Assignment 5: Data Visualization

Lydie Costes

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 and load packages
getwd()
```

```
## [1] "/Users/lydiecostes/Documents/Duke/DataAnalytics/GithubRepos/Environmental_Data_Analytics_2022/As
```

```
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()
```

```r
library(cowplot)
# Import data
PeterPaul <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                      stringsAsFactors = TRUE)
Niwot <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                  stringsAsFactors = TRUE)

# 2. Check format of date columns and fix
str(PeterPaul$sampledate) # Factor
```

```
##  Factor w/ 1103 levels "1984-05-27","1984-05-28",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
PeterPaul$sampledate <- as.Date(PeterPaul$sampledate, "%Y-%m-%d")
str(Niwot$collectDate) # Factor
```

```
##  Factor w/ 24 levels "2016-06-16","2016-07-14",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
Niwot$collectDate <- as.Date(Niwot$collectDate, "%Y-%m-%d")
```

Define your theme

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

```r
# 3. Build and set default theme
mytheme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)
```

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.

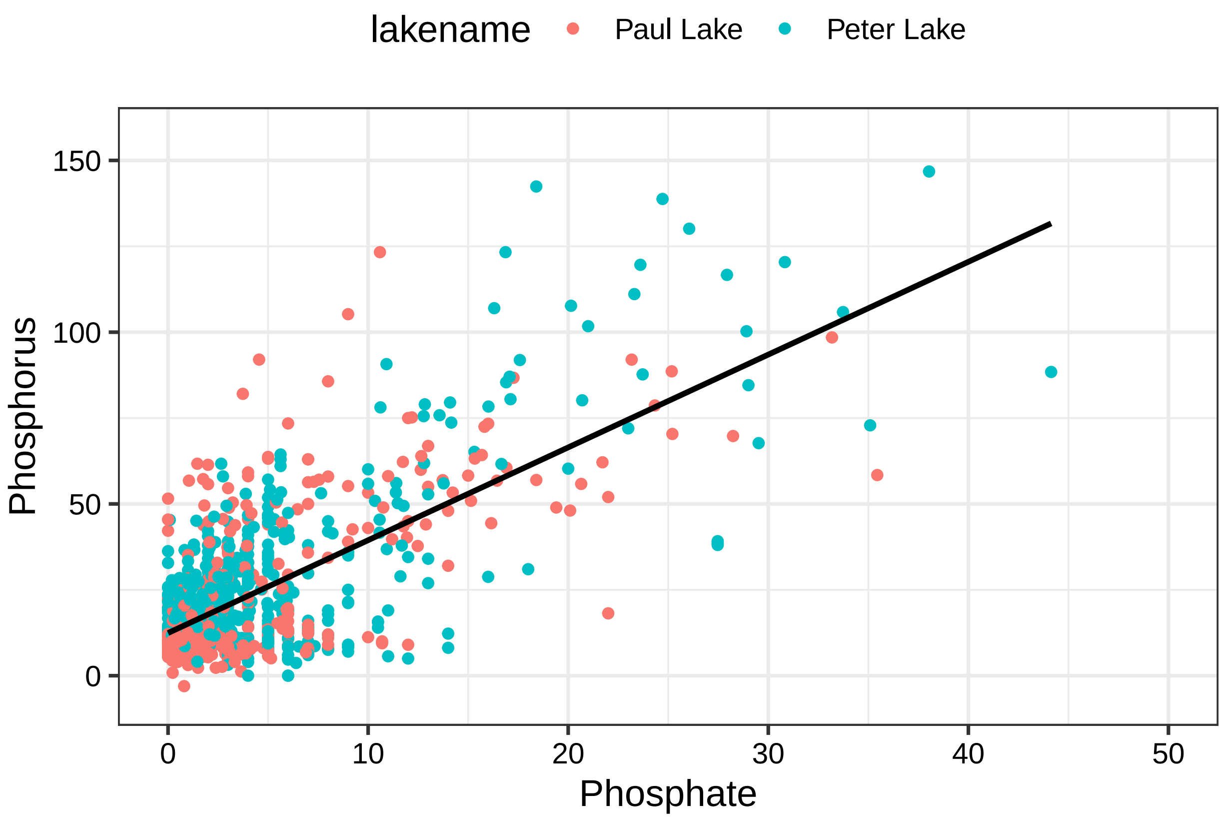
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and `ylim()`).

```r
# 4. Plot phosphorus by phosphate
ggplot(PeterPaul, aes(x=po4, y=tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  xlim(0,50) +
  labs(x = "Phosphate", y = "Phosphorus")
```

```
## `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.

```
# 5. Boxplots
# Set month to be a factor
PeterPaul$month <- as.factor(PeterPaul$month)
# Plot 1: temperature
boxplot_temp <- ggplot(PeterPaul, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot()  +
  facet_wrap("lakename") +
  labs(y = "Temp") +
  theme(legend.position = "none")
# Plot 2: TP
boxplot_tp <- ggplot(PeterPaul, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  facet_wrap("lakename") +
  labs(y = "TP") +
  theme(legend.position = "none")
# Plot 3: TN
boxplot_tn <- ggplot(PeterPaul, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  facet_wrap("lakename") +
  labs(y = "TN") +
  theme(legend.position = "none")
```

```r
boxplot_temp2 <- ggplot(PeterPaul, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot()  +
  facet_wrap("lakename") +
  labs(y = "Temp")

mylegend <- get_legend(boxplot_temp2)
```

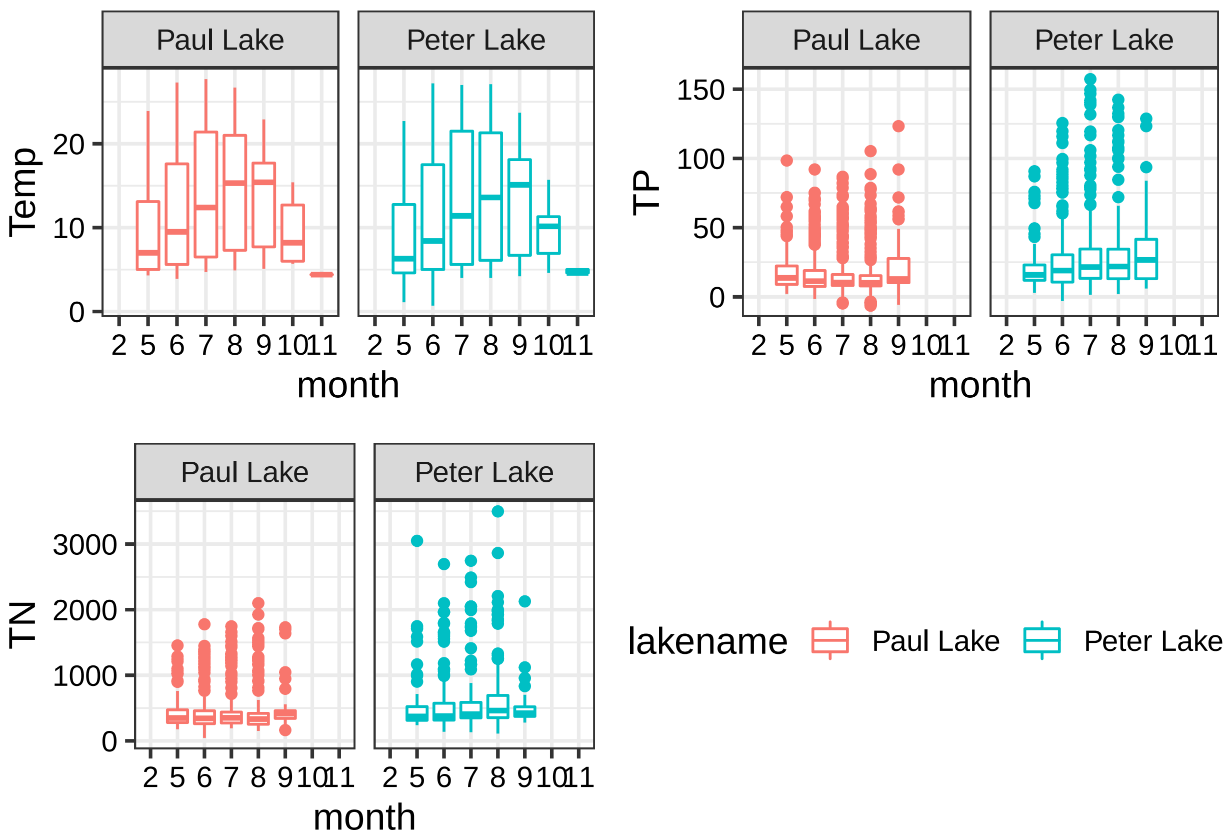
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```

```r
plot_grid(boxplot_temp, boxplot_tp, boxplot_tn, mylegend, nrow = 2)
```

```
## 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).
```



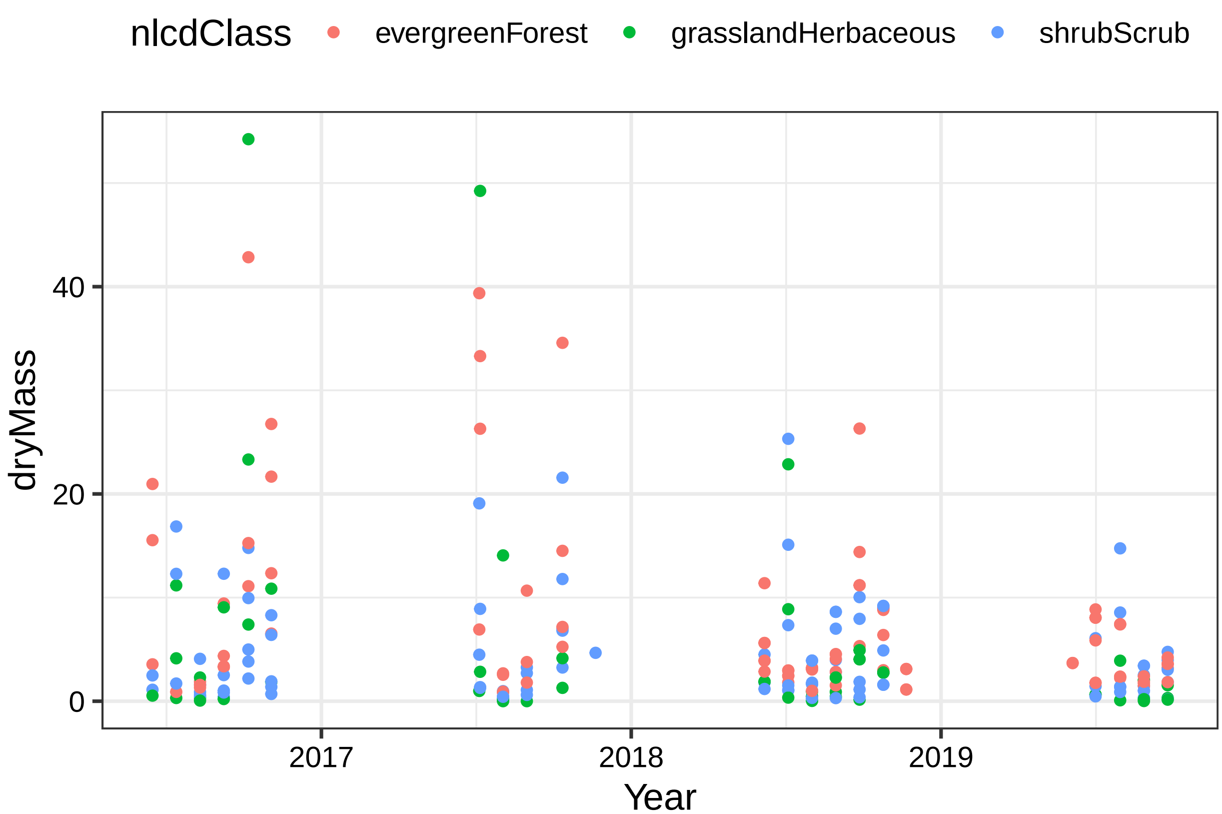
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature variations are pretty similar between the two lakes. As expected, temperature is higher in the summer and lower in the winter. Total phosphorous varies between the two lakes with Paul Lake declining in the summer and then increasing again towards fall. Peter Lake experiences a slow increase over the season. Total nitrogen appears to increase slowly over the growing season, which makes sense with agricultural inputs.

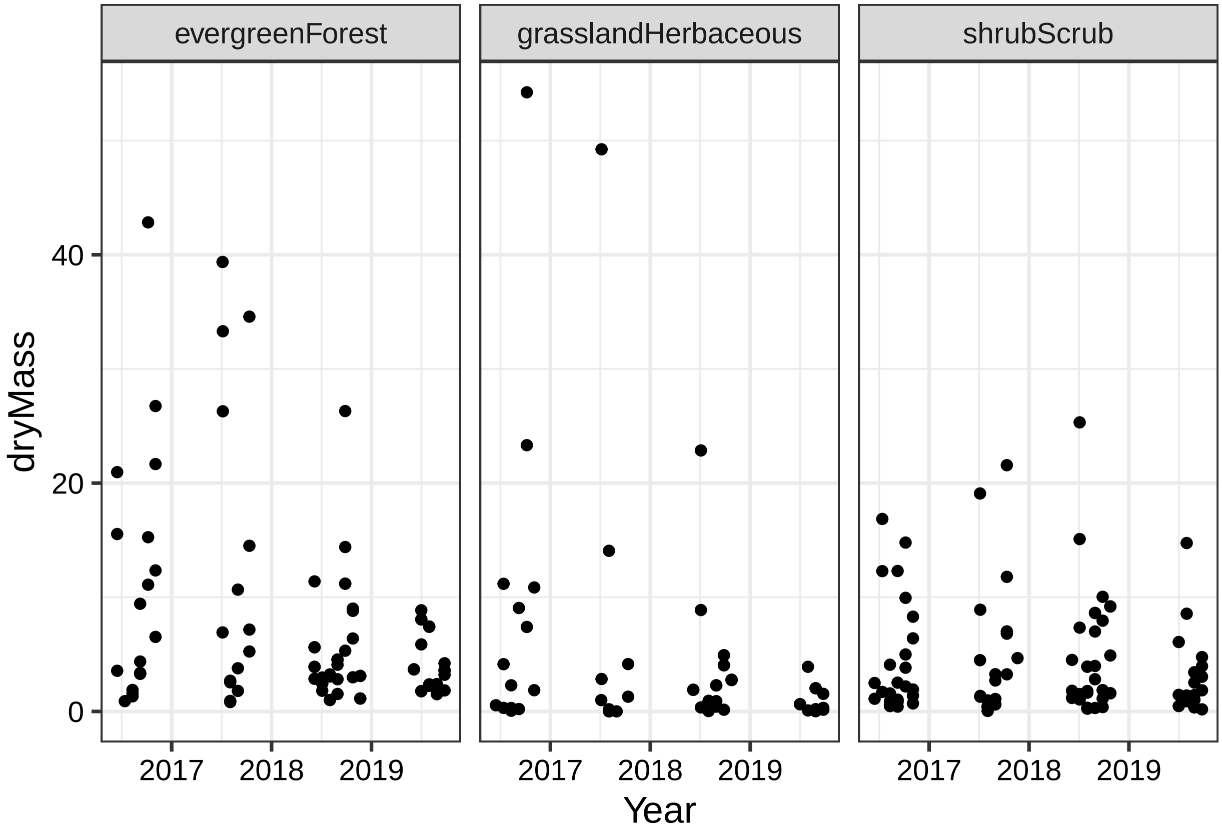
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```r
# 6.
ggplot(filter(Niwot, functionalGroup == "Needles"),
       aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point() +
  labs(x="Year")
```




```r
# 7.
ggplot(filter(Niwot, functionalGroup == "Needles"),
       aes(x=collectDate, y=dryMass)) +
  geom_point() +
  facet_wrap("nlcdClass") +
  labs(x="Year")
```



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

> Answer: Plot 7 is more effective, because it's difficult to view the differences between the NLCD Classes when they are all overlaid.