Assignment 5: Data Visualization

Rachel Schoenecker

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] "C:/Users/rscho/OneDrive/Documents/MEM Courses/Sem2/ENV 872/Environmental_Data_Analytics_2022/As
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
## Warning: package 'readr' was built under R version 4.1.1
```

```
## Warning: package 'purrr' was built under R version 4.1.1
```

```
## Warning: package 'forcats' was built under R version 4.1.1
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.1.2
```

```
PP.Nutrients <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
Niwot.Litter <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(PP.Nutrients$sampledate)
```

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

```
PP.Nutrients$sampledate <- as.Date(PP.Nutrients$sampledate, format = "%Y-%m-%d")
class(PP.Nutrients$sampledate)
```

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

```
class(Niwot.Litter$collectDate)
```

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

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

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

Define your theme

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

```
#3
mytheme <- theme_bw(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        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.

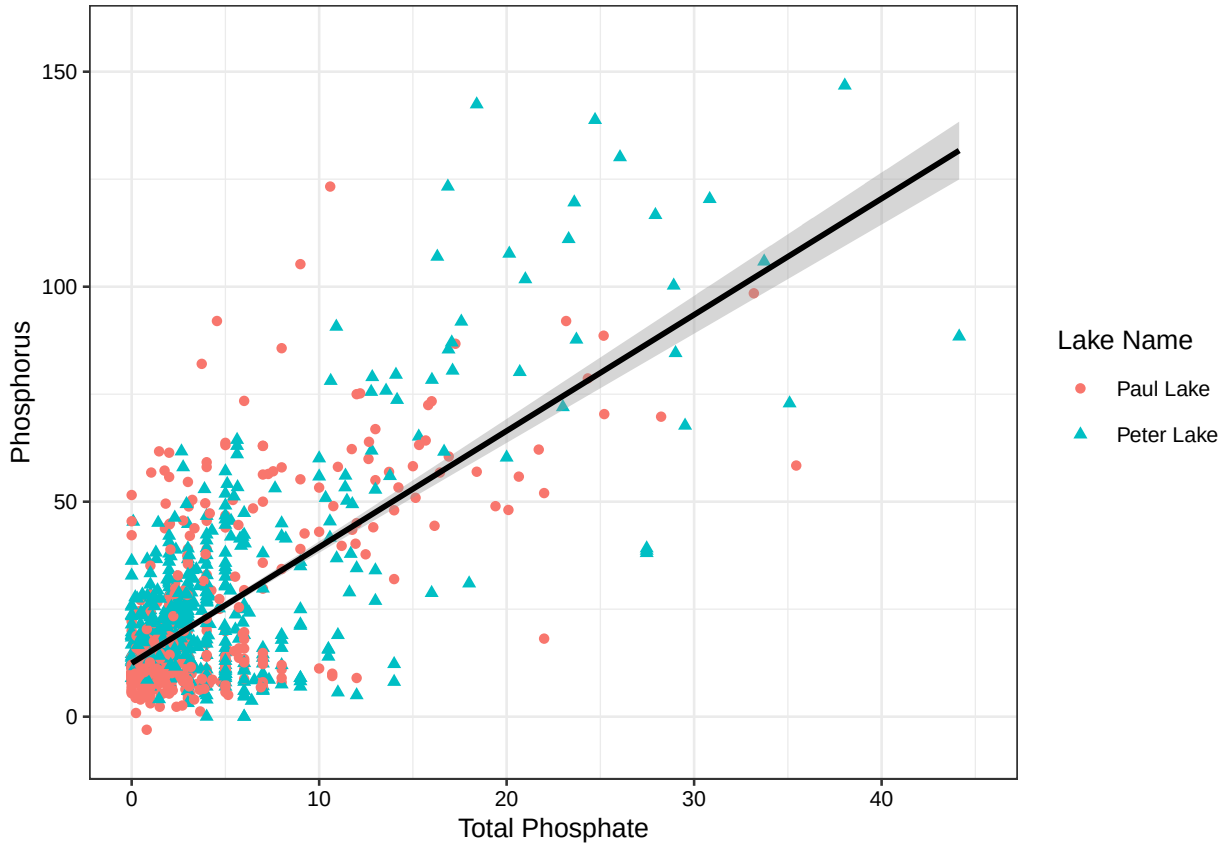
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and `ylim()`).

```
#4
lake.plot1 <- ggplot(PP.Nutrients, aes(x = po4, y = tp_ug)) +
  geom_point(aes(color = lakename, shape = lakename)) +
  geom_smooth(method = lm, col = "black") + xlim(0,45) +
  labs(x = "Total Phosphate", y = "Phosphorus",
       color = "Lake Name", shape = "Lake Name")
print(lake.plot1)
```

```
## `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).
```



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
lake.plot.temp <- ggplot(PP.Nutrients[PP.Nutrients$month != 2,],
                         aes(x = as.factor(month), y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  labs(x = NULL, y = "Temperature, C", color = "Lake Name")

lake.plot.tp <- ggplot(PP.Nutrients[PP.Nutrients$month != 2,],
                       aes(x = as.factor(month), y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(x = NULL, y = "Total Phosphorus", color = "Lake Name")

lake.plot.tn <- ggplot(PP.Nutrients[PP.Nutrients$month != 2,],
                       aes(x = as.factor(month), y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(x = "Month", y = "Total Nitrogen", color = "Lake Name")

legend <- get_legend(lake.plot.temp)
```

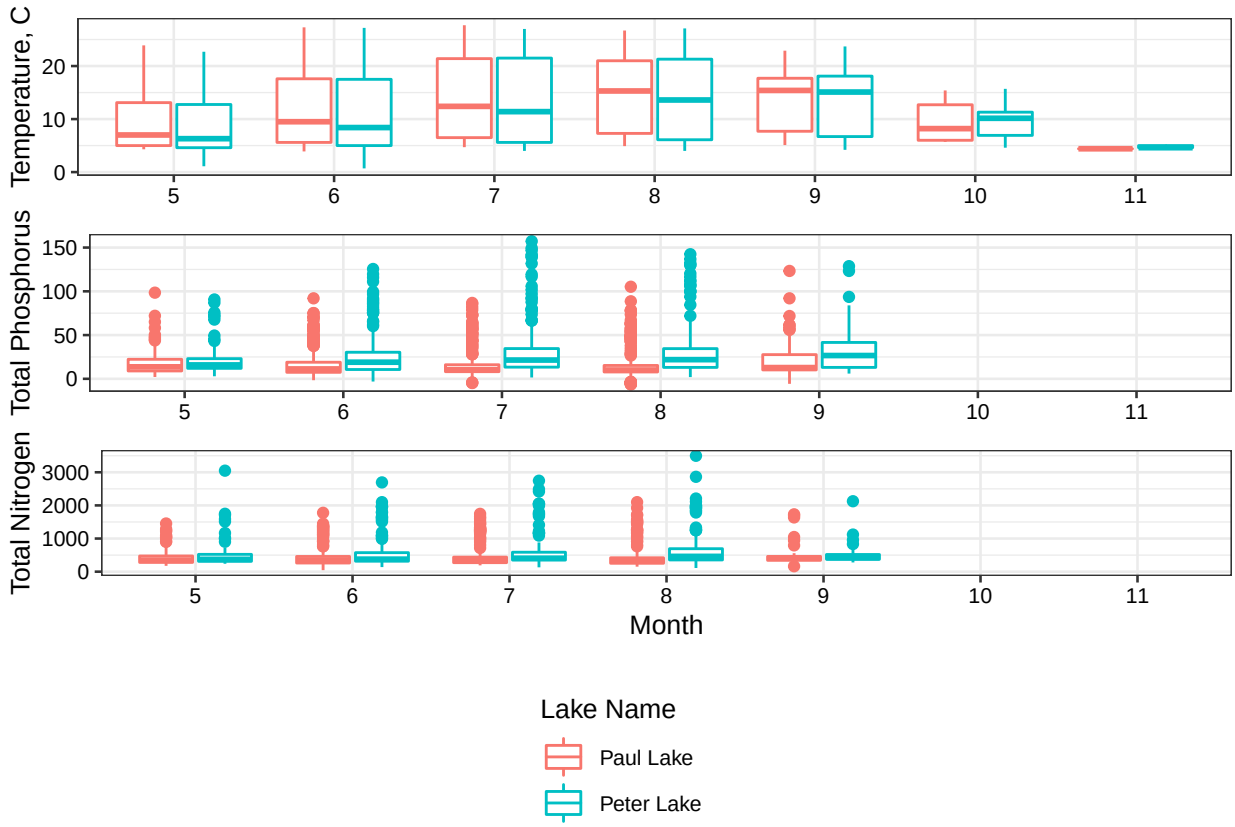
```
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```

```
lake.plots <- plot_grid(lake.plot.temp + theme(legend.position = "none"),
                        lake.plot.tp + theme(legend.position = "none"),
```

```
                        lake.plot.tn + theme(legend.position = "none"), legend,
                        align = "h", nrow = 4)
```

```
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
## Warning: Removed 20713 rows containing non-finite values (stat_boxplot).
## Warning: Removed 21567 rows containing non-finite values (stat_boxplot).
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
print(lake.plots)
```



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

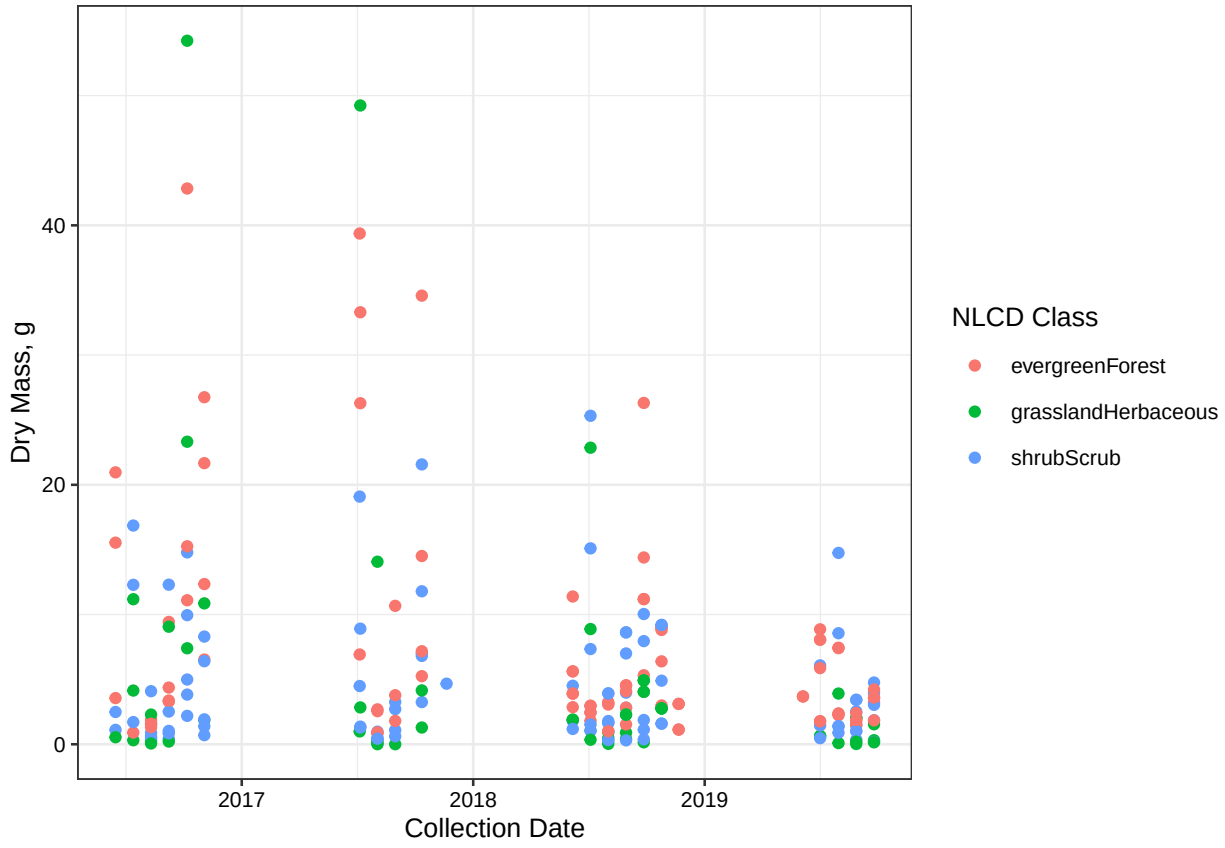
Answer: All three variables of interest appear to increase in the summer months (specifically July and August), and while temperature seems to be fairly similar between the lakes, Phosphorus and Nitrogen appear to be slightly higher in Peter Lake than Paul Lake.

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)

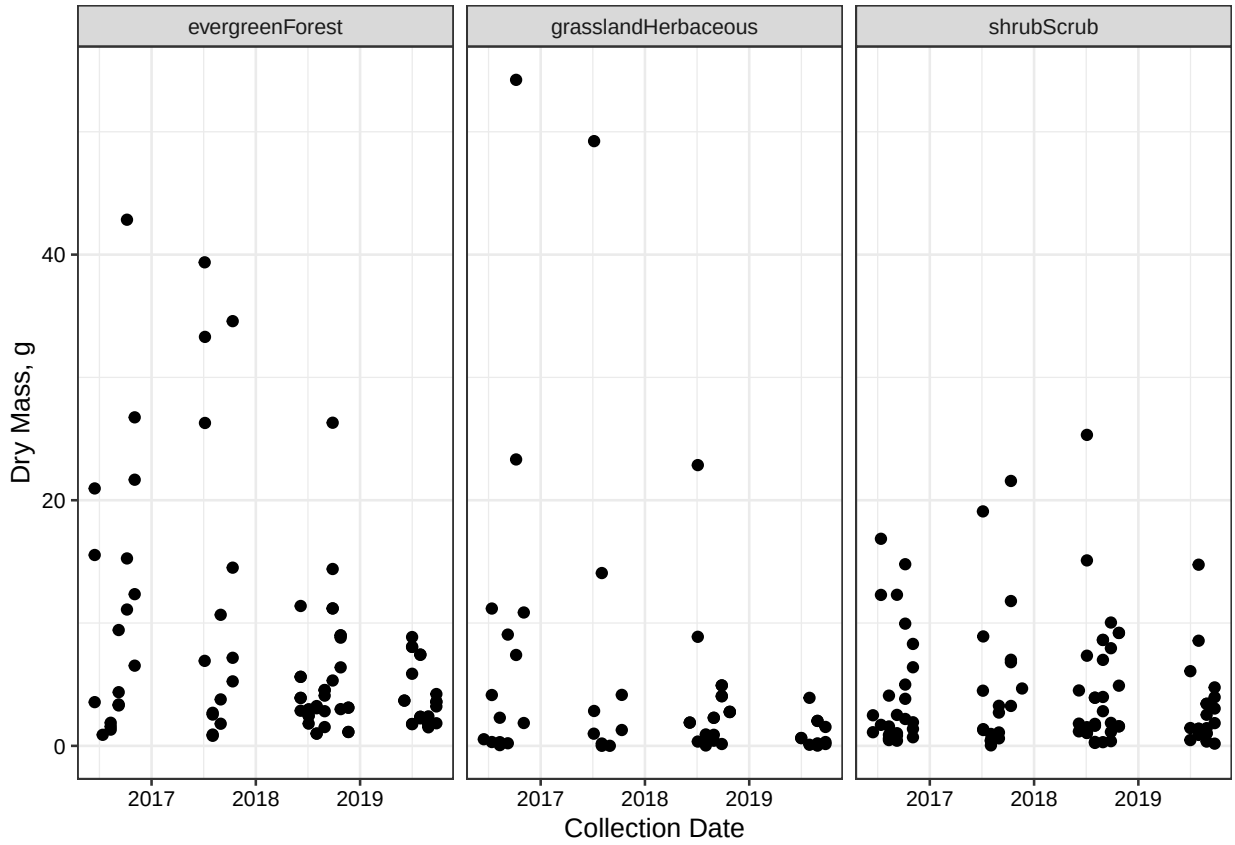
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
litter.comp <-
  ggplot(Niwot.Litter[Niwot.Litter$functionalGroup == "Needles",]) +
```

```
  geom_point(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  labs(x = "Collection Date", y = "Dry Mass, g", color = "NLCD Class")
print(litter.comp)
```




```
#7
litter.sep <-
  ggplot(Niwot.Litter[Niwot.Litter$functionalGroup == "Needles",]) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  labs(x = "Collection Date", y = "Dry Mass, g") +
  facet_wrap("nlcdClass")
print(litter.sep)
```



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

Answer: I think plot 7 where the classes are separated is more effective. Although the classes are color-coded in plot 6, it's hard to pick up on any trends within each class because there are so many points in the same plot. The only trend I identified was that litter mass seemed to be decreasing overall. However, when the different classes were separated out, it was easier to see their individual trends. Plot 7 showed that while litter mass in forests and grasslands does appear to be decreasing, it is remaining somewhat more stable in shrub/scrub cover.