Assignment 5: Data Visualization

Elizabeth Good

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.

```
#1
# load in needed packages
library(tidyverse)
```

```
## -- 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(lubridate)
library(here)
```

```
## here() starts at C:/Users/goode/OneDrive/Documents/Duke/ENV872_EDE/EDE_Fall2023
```

```
library(cowplot)
```

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

```
# verify home directory
here()
```

```
## [1] "C:/Users/goode/OneDrive/Documents/Duke/ENV872_EDE/EDE_Fall2023"
```

```
# read in processed data
PeterPaul.chem.nutrients <-
  read.csv(
    here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)

NEON.NIWO.litter <-
  read.csv(
    here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)

#2
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(NEON.NIWO.litter$collectDate)
```

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

```
# they are currently all of class factor

PeterPaul.chem.nutrients$sampledate <-
  ymd(PeterPaul.chem.nutrients$sampledate)
NEON.NIWO.litter$collectDate <-
  ymd(NEON.NIWO.litter$collectDate)

class(PeterPaul.chem.nutrients$sampledate)
```

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

```r
class(NEON.NIWO.litter$collectDate)
```

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

```r
# now they are all of class "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

```r
#3
new_theme <- theme_bw() +
  theme(axis.text = element_text(color = "navy",
                                 size = 12),
        axis.title = element_text(color = "gray40",
                                  size = 12),
        plot.title = element_text(color = "gray40",
                                  face = "bold",
                                  hjust = 0.5),
        legend.position = "bottom")

theme_set(new_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/or `ylim()`).

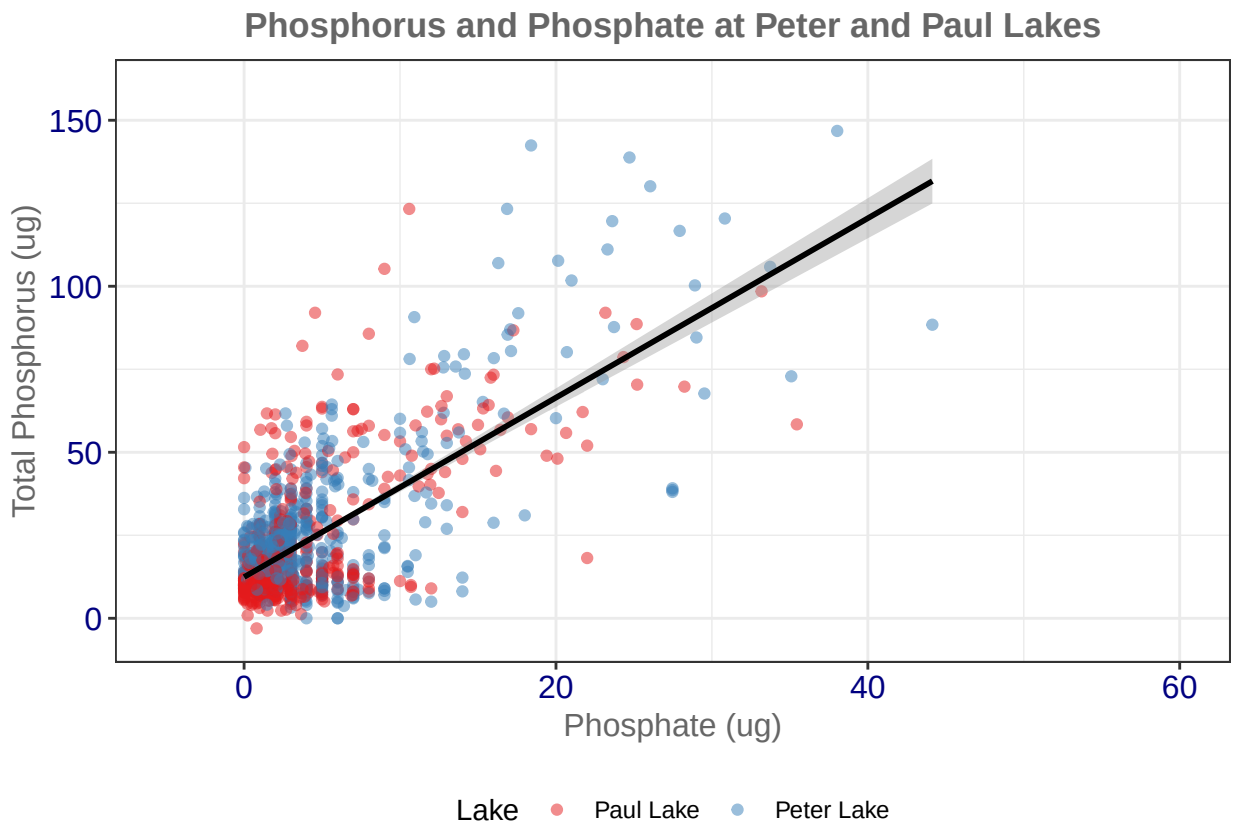
```r
#4
phosphorus_phosphate_plot <- PeterPaul.chem.nutrients %>%
  ggplot(mapping = aes(x = po4,
                       y = tp_ug,
                       color = lakename)) +
  ggtitle("Phosphorus and Phosphate at Peter and Paul Lakes") +
  scale_y_continuous(name = "Total Phosphorus (ug)",
                     limits = (c(-5,160))) +
  scale_x_continuous(name = "Phosphate (ug)",
                     limits = (c(-5,60))) +
  labs(color = 'Lake') +
  geom_point(alpha = 0.5) +
```

```
  scale_color_brewer(palette = "Set1") +
  geom_smooth(method = lm, color = "black")

print(phosphorus_phosphate_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()`).
```



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
a <-
  ggplot(PeterPaul.chem.nutrients,
           aes(x = factor(month,
                          levels = c(1:12),
```

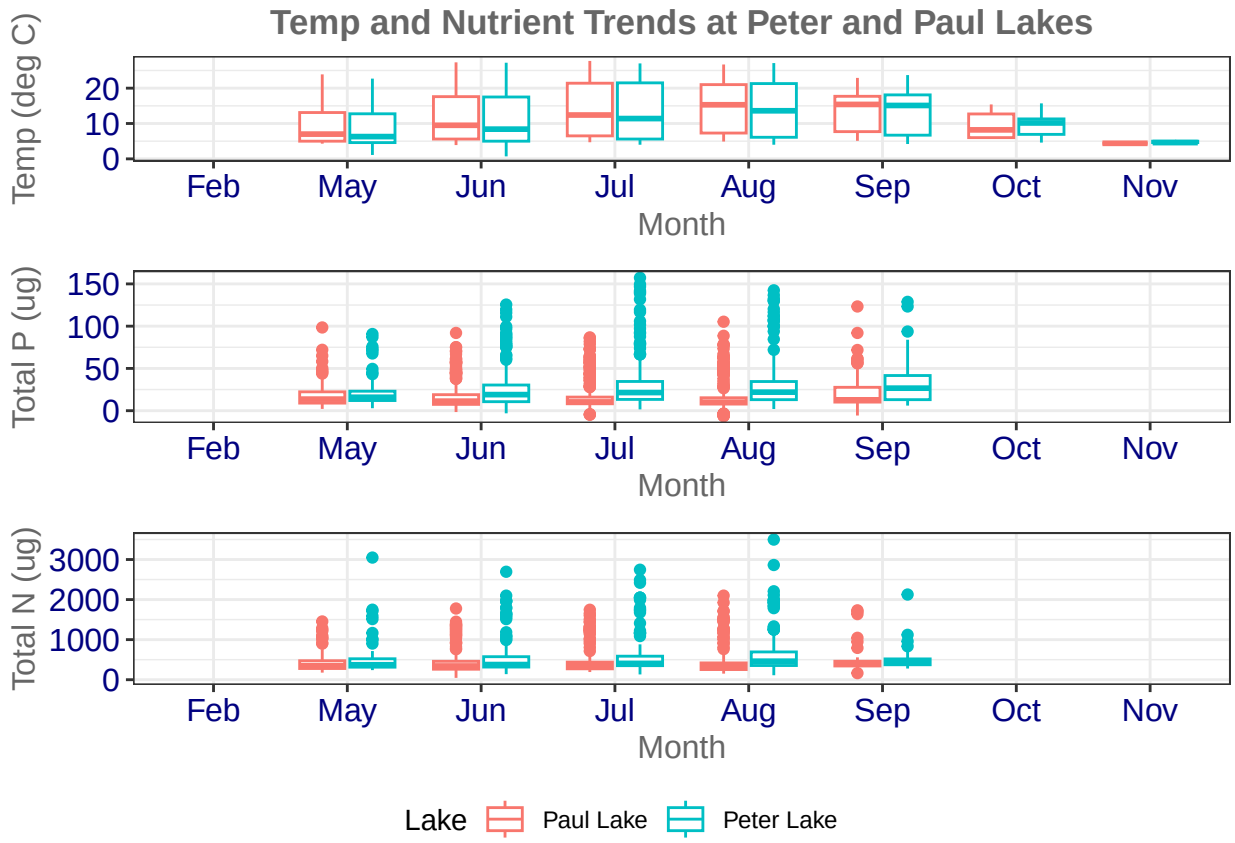
```
                           labels = c(month.abb)),
               y = temperature_C)) +
  ggtitle("Temp and Nutrient Trends at Peter and Paul Lakes") +
  geom_boxplot(aes(color = lakename)) +
  labs(color = 'Lake') +
  scale_y_continuous(name = "Temp (deg C)") +
  scale_x_discrete(name = "Month")

# print(a)

b <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = factor(month,
                        levels = c(1:12),
                        labels = c(month.abb)),
             y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(color = 'Lake') +
  scale_y_continuous(name = "Total P (ug)") +
  scale_x_discrete(name = "Month")

# print(b)

c <-
  ggplot(PeterPaul.chem.nutrients,
         aes(x = factor(month,
                        levels = c(1:12),
                        labels = c(month.abb)),
             y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(color = 'Lake') +
  scale_y_continuous(name = "Total N (ug)") +
  scale_x_discrete(name = "Month")

# print(c)

combined_plots <-
  plot_grid(a + theme(legend.position = "none"),
            b + theme(legend.position = "none"),
            c + theme(legend.position = "none"),
            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()‘).
```

```
legend <-
  get_legend(a)
```

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

```
combined_plots_legend <-
  plot_grid(combined_plots,
            legend,
            ncol = 1,
            rel_heights = c(1, .1))

print(combined_plots_legend)
```



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

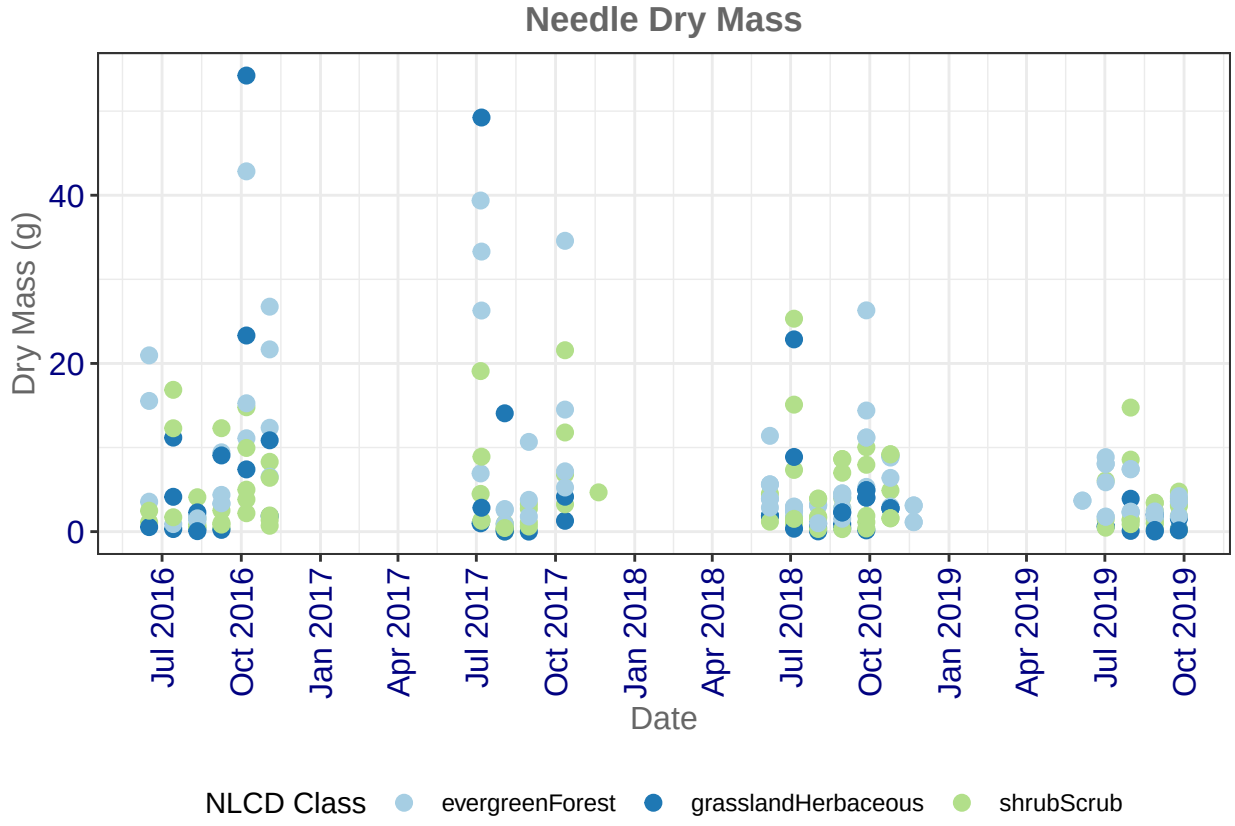
> Answer: There is a noticeable rise in temperature during summer months across both lakes. There doesn't appear to be a lot of seasonal variability in total phosphorus or nitrogren, but it does look like Peter Lake has slightly higher concentrations of both nutrients compared to Paul Lake.

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.

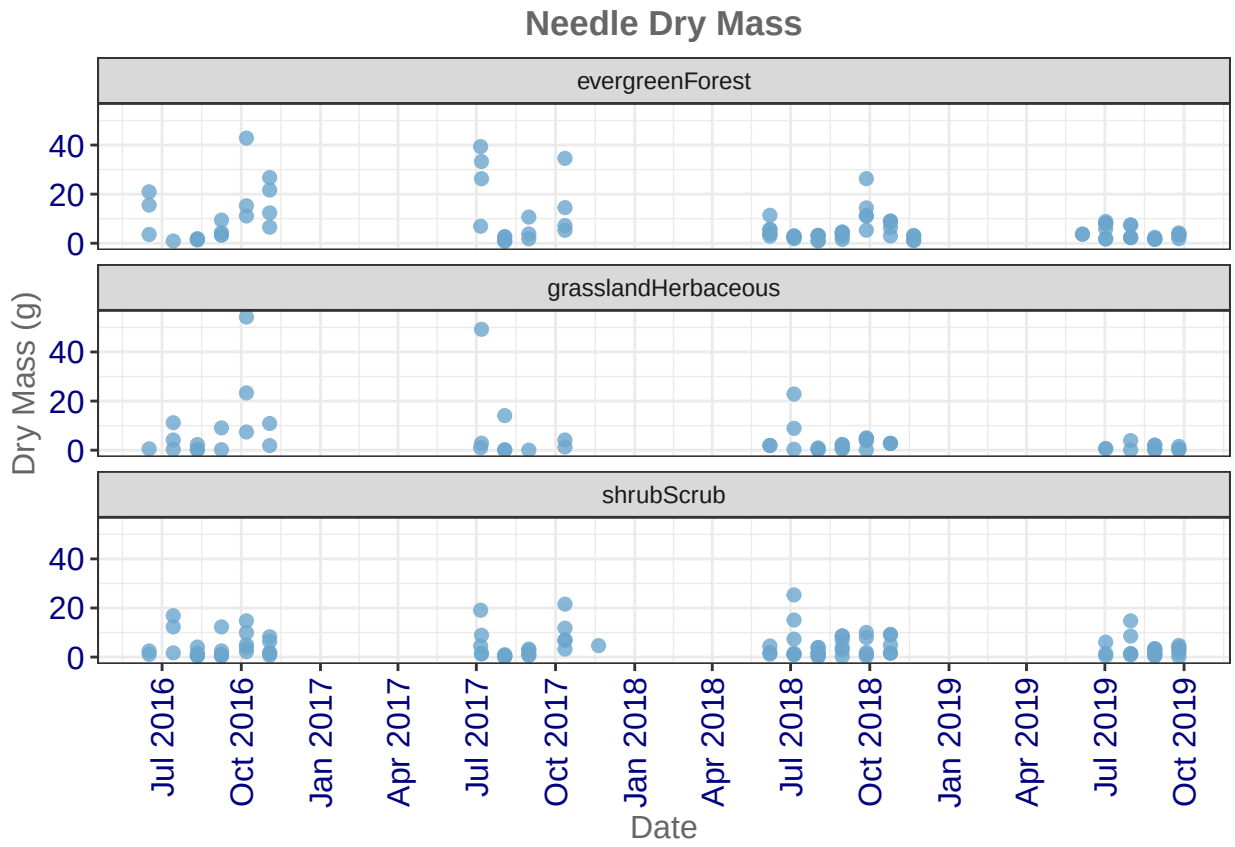
```
#6
needle_plot_color <-
  ggplot(NEON.NIWO.litter %>%
```

```
          filter(functionalGroup == "Needles"),
        aes(x = collectDate,
            y = dryMass,
            color = nlcdClass)) +
  ggtitle("Needle Dry Mass") +
  geom_point(alpha = 1,
             size = 2.5) +
  scale_x_date(name = "Date",
               date_breaks = "3 months",
               date_labels = "%b %Y") +
  scale_y_continuous(name = "Dry Mass (g)") +
  labs(color = "NLCD Class")+
  scale_color_brewer(palette = "Paired") +
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5,
                                   hjust = 1))

print(needle_plot_color)
```




```
#7
needle_plot_facet <-
  ggplot(NEON.NIWO.litter %>%
           filter(functionalGroup == "Needles"),
         aes(x = collectDate,
             y = dryMass)) +
```

```
  ggtitle("Needle Dry Mass") +
  facet_wrap(vars(nlcdClass),
             nrow = 3) +
  geom_point(alpha = 0.8,
             size = 2,
             color = "skyblue3") +
  scale_x_date(name = "Date",
               date_breaks = "3 months",
               date_labels = "%b %Y") +
  scale_y_continuous(name = "Dry Mass (g)") +
  labs(color = "NLCD Class")+
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5,
                                   hjust=1))

print(needle_plot_facet)
```



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

Answer: Personally, I think the second plot separated into facets instead of separated by color is a more effective visual. There's a lot of point overlap, making it difficult to distinguish classes in the first plot. Because the y axes stay the same across the faceted plots, it's easy to see at a glance which classes are similar and which have a greater distribution without the same difficulty trying the distinguish point colors. For example, it's easy to see in the second plot that in 2016 and 2017 there's a greater distribution in the evergreen and grassland classes, while the point distribution looks very similar across all three classes in 2018 and 2019. I think this is harder to

see at a gland in the first plot when you're trying to see which colors are present in a big cluster of points.