs(color = "NLCD Classes",
       x = "Collection Year",
       y = "Dry Mass")
```



```
#7 Plot with NLCD classes separated into 3 facets
ggplot(filter(Niwot_Ridge_processed,
              functionalGroup == "Needles"),
       aes(x = collectDate, y = dryMass,
           color = nlcdClass)) +
  geom_point()+
  labs(color = "NLCD Classes",
       x = "Collection Year",
       y = "Dry Mass") +
  facet_wrap(vars(nlcdClass),
             nrow = 3)
```



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

Answer: Plot 7 is definitely more helpful visually because splitting the data into three graphs based on the NLCD classes, allows for patterns or contrasts of dry mass within each vegetation type to become much more clear. When all of the data is plotted on the same axis in the same graph, it becomes very difficult to see what types of inferences can be made about the data.