# Assignment 5: Data Visualization

Cal Oakley

## 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] "K:/GradSchool/Spring2022/EnvironmentalDataAnalytics/Environmental_Data_Analytics_2022/Assignmen
```

```
library(tidyverse) #for plotting
library(cowplot) #for plotting multiple plots simultaneously
library(lubridate) #for reformatting dates later
PeterPaul_ChemNut <- read.csv(
  "../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
NiwotLitter <- read.csv(
  "../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"
) #loading datasets
#2
str(PeterPaul_ChemNut) #to check if dates are read as date format
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : chr  "Paul Lake" "Paul Lake" "Paul Lake" "Paul Lake" ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : chr  "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-27" ...
```

```
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
str(NiwotLitter)
```

```
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : chr  "NIWO_062" "NIWO_061" "NIWO_062" "NIWO_064" ...
##  $ trapID          : chr  "NIWO_062_050" "NIWO_061_169" "NIWO_062_050" "NIWO_064_103" ...
##  $ collectDate     : chr  "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-16" ...
##  $ functionalGroup : chr  "Seeds" "Other" "Woody material" "Seeds" ...
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : chr  "N" "N" "N" "N" ...
##  $ subplotID       : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : chr  "shrubScrub" "evergreenForest" "shrubScrub" "evergreenForest" ...
##  $ plotType        : chr  "tower" "tower" "tower" "tower" ...
##  $ geodeticDatum   : chr  "WGS84" "WGS84" "WGS84" "WGS84" ...
```

```
PeterPaul_ChemNut$sampledate  <- ymd(PeterPaul_ChemNut$sampledate)
NiwotLitter$collectDate <- ymd(NiwotLitter$collectDate) #reformatting dates
```

## Define your theme

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

```
#3
CalsTheme <- theme_classic(base_size = 16) +
  theme(axis.text = element_text(color = "gray"), legend.position = "left",
        legend.justification = 2)
theme_set(CalsTheme)
```

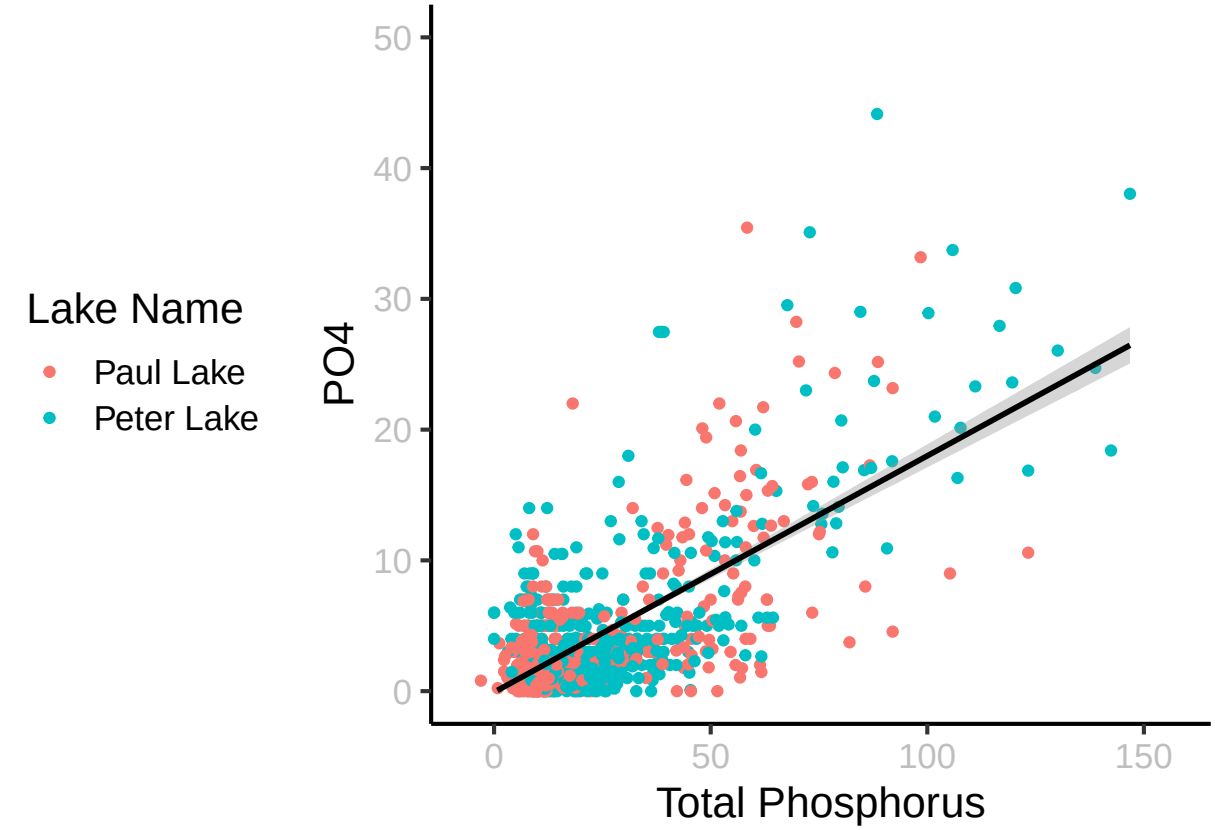
## Create graphs

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
A05_plot1 <- ggplot(PeterPaul_ChemNut, aes(x = tp_ug, y = po4,
                                           color = lakename)) +
  geom_point() +
  ylim(0, 50) +
  labs(x = "Total Phosphorus", y = "PO4", color = "Lake Name") +
```

```
  geom_smooth(method = lm, color = "black")
print(A05_plot1)
```



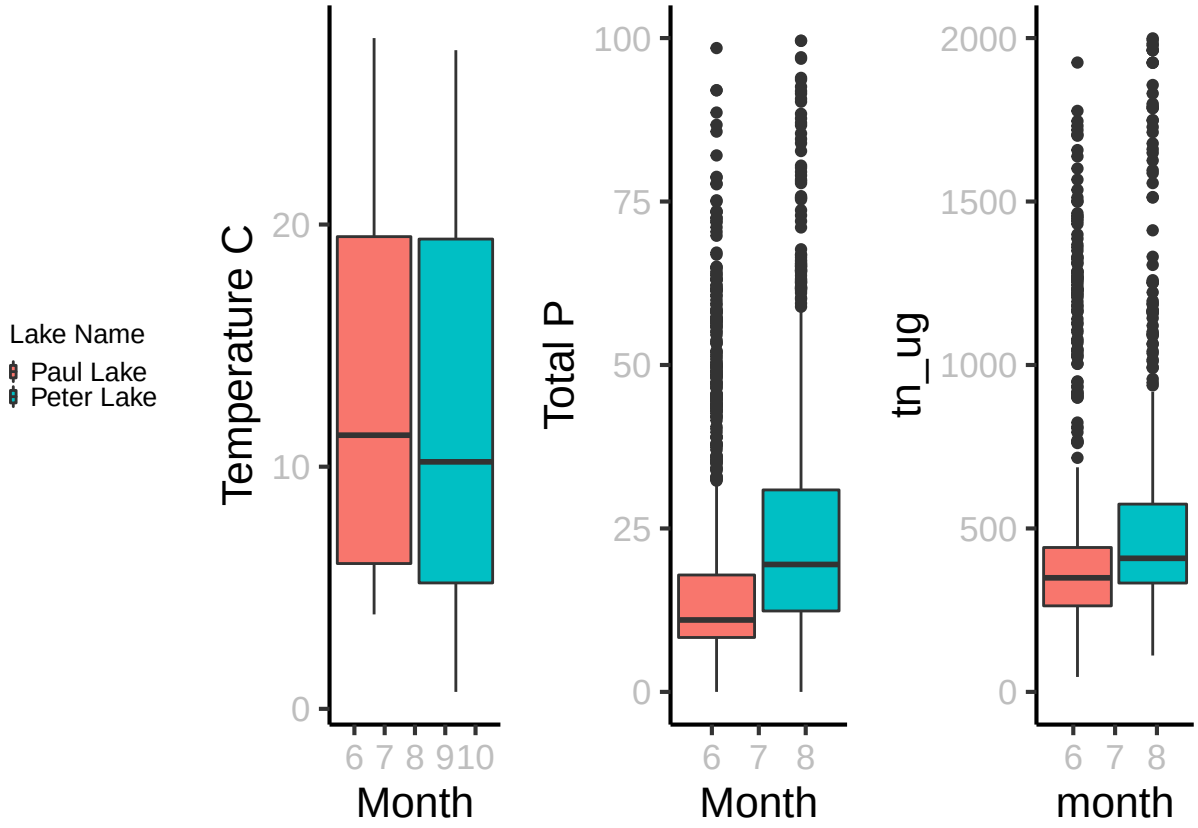

5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
#5
A05_Plot2 <- ggplot(PeterPaul_ChemNut, aes(x = month, y = temperature_C,
                    fill = lakename)) +
  geom_boxplot() +
  labs(fill = "Lake Name", x = "Month", y = "Temperature C") +
  theme(legend.key.size = unit(0.1, "cm"),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10))

A05_Plot3 <- ggplot(PeterPaul_ChemNut, aes(x = month, y = tp_ug,
                    fill = lakename)) +
  geom_boxplot() +
  ylim(0,100) +
  labs(x = "Month", y = "Total P")

A05_Plot4 <- ggplot(PeterPaul_ChemNut, aes(x = month, y = tn_ug,
                    fill = lakename)) +
  geom_boxplot() +
  ylim(0, 2000)
```

```
  labs(x = "Month", y = "Total N")
```

```
## $x
## [1] "Month"
##
## $y
## [1] "Total N"
##
## attr(,"class")
## [1] "labels"
```

```
A05_cowplot <- plot_grid(A05_Plot2,
                         A05_Plot3 + theme(legend.position = "none"),
          A05_Plot4 + theme(legend.position = "none"),
          #+ theme(legend.position = "none") removes redunant legends from plot
          ncol = 3, align = 'h', rel_widths = c(.4, .25, .25))
print(A05_cowplot)
```



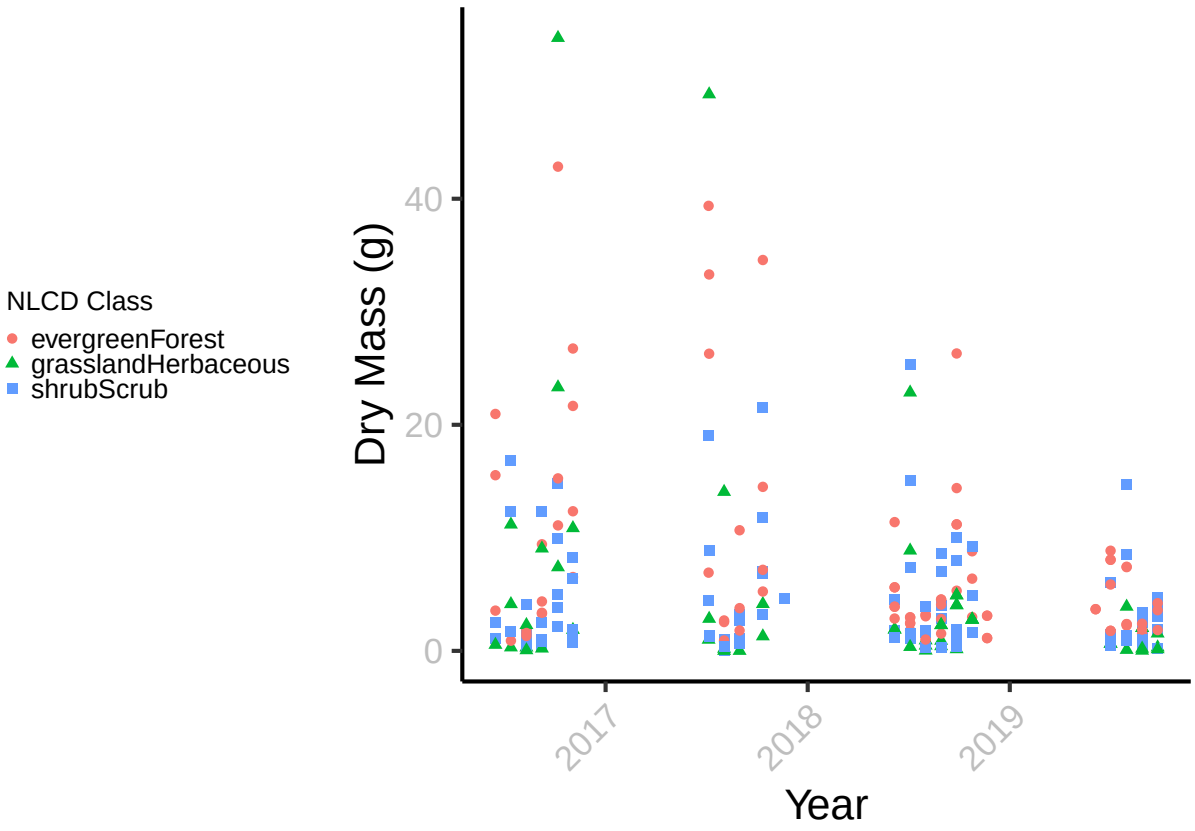

Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Temperatures in Paul Lake during early Summer are essentially the same as the temperatures in Peter Lake in late Summer/early Fall. Meanwhile the total Phosphorus and Nitrogen increase in Peter Lake later in the summer when compared with Paul Lake.

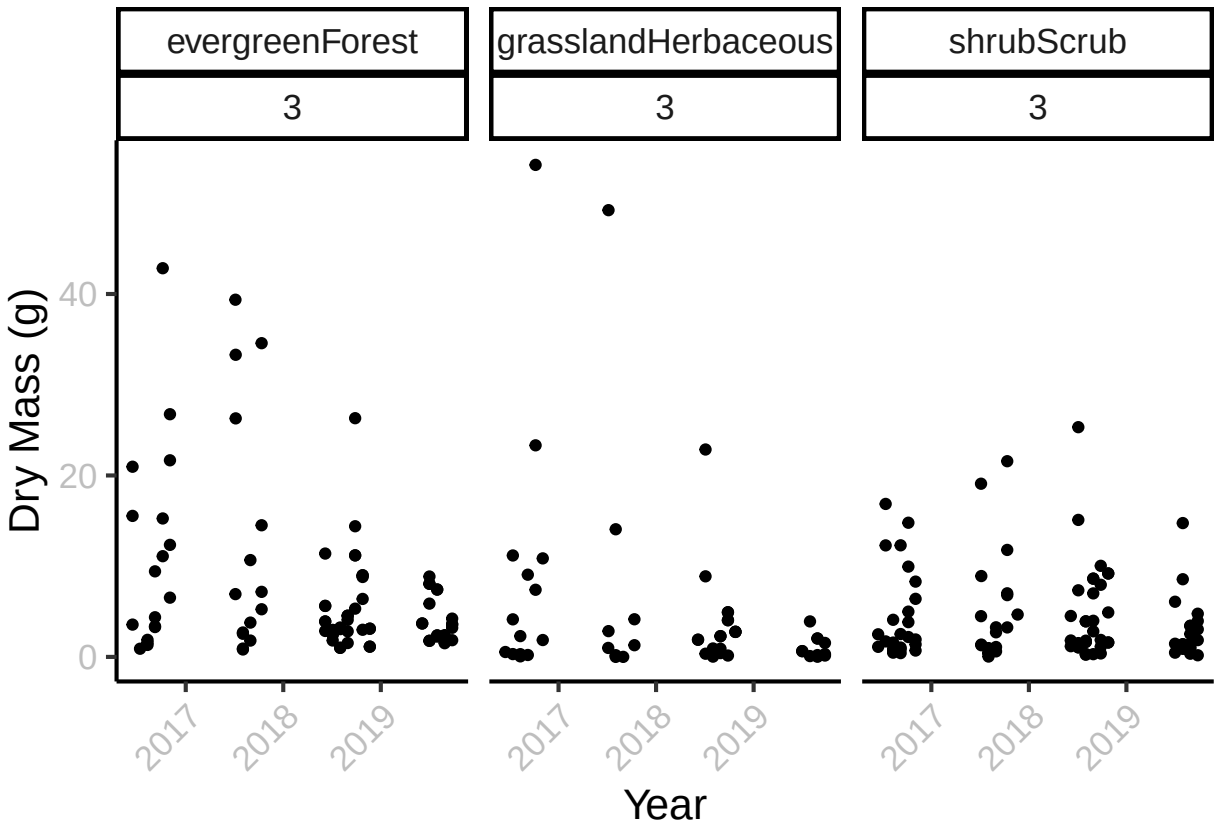
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
# A05_plot5 <- ggplot(subset(NiwotLitter, format.Date(collectDate, "%m") >= 06 &
#                                 format.Date(collectDate, "%m") <= 09,
#                               functionalGroup == "Needles")) +
#                         geom_point(aes(x = collectDate, y = subplotID,
#                                        color = nlcdClass,
#                                        shape = nlcdClass)) +
#   scale_x_date(date_breaks = "1 month", date_labels = "%m-%Y") +
#   theme(axis.text.x = element_text(angle = 45,  hjust = 1))
#above is my first attempt at solving this, I realized that there were many
#months for which data didn't exist, but I couldn't find a way to subset dates
#by just month. Each time I did this it said that it couldn't find my y variable.
#My subsetting must be confusing ggplot into thinking there is no data
#I settled for the code below because I think it meets the requirements
A05_plot5 <- ggplot(subset(NiwotLitter, functionalGroup == "Needles")) +
                      geom_point(aes(x = collectDate, y = dryMass,
                                     color = nlcdClass,
                                     shape = nlcdClass)) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(x = "Year", y = "Dry Mass (g)", color = 'NLCD Class',
       shape = 'NLCD Class') +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1),
        legend.key.size = unit(0.1, "cm"),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10))
print(A05_plot5)
```




```
#7
A05_plot6 <- ggplot(subset(NiwotLitter, functionalGroup == "Needles")) +
                      geom_point(aes(x = collectDate, y = dryMass)) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(x = "Year", y = "Dry Mass (g)", color = 'Year') +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1)) +
  facet_wrap(vars(nlcdClass, ncol = 3))
print(A05_plot6)
```




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

> Answer: Originally I thought 6 was more effective because 7 didn't display the dates as well, but then I tried knitting and realized 6 didn't deal with dates much better. So I think 7 is better, because although it has the same issues with dates (where their ticks are positioned and their poor resolution) that 6 does, it is very clear which samples are from which NLCD Classes.