Assignment 5: Data Visualization

Vanshika Mittal

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 loading the libraries
library(tidyverse); library(lubridate); library(here); library(cowplot)
```

```
## -- 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
```

```
## here() starts at /home/guest/EDE_Fall2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
#Assinging a variable to our folder and calling relevant datasets
processed_data = "Data/Processed_KEY"
A05.NTL.PeterPaul <- read.csv(here(processed_data, "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processe
                              stringsAsFactors = TRUE, header = TRUE)
A05.Litter <- read.csv(
  here(processed_data, "NEON_NIWO_Litter_mass_trap_Processed.csv"),
                       stringsAsFactors = TRUE, header = TRUE)

#2 formatting date variables in R
A05.NTL.PeterPaul$sampledate <- ymd(A05.NTL.PeterPaul$sampledate)
A05.Litter$collectDate <- ymd(A05.Litter$collectDate)
```

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
my_theme <- theme_minimal() +
  theme(plot.background = element_rect('white'),
    legend.position = "top",
    plot.title = element_text(color = 'blue'),
    legend.box.background = element_rect(color = "black"))

theme_set(my_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()`).

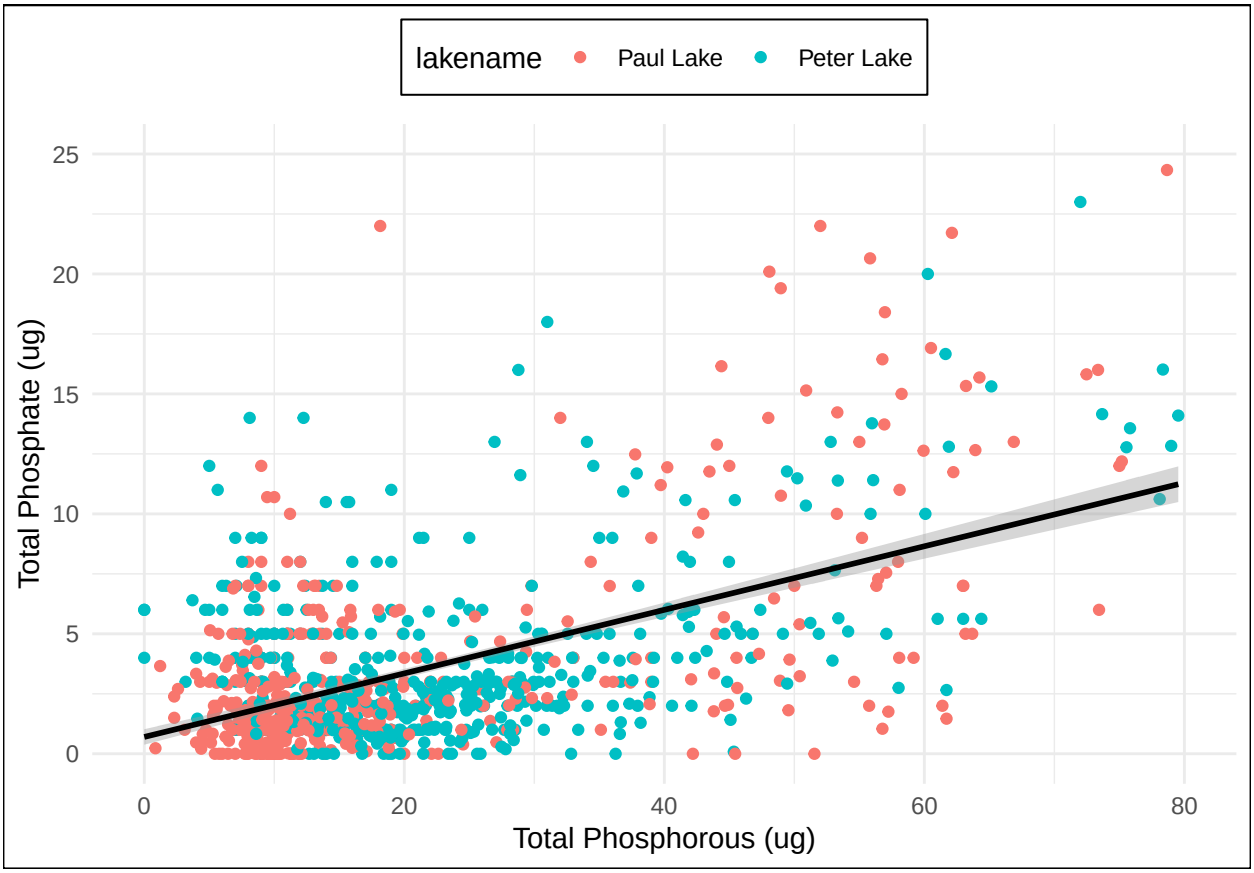
```
#4
P.by.PO4 <- ggplot(AO5.NTL.PeterPaul,aes(x= tp_ug, y = po4, color = lakename))+
            geom_point() +
            geom_smooth(method = lm, color = 'black') +
            labs(x = "Total Phosphorous (ug)", y = "Total Phosphate (ug)") +
            xlim(0, 80) +
            ylim (0, 25)

print(P.by.PO4)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 21988 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21988 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
#making boxplots for each variable
Temp_Boxplot <- ggplot(A05.NTL.PeterPaul,
                       aes(x= factor(month, levels = 1:12, labels = month.abb),
                       y = temperature_C, color = lakename)) +
                geom_boxplot() +
                scale_x_discrete(name = "", drop = FALSE) +
                theme(legend.position = 'top') +
                labs(x = "Month" , y= "Temperature (C)")

TP_Boxplot <- ggplot(A05.NTL.PeterPaul,
                     aes(x= factor(month, levels = 1:12, labels = month.abb),
                         y = tp_ug, color = lakename )) +
              geom_boxplot() +
              theme(legend.position = 'none') +
              scale_x_discrete(name = "", drop = FALSE) +
              labs(x = "Month" , y= "Total Phosphorous (ug)")

TN_Boxplot <- ggplot(A05.NTL.PeterPaul,
                     aes(x= factor(month, levels = 1:12, labels = month.abb),
                          y = tn_ug, color = lakename )) +
              geom_boxplot() +
              scale_x_discrete(name = "", drop = FALSE) +
              theme(legend.position = 'none') +
              labs(x = "Month" , y= "Total Nitrogen (ug)")

#combining the boxplots into one
Combined_Boxplot <- plot_grid(Temp_Boxplot, TP_Boxplot,TN_Boxplot,
                              nrow = 3, align = "hv")
```

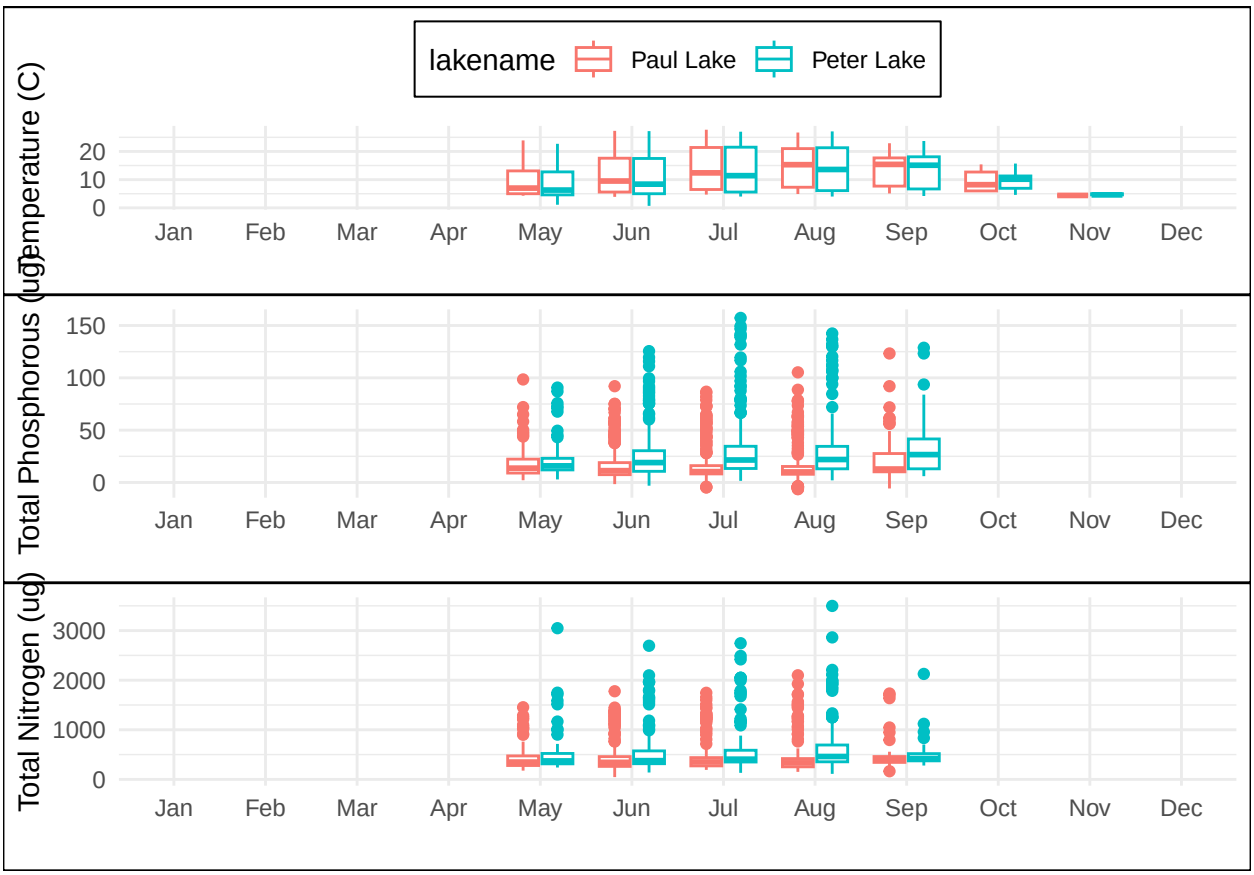
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).

## 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.
```

```
print(Combined_Boxplot)
```



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

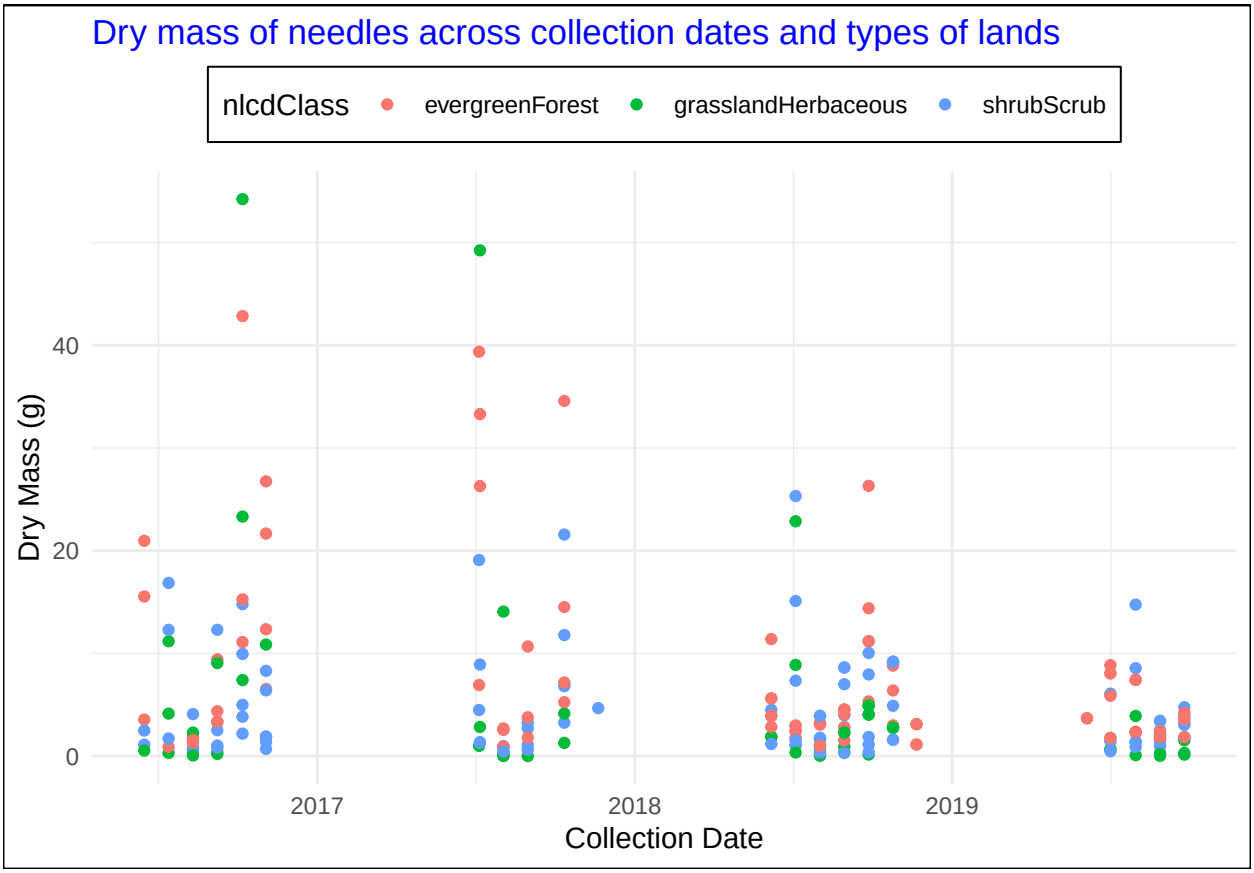
> Answer: Temperature in both Peter and Paul lakes is similar and decreasing over the months. The total phosphorous content in both lakes does not fluctuate much though there is a slight dip in the phosphorous in July and August before rising again in September in Paul Lake The total nitrogen content does not fluctuate much in the two lakes over seasons too.

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.

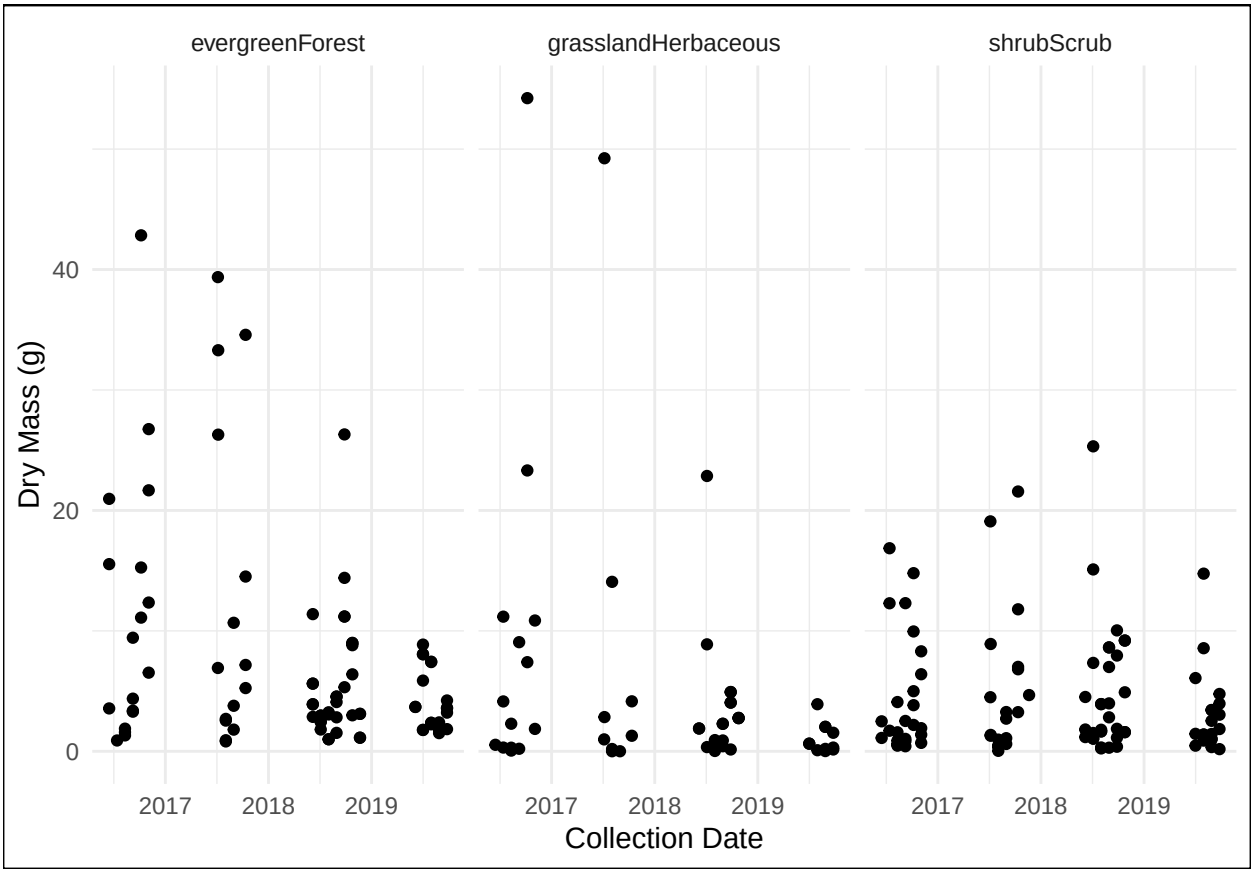
```
#6
Needles_plot <- ggplot(subset(A05.Litter, functionalGroup == "Needles")) +
                geom_point(aes(x = collectDate,y = dryMass,color = nlcdClass)) +
                labs(x= "Collection Date", y = "Dry Mass (g)",
title = "Dry mass of needles across collection dates and types of lands")

print(Needles_plot)
```




```
#7
Needles_faceted <- ggplot(subset(A05.Litter, functionalGroup == "Needles"),
                          aes(x= collectDate, y = dryMass)) +
                   geom_point() +
                   facet_wrap(vars(nlcdClass), nrow = 1) +
                   labs(x= "Collection Date", y = "Dry Mass (g)")

print(Needles_faceted)
```



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

Answer: I think 7 is visually more effective as it allows you to see an individual trend within all the types of lands while also comparing them side by side. However, in 6 there is an overlap of many data points which makes it difficult to read.