

# Assignment 5: Data Visualization

Samantha\_White-Murillo

Spring 2024

## OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

## Directions

1. Rename this file `Samantha_White-Murillo_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
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.5
## 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 /home/guest/EDA_Spring2024_SamanthaWM
```

```
library(cowplot)
```

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

```
here()
```

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

```
NiwotRidge.litter <- read.csv(
  file = here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)
```

```
#2
```

```
class(PeterPaul.chem.nutrients$sampdate)
class(NiwotRidge.litter$collectDate)
```

```
PeterPaul.chem.nutrients$sampdate <- ymd(PeterPaul.chem.nutrients$sampdate)
NiwotRidge.litter$collectDate <- ymd(NiwotRidge.litter$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
```

```
mytheme <- theme_light(base_size = 10) +
  theme(plot.title = element_text(face = "bold", size = 15,
                                   color = "purple"),
        axis.title = element_text(color = "darkblue", )
  ) +
  theme(legend.position = "bottom")
```

## 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 line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
PeterPaul_plot <-
  ggplot(subset(PeterPaul.chem.nutrients, depth == 0),
    aes(x = po4,
        y = tp_ug)) +
  geom_point() +
  facet_wrap(vars(lakename)) +
  labs(x = "Phosphorous",
       y = "Phosphate",
       title = "P vs P04") +
  geom_smooth(method = lm) +
  mytheme +
  xlim(0, 15) +
  ylim(0, 55)

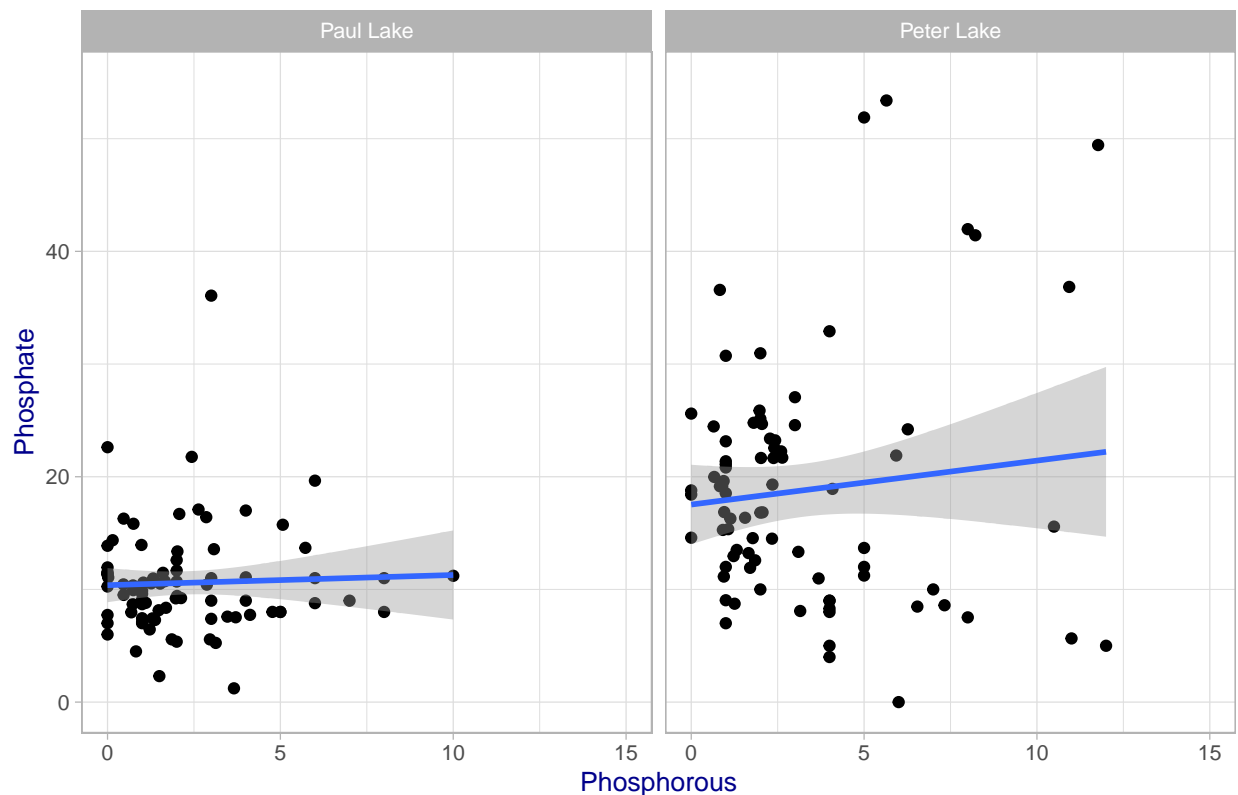
print(PeterPaul_plot)
```

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

```
## Warning: Removed 904 rows containing non-finite values ('stat_smooth()').
```

```
## Warning: Removed 904 rows containing missing values ('geom_point()').
```

## P vs P04



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.

Tips: \* Recall the discussion on factors in the lab section as it may be helpful here. \* Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) \* Setting a legend's position to "none" will remove the legend from a plot. \* Individual plots can have different sizes when combined using `cowplot`.

```
#5
library(cowplot)

Temperature <- ggplot(PeterPaul.chem.nutrients,
  aes(x = month,
      y = temperature_C,
      color = lakename)) +
  geom_point() +
  labs(x = "Month", y = "Temperature", color = "Lake Name") +
  mytheme +
  theme(legend.position = "right")

TP <- ggplot(PeterPaul.chem.nutrients,
  aes(x = month,
      y = tp_ug,
```

```

        color = lakename)) +
geom_point() +
labs(x = "Month", y = "Phosphate") +
mytheme +
theme(legend.position = "none") +
ylim(0, max(PeterPaul.chem.nutrients$tp_ug, na.rm = TRUE))

TN <- ggplot(PeterPaul.chem.nutrients,
            aes(x = month,
                y = tn_ug,
                color = lakename)) +
geom_point() +
labs(x = "Month", y = "Nitrogen") +
mytheme +
theme(legend.position = "none")

Temperature_TP_TN <- plot_grid(TP, TN, Temperature) +
  labs(title = "Temperature TP TN")

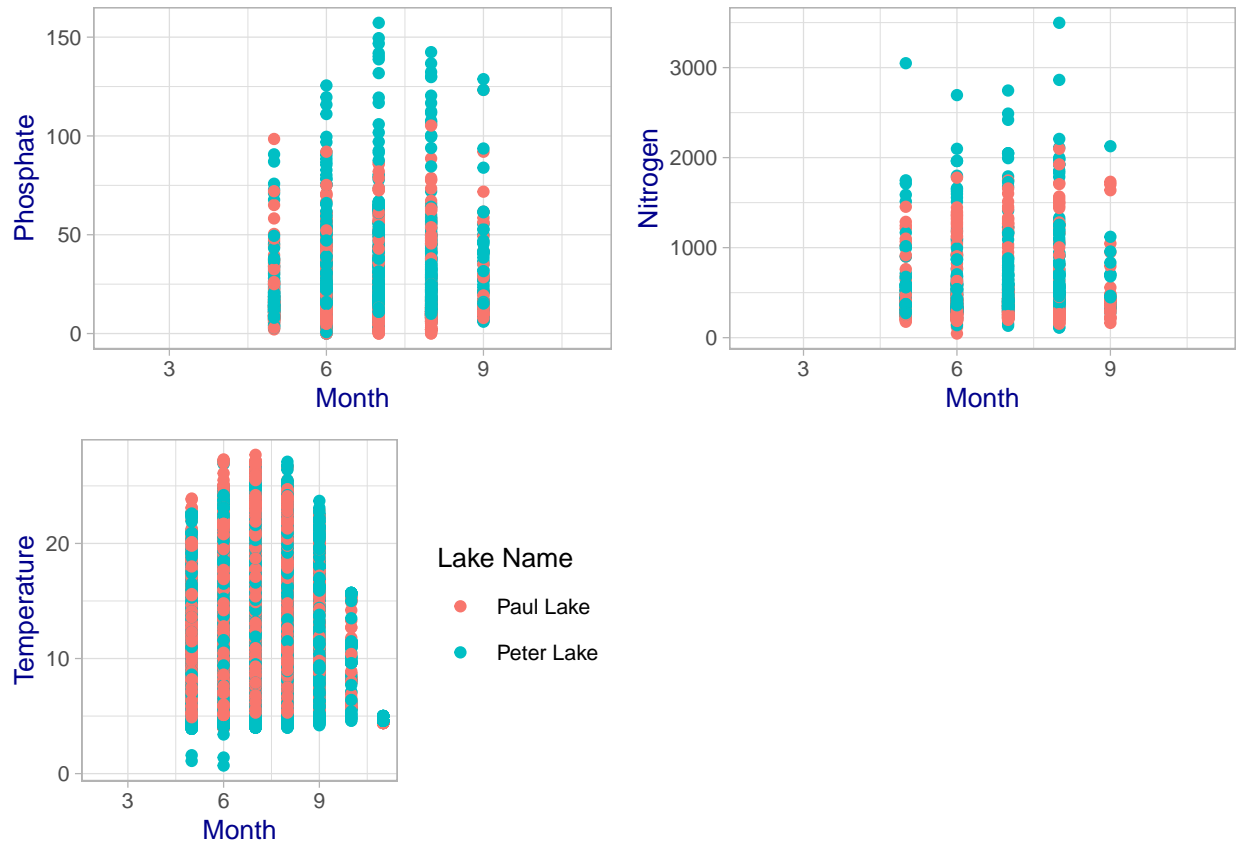
```

```
## Warning: Removed 20766 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 21583 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 3566 rows containing missing values ('geom_point()').
```

```
print(Temperature_TP_TN)
```



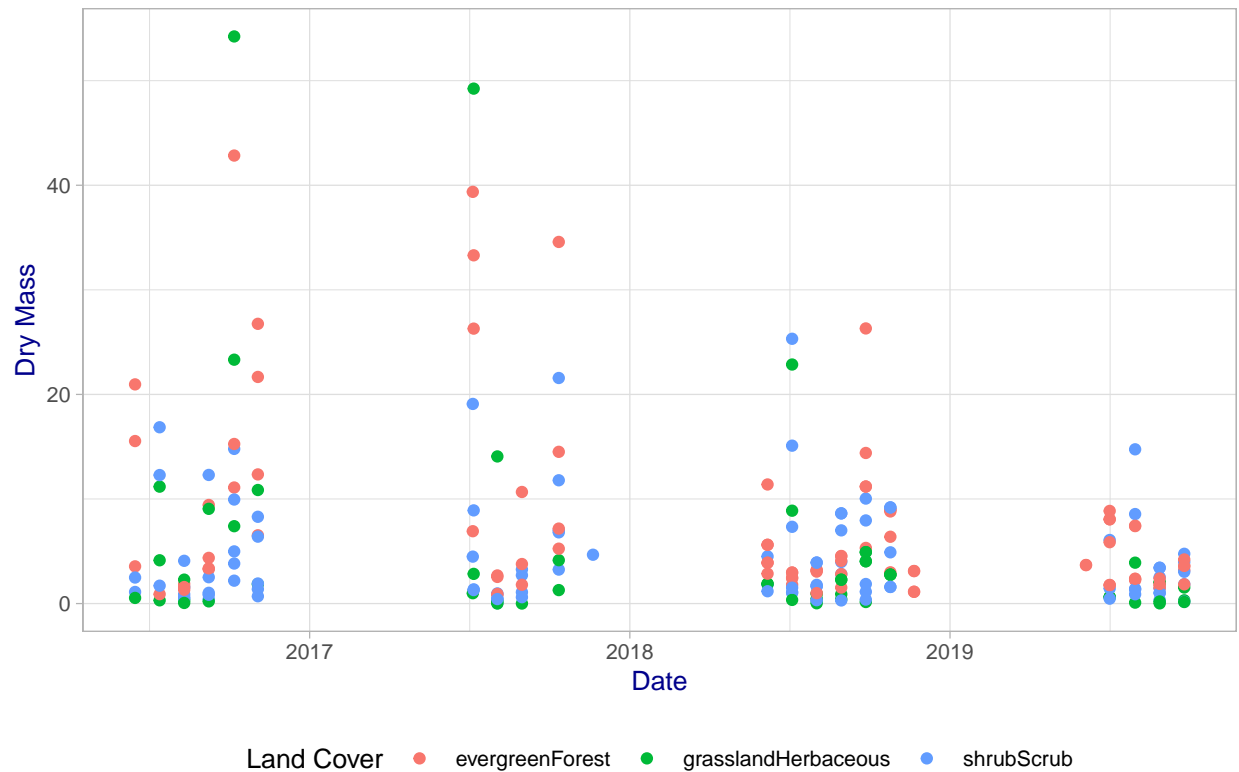
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Concentrations for phosphorous and nitrogen are higher in Peter Lake than Paul Lake. The month registered with warmest temperatures was July, coincidentally for Peter Lake, that's the month with the most phosphorous level registered. Also is important to mention that the data is only registered for summer and beginning of fall season (making it difficult to make seasonal comparisons).

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.

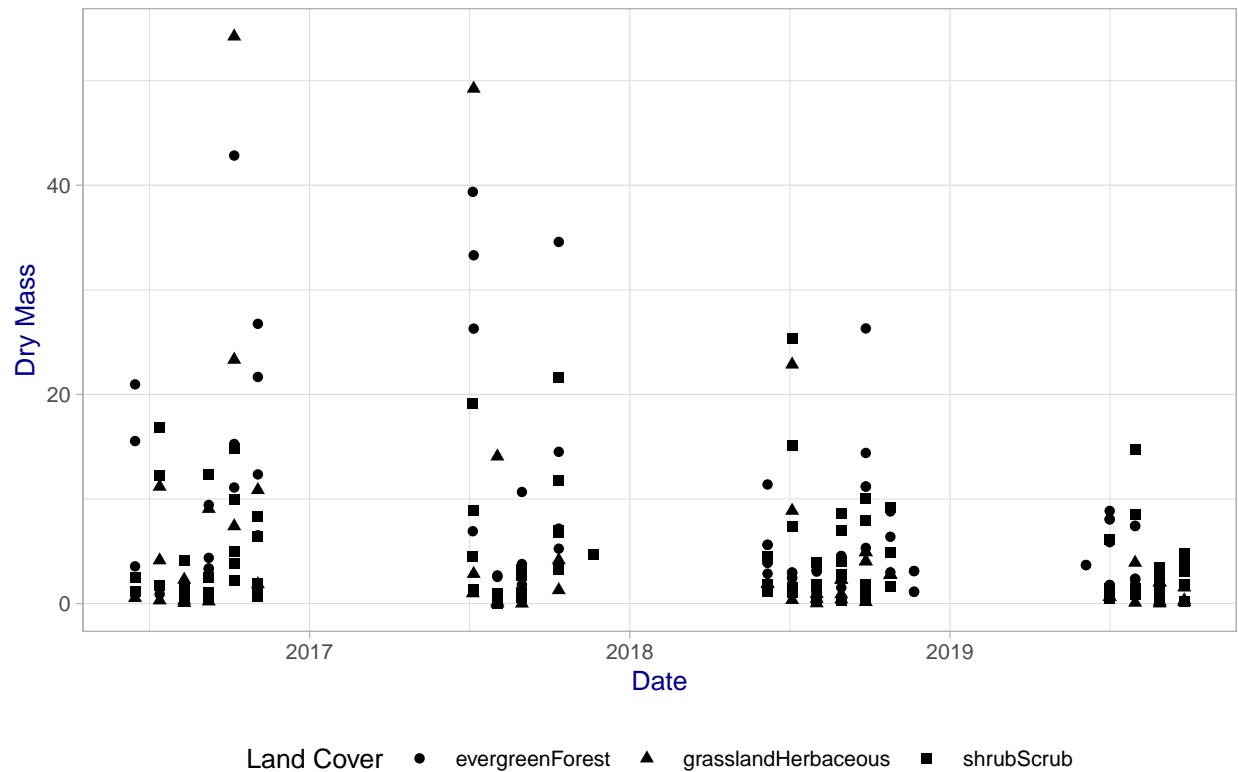
```
#6
Needle_litter <- NiwotRidge.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate,
             y = dryMass,
             color = nlcdClass)) +
  geom_point() +
  labs(x = "Date", y = "Dry Mass", color = "Land Cover",
       title = "Needle litter") +
  mytheme
print(Needle_litter)
```

## Needle litter



```
#7
Needle_litter_facets <- NiwotRidge.litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate,
             y = dryMass,
             shape = nlcdClass)) +
  geom_point() +
  labs(x = "Date", y = "Dry Mass", shape = "Land Cover",
       title = "Needle litter faced") +
  mytheme
print(Needle_litter_facets)
```

## Needle litter faced



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

Answer: I consider the color classification is more effective in this case because the data number is broad and sometimes overlaps, making it difficult to differentiate the same color shapes. Maybe with less data and only two Land Covers, shape would be a more effective option.