Assignment 5: Data Visualization

Sophia Bryson

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
  #verify working directory
  getwd()
```

```
## [1] "Z:/ENV872/Environmental_Data_Analytics_2022"
```

```
  #load packages
  library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
  library(cowplot)
  library(wesanderson) #for the pretty colors

  #load in datasets
```

```
  PPLakesData <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                          stringsAsFactors = TRUE)
  NRLitterData <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                           stringsAsFactors = TRUE)

#2
  #check if dates are formatted correctly
  class(PPLakesData$sampledate); class(NRLitterData$collectDate)
```

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

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

```
  #they're not, so fix them
  PPLakesData <- PPLakesData %>% mutate(sampledate = as.Date(sampledate,
                                                             format = "%Y-%m-%d"))
  NRLitterData <- NRLitterData %>% mutate(collectDate = as.Date(collectDate,
                                                                format = "%Y-%m-%d"))
```

Define your theme

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

```
#3
  #Build - modify 'minimal'
  custom_theme <- theme_minimal(base_size = 11, base_family = "sans") +
                  theme(panel.background = element_rect(fill = "ivory2"),
                        panel.grid.major = element_line(color = "ivory3"),
                        panel.grid.minor = element_line(color = "ivory"),
                        axis.text = element_text(color = "ivory4"),
                        legend.position = "bottom")

  #Set as default
  theme_set(custom_theme)
```

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.

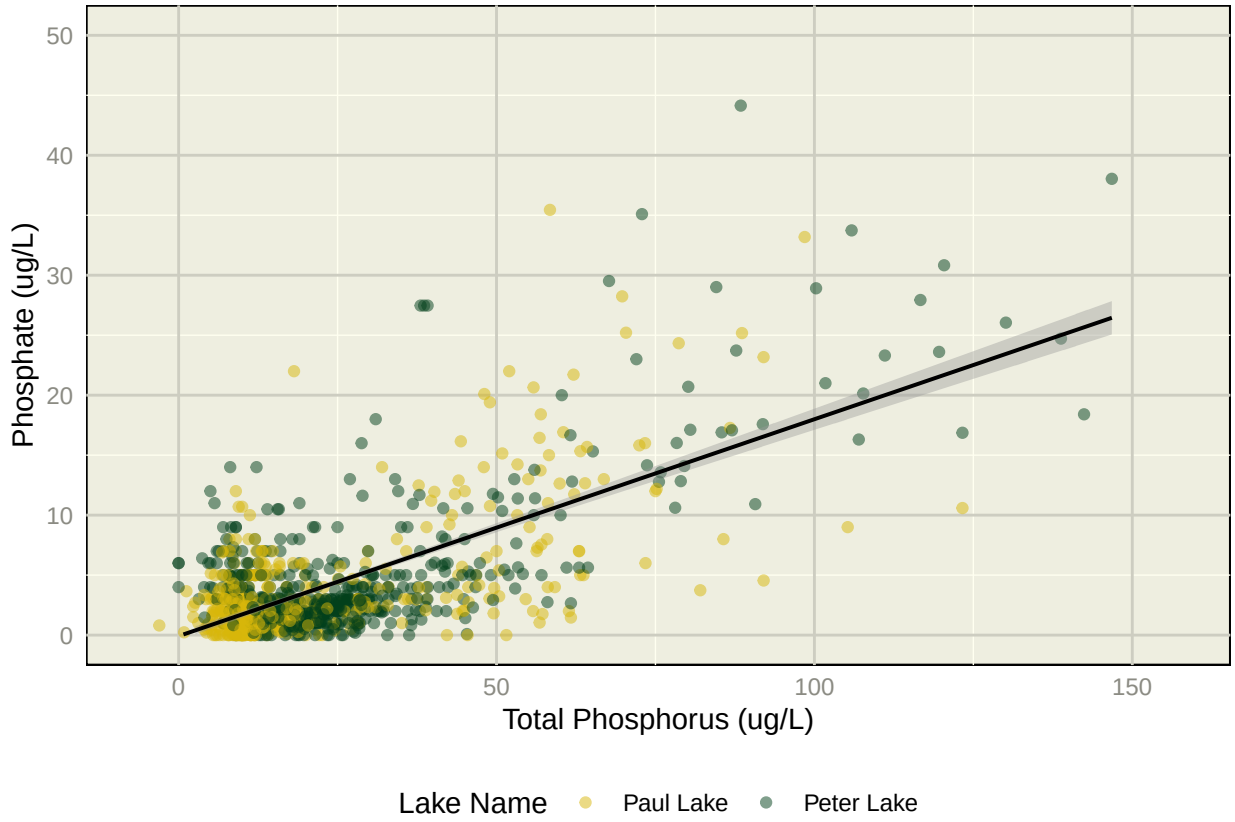
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and `ylim()`).

```
#4
Lakes_P_plot <- ggplot(data = PPLakesData, aes(x = tp_ug, y = po4, color = lakename)) +
                       geom_point(alpha = 0.5) +
                       geom_smooth(method = "lm", color = "black", size = .7) +
                       ylim(0, 50) +
                       labs(x = "Total Phosphorus (ug/L)", y = "Phosphate (ug/L)",
                            color = "Lake Name") +
                       scale_color_manual(values = wes_palettes$Cavalcanti1)

Lakes_P_plot
```

```
## `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.

```
#5
Lakes_temp_plot <- ggplot(data = PPLakesData, aes(x = as.factor(month),
                                                  y = temperature_C,
                                                  color = lakename)) +
                          geom_boxplot(alpha = 0.7) +
                          labs(x = "", y = "Temp(C)", color = "Lake Name") + #keep the yaxis name short
                          scale_color_manual(values = wes_palettes$Cavalcanti1) +
                          theme(legend.position = "none")


Lakes_TP_plot <- ggplot(data = PPLakesData, aes(x = as.factor(month),
                                                y = tp_ug,
                                                color = lakename)) +
                          geom_boxplot(alpha = 0.7) +
                          labs(x = "", y = "TP (ug/L)", color = "Lake Name") + #keep the yaxis name sho
                          scale_color_manual(values = wes_palettes$Cavalcanti1) +
                          theme(legend.position = "none")
```

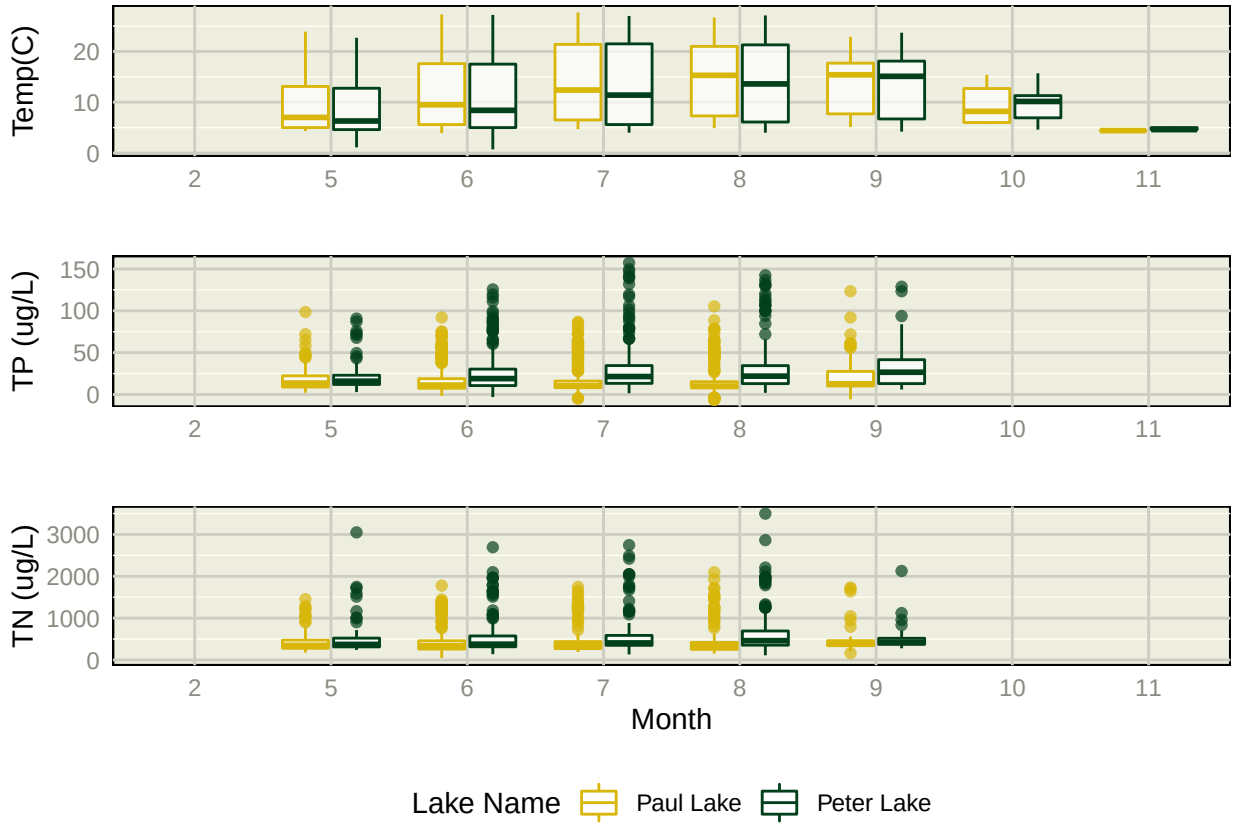
```
Lakes_TN_plot <- ggplot(data = PPLakesData, aes(x = as.factor(month),
                                                y = tn_ug,
                                                color = lakename)) +
                          geom_boxplot(alpha = 0.7) +
                          labs(x = "Month", y = "TN (ug/L)", color = "Lake Name") + #keep the yaxis nam
                          scale_color_manual(values = wes_palettes$Cavalcanti1)

plot_grid(Lakes_temp_plot, Lakes_TP_plot, Lakes_TN_plot, nrow = 3, align = 'v',
          rel_heights = c(1, 1, 1.45))
```

```
## 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).
```

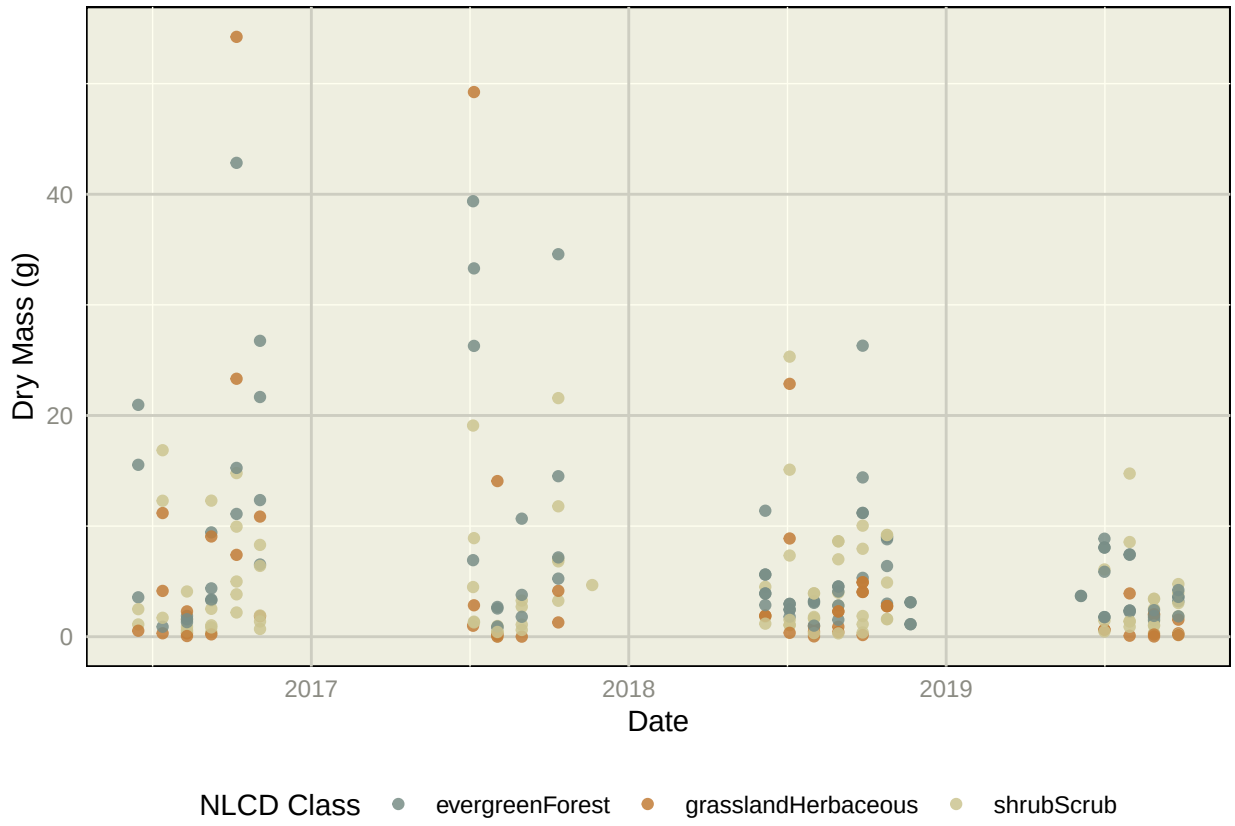


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: In both of the lakes, temperature rises during the summer months (unsurprisingly), though it also has the greatest variablility during these months. In Peter Lake, both TP and TN also rise and increase in variability in the summer months. For Paul lake, both TP and TN decrease slightly during the summer months, though the summer variability is still greater than spring and fall.

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)

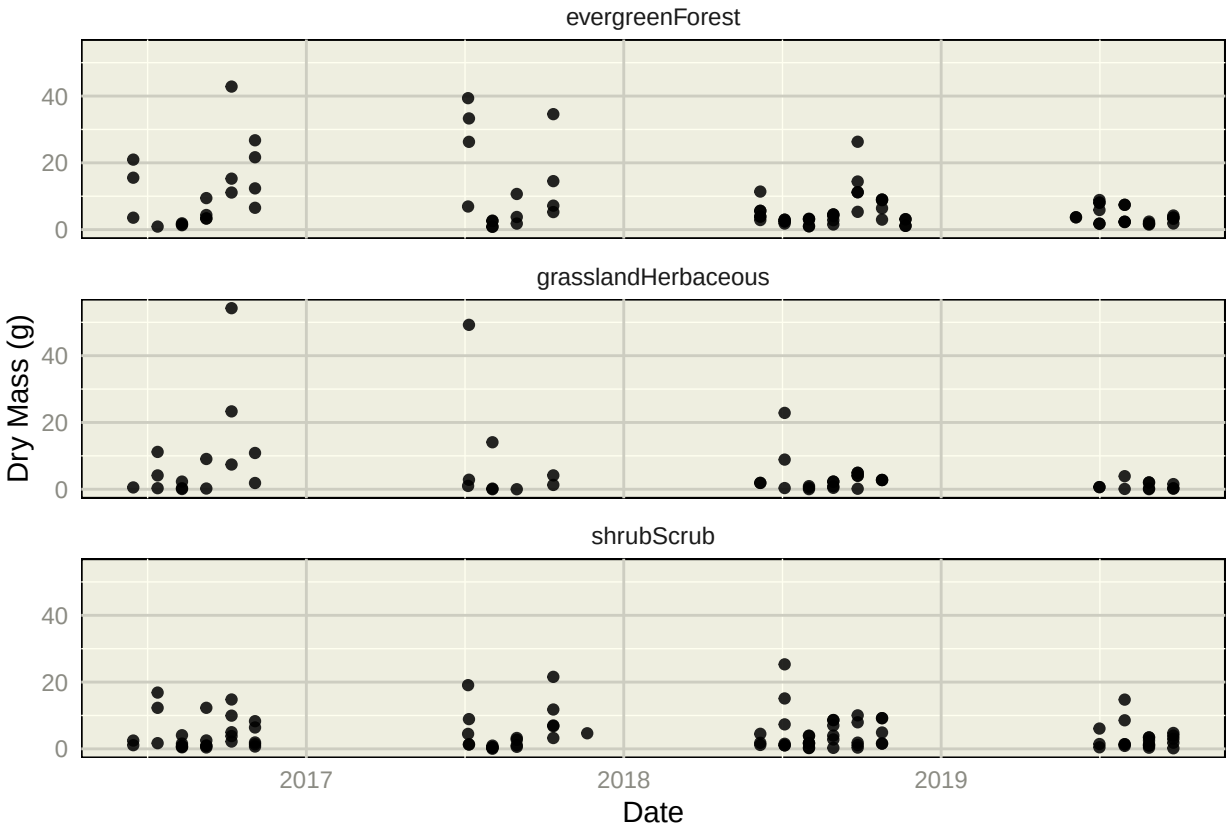
```
#6
NeedleLitter_color_plot <- ggplot(data = subset(NRLitterData, functionalGroup == "Needles"),
                                  aes(x = collectDate, y = dryMass, color = nlcdClass)) +
                                  geom_point(alpha = 0.85) +
                                  labs(x = "Date", y = "Dry Mass (g)", color = "NLCD Class") +
                                  scale_color_manual(values = wes_palettes$Moonrise2)
NeedleLitter_color_plot
```



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

```
#7
NeedleLitter_facet_plot <- ggplot(data = subset(NRLitterData, functionalGroup == "Needles"),
                                  aes(x = collectDate, y = dryMass)) +
                                  geom_point(alpha = 0.85) +
                                  labs(x = "Date", y = "Dry Mass (g)") +
                                  facet_wrap(facets = "nlcdClass", nrow = 3)

NeedleLitter_facet_plot
```



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

> Answer: I think 7 more effectively allows the viewer to distinguish between the classes - the colors in 6 all overlap so as to obscure the distinct classes, and only outliers are really visible. The fact wrap allows for more rapid visual comparison between and across classes by parsing them out more cleanly.