Assignment 5: Data Visualization

Jiyoung Park

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct 21th @ 5:00pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse, lubridate, & 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_PeterPa` version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processe` version).

2. Make sure R is reading dates as date format; if not change the format to date.

```
# 1
setwd("~/EDA-Fall2022")
require("knitr")
```

```
## Loading required package: knitr
```

```
opts_knit$set(root.dir = "~/EDA-Fall2022")
getwd()
```

```
## [1] "C:/Users/hyjgp/Documents/EDA-Fall2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# install.packages('cowplot')
library(cowplot)
```

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

```
NTL_LTER <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
    stringsAsFactors = TRUE)
NEON_NIWO <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)
# 2
NEON_NIWO$collectDate <- as.Date(NEON_NIWO$collectDate, format = "%Y-%m-%d")
NTL_LTER$sampledate <- as.Date(NTL_LTER$sampledate, format = "%Y-%m-%d")
```

Define your theme

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

```
# 3
mytheme <- theme_classic(base_size = 14) + theme(axis.text = element_text(color = "black"),
    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.

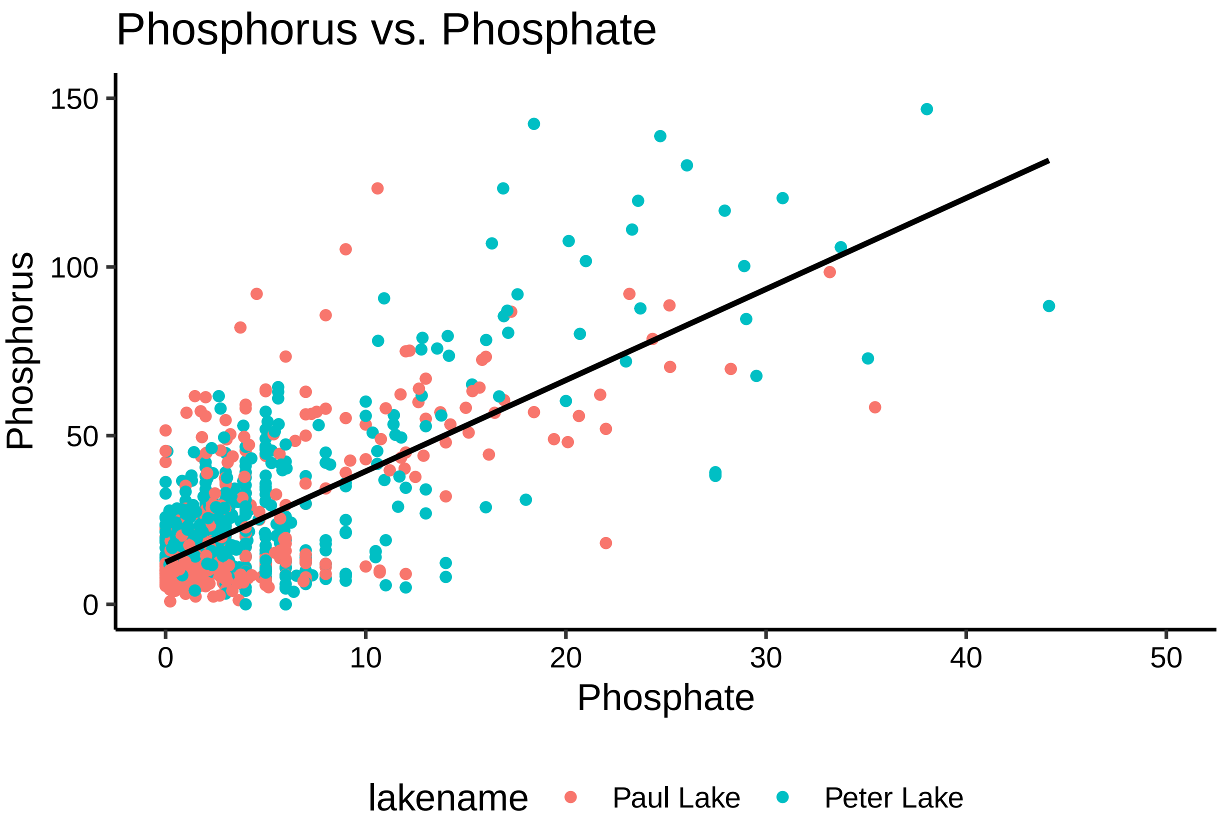
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
# 4
ggplot(NTL_LTER, aes(x = po4, y = tp_ug)) + geom_point(aes(color = lakename)) + geom_smooth(method = lm
    color = "black", se = FALSE) + mytheme + xlim(0, 50) + ylim(0, 150) + labs(title = "Phosphorus vs.
    x = "Phosphate", y = "Phosphorus")
```

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

## Warning: Removed 21948 rows containing non-finite values (stat_smooth).

## Warning: Removed 21948 rows containing missing values (geom_point).
```

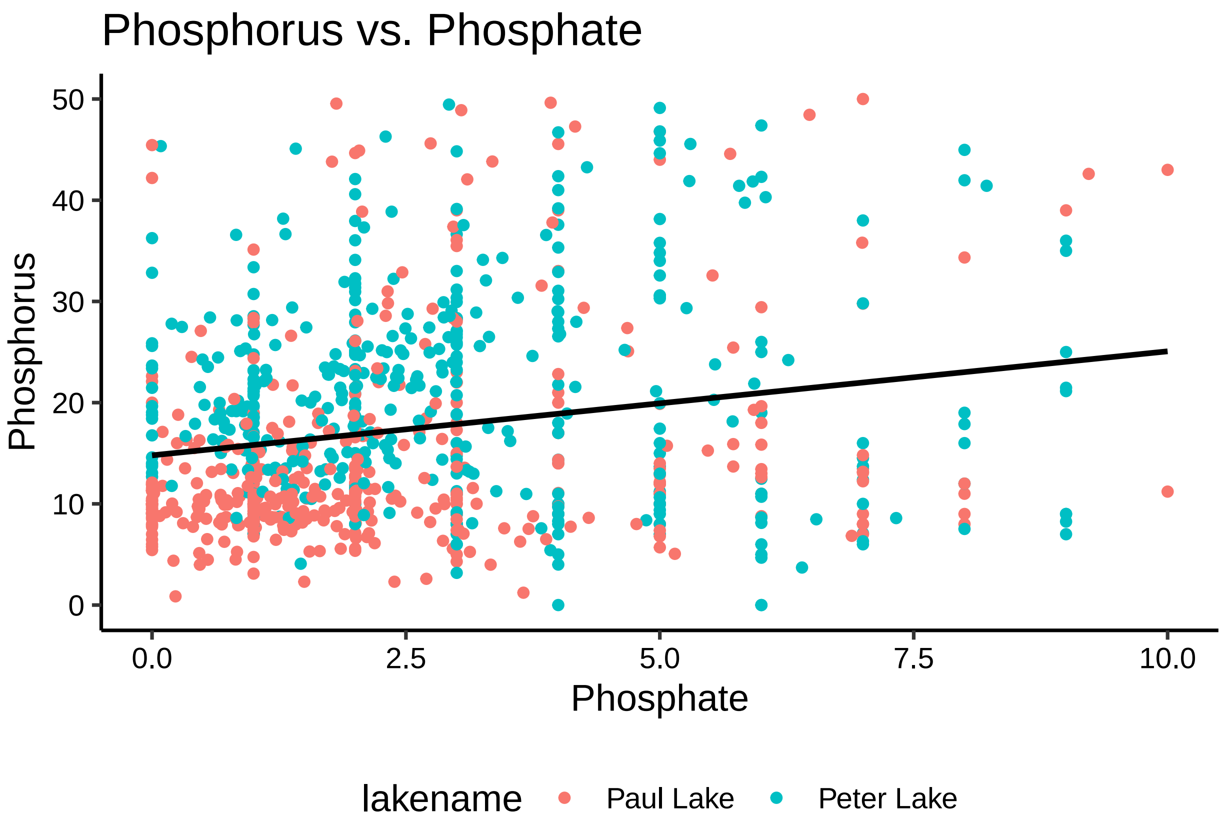


```
# zooming in more
ggplot(NTL_LTER, aes(x = po4, y = tp_ug)) + geom_point(aes(color = lakename)) + geom_smooth(method = lm
    color = "black", se = FALSE) + mytheme + xlim(0, 10) + ylim(0, 50) + labs(title = "Phosphorus vs. Ph
    x = "Phosphate", y = "Phosphorus")
```

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

## Warning: Removed 22100 rows containing non-finite values (stat_smooth).

## Warning: Removed 22100 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: R has a build in variable called `month.abb` that returns a list of months; see https://r-lang.com/month-abb-in-r-with-example

```
# 5
abbmonth <- month.abb[NTL_LTER$month]

temp <- ggplot(NTL_LTER, aes(x = abbmonth, y = temperature_C)) + geom_boxplot(aes(color = lakename)) +
    mytheme + theme(legend.position = "none") + labs(title = "Temperature by months",
    x = "months", y = "Temperature (C)")

TP <- ggplot(NTL_LTER, aes(x = abbmonth, y = tp_ug)) + geom_boxplot(aes(color = lakename)) +
    mytheme + theme(legend.position = "none") + labs(title = "TP by months", x = "months",
    y = "TP")

TN <- ggplot(NTL_LTER, aes(x = abbmonth, y = tn_ug)) + geom_boxplot(aes(color = lakename)) +
    mytheme + theme(legend.position = "none") + labs(title = "TN by months", x = "months",
    y = "TN")

temp_legend <- ggplot(NTL_LTER, aes(x = abbmonth, y = temperature_C)) + geom_boxplot(aes(color = lakena
legend <- get_legend(temp_legend)
```

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

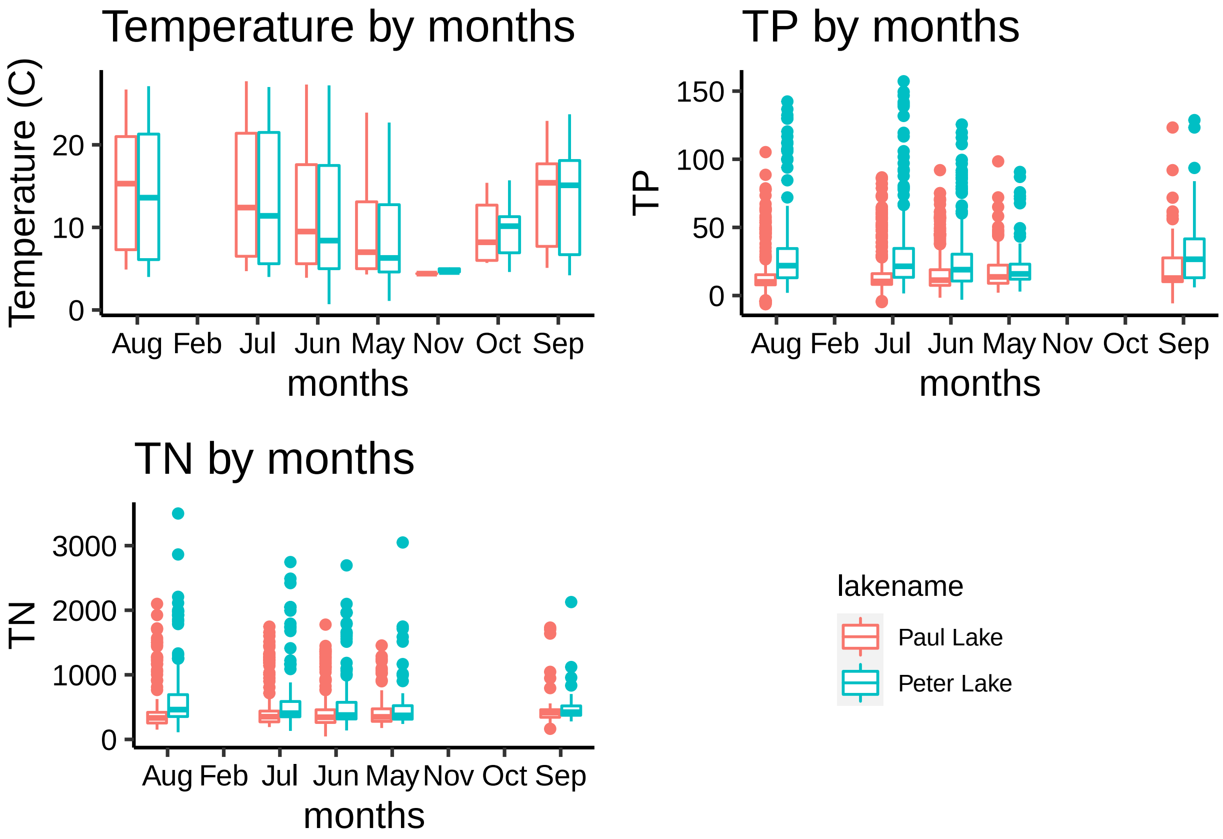
```
plot_grid(temp, TP, TN, legend, align = "h")
```

```
## 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.
```

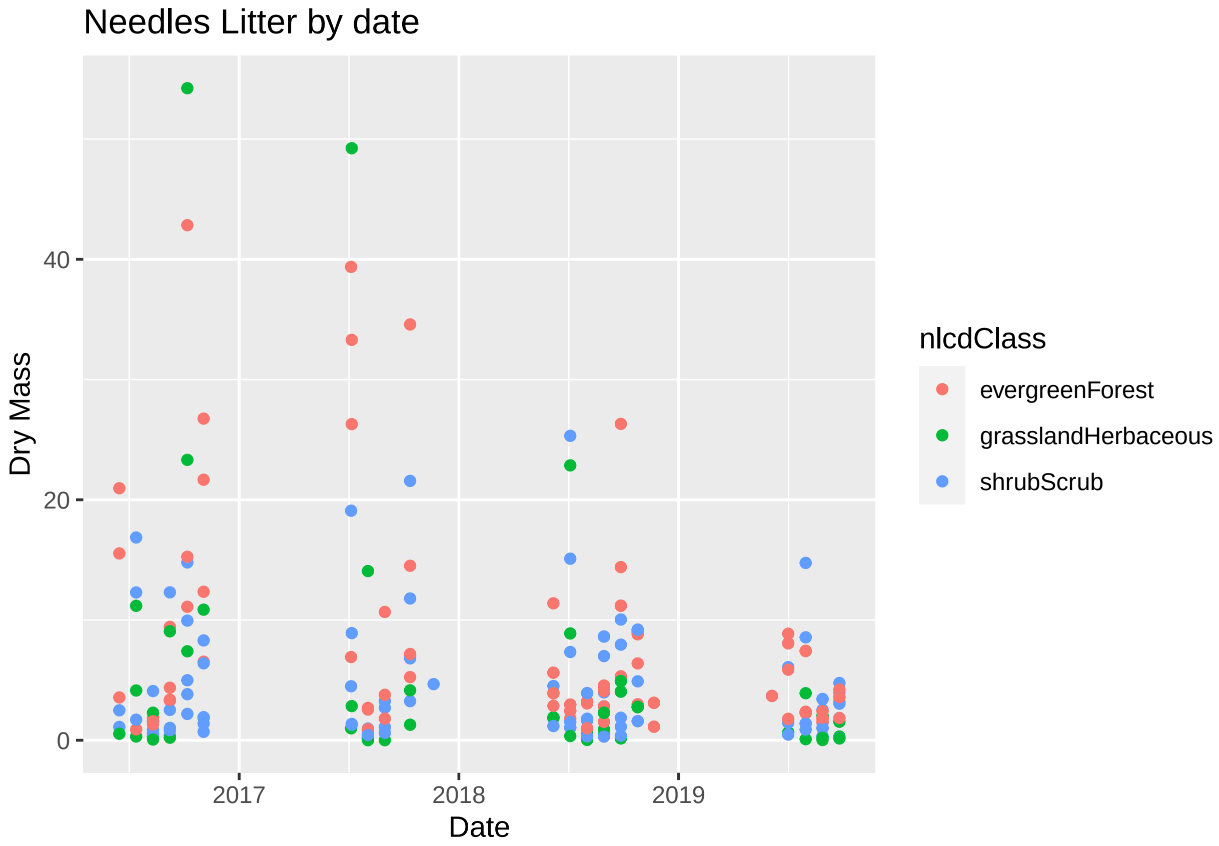


Question: What do you observe about the variables of interest over seasons and between lakes?

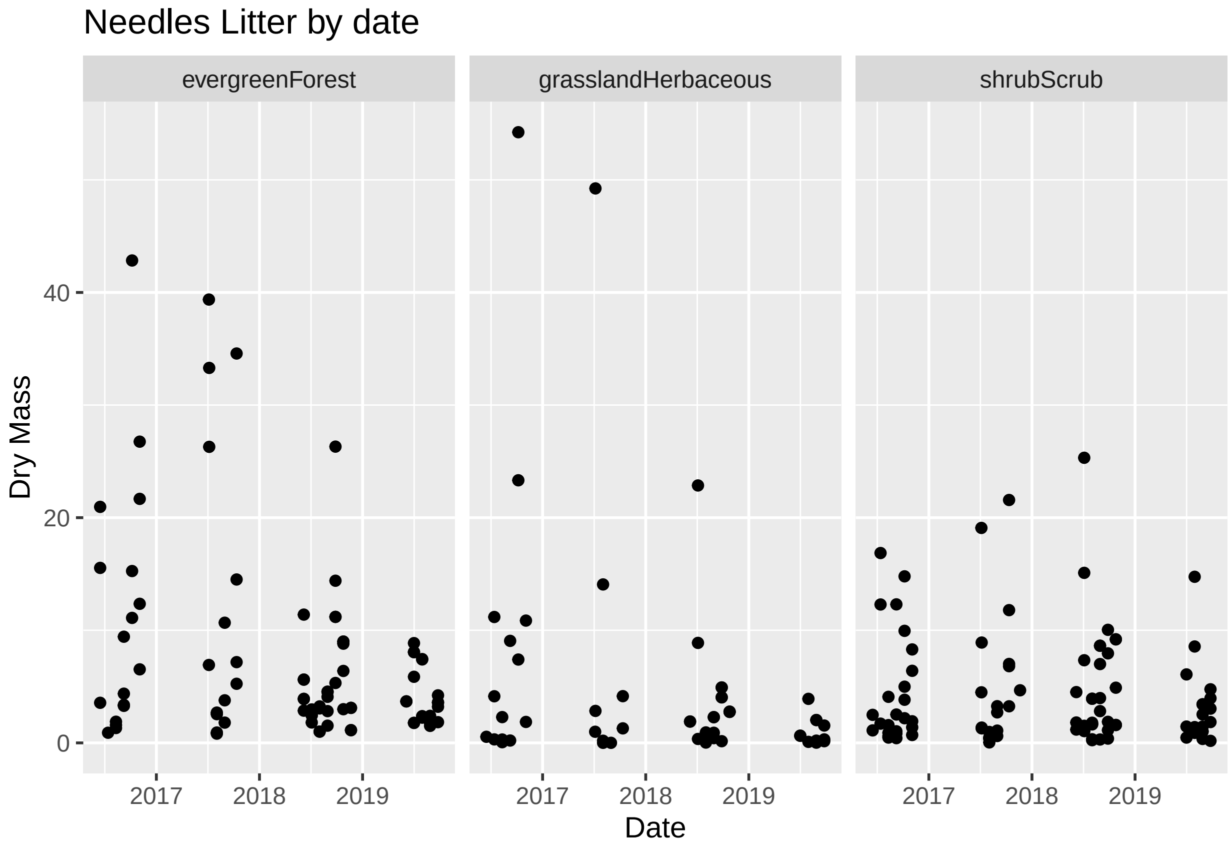
> Answer: high TP and TN concentration with months that are high in temperatures. No measures found in months with cold temperatures(Feb, Nov, Oct)

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
NEON_NIWO_needles <- filter(NEON_NIWO, functionalGroup == "Needles")
ggplot(NEON_NIWO_needles, aes(x = collectDate, y = dryMass)) + geom_point(aes(color = nlcdClass)) +
    labs(title = "Needles Litter by date", x = "Date", y = "Dry Mass")
```



```
# 7
ggplot(NEON_NIWO_needles, aes(x = collectDate, y = dryMass)) + geom_point() + facet_wrap(vars(nlcdClass
    labs(title = "Needles Litter by date", x = "Date", y = "Dry Mass")
```



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

Answer: i believe plot 7 is more effective at comparing than plot 6 because i can see the trends for drymass for three different ncld class more clearly in plot 7.