Assignment 5: Data Visualization

Katryna Niva

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 14th @ 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_PeterPau version) and the processed data file for the Niwot Ridge litter dataset (use the [NEON_NIWO_Litter_mass_trap_Processed version).

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

```
# 1

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   1.2.0     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
```

```r
library(cowplot)
```

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

```r
PeterPaulNutrients <- read_csv("Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.cs
```

```
## Rows: 23008 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): lakename
## dbl  (13): year4, daynum, month, depth, temperature_C, dissolvedOxygen, irra...
## date  (1): sampledate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
Litter <- read_csv("Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
```

```
## Rows: 1692 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): plotID, trapID, functionalGroup, qaDryMass, nlcdClass, plotType, g...
## dbl  (5): dryMass, subplotID, decimalLatitude, decimalLongitude, elevation
## date (1): collectDate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# 2

class(PeterPaulNutrients$sampledate)
```

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

```r
class(Litter$collectDate)
```

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

```r
# Both are already in date format
```

Define your theme

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

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

theme_set(mytheme)
```

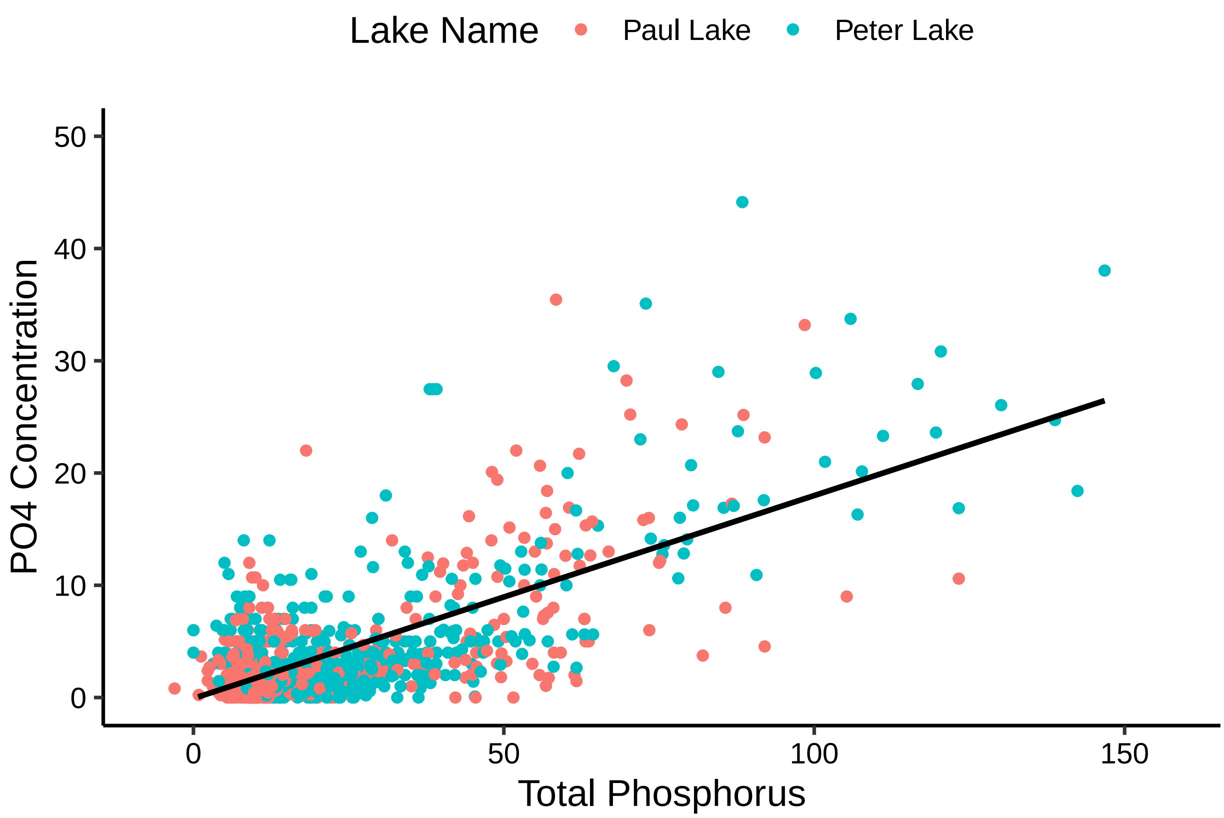
Create graphs

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/or `ylim()`).

```
# 4

Ex4 <- ggplot(PeterPaulNutrients, aes(x = tp_ug, y = po4, color = lakename)) + geom_point() +
    scale_y_continuous(limits = c(0, 50)) + ylab("PO4 Concentration") + xlab("Total Phosphorus") +
    labs(color = "Lake Name") + geom_smooth(method = "lm", se = FALSE, aes(color = NULL),
    color = "black") + theme(legend.position = "top")

print(Ex4)
```

```
## `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).
## Warning: Removed 2 rows containing missing values (geom_smooth).
```



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

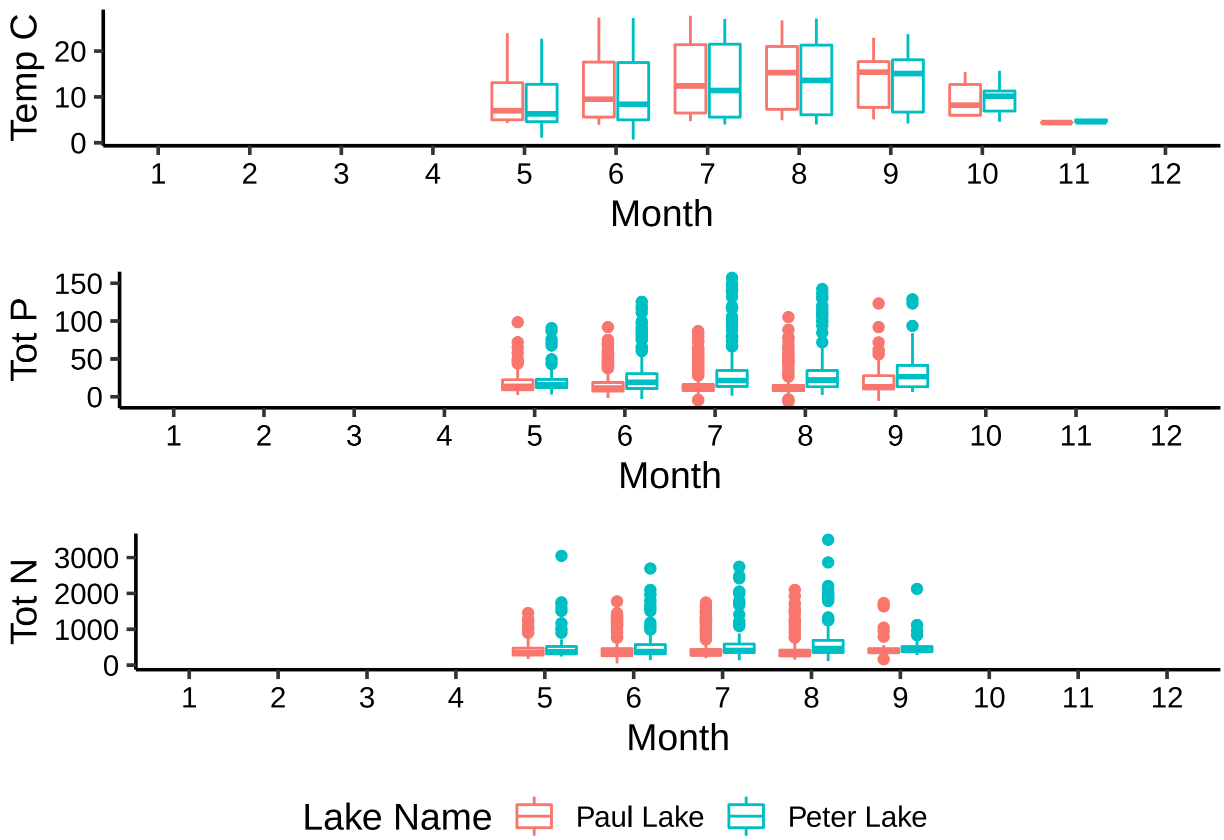
```
# 5

Ex5Temp <- ggplot(PeterPaulNutrients, aes(x = factor(month, levels = c(1:12)), y = temperature_C,
    color = lakename)) + scale_x_discrete("Month", drop = FALSE) + ylab("Temp C") +
    labs(color = "Lake Name") + geom_boxplot() + theme(legend.position = "top")

Ex5TP <- ggplot(PeterPaulNutrients, aes(x = factor(month, levels = c(1:12)), y = tp_ug,
    color = lakename)) + geom_boxplot() + scale_x_discrete("Month", drop = FALSE) +
    ylab("Tot P")

Ex5TN <- ggplot(PeterPaulNutrients, aes(x = factor(month, levels = c(1:12)), y = tn_ug,
    color = lakename)) + geom_boxplot() + scale_x_discrete("Month", drop = FALSE) +
    ylab("Tot N")

NoLegendEx5 <- plot_grid(Ex5Temp + theme(legend.position = "none"), Ex5TP + theme(legend.position = "no
    Ex5TN + theme(legend.position = "none"), nrow = 3, align = "h", rel_heights = c(3,
        3, 3), hjust = 1)
```

```
## 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(Ex5Temp)
```

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

```
Ex5 <- plot_grid(NoLegendEx5, legend, rel_heights = c(4, 0.4), nrow = 2)

print(Ex5)
```

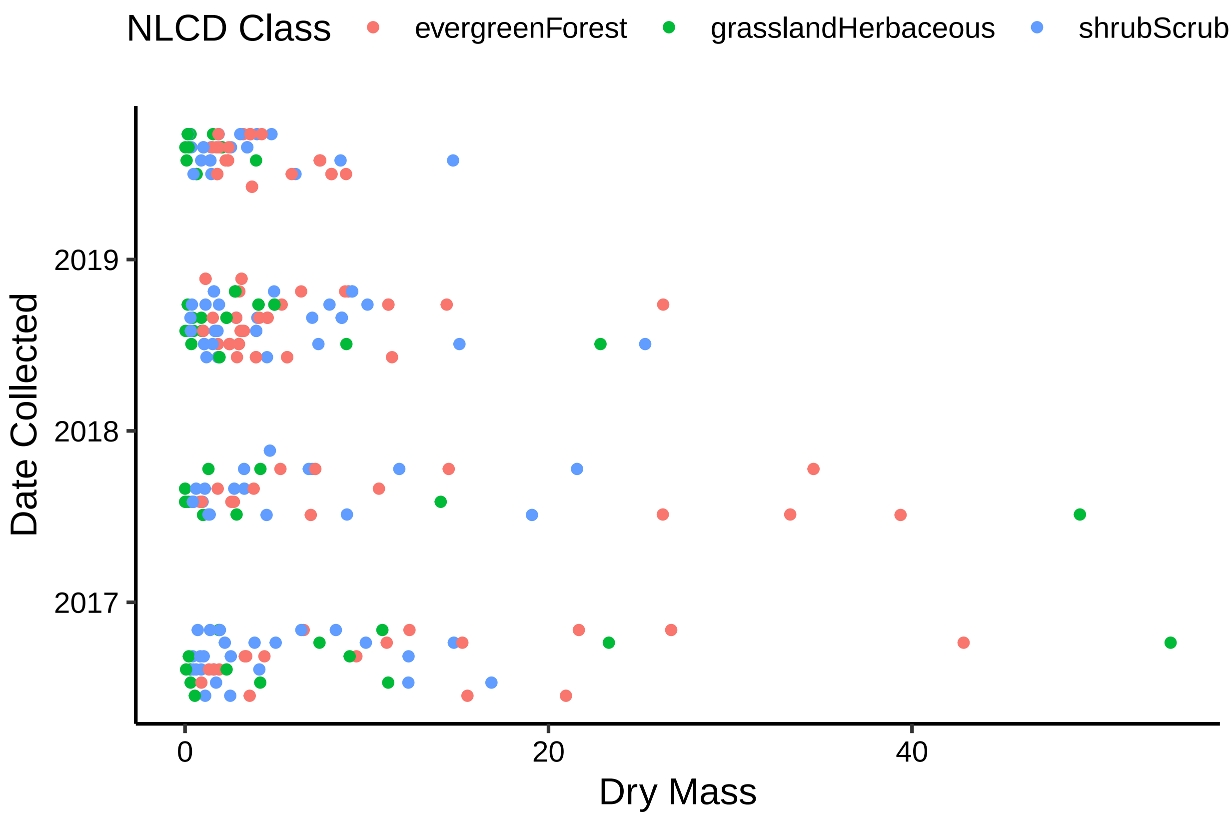


Question: What do you observe about the variables of interest over seasons and between lakes?

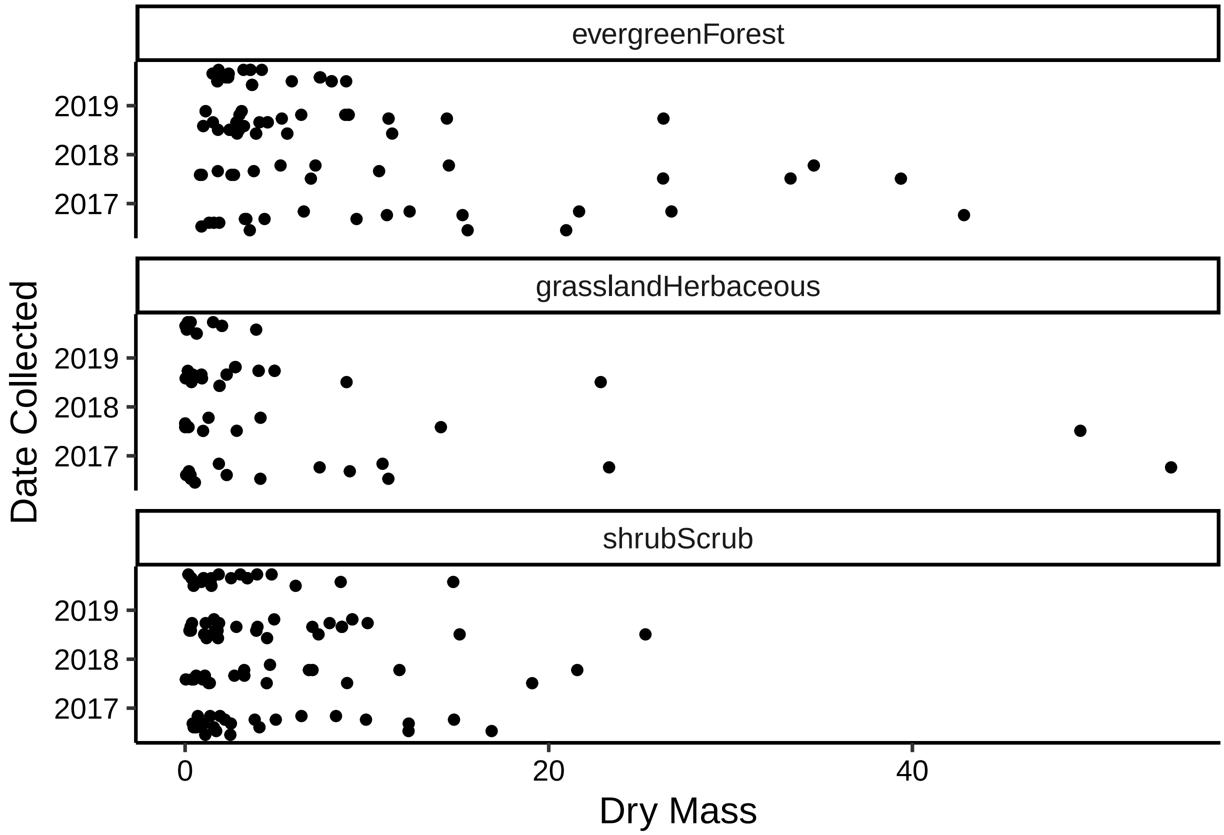
> Answer: Although data was not collected in all months, patterns can definitely be seen. The temperature increases at both locations during the summer months (as can be expected). Total phosphorus was highest during the summer months at Peter Lake and lowest during the summer months at Paul Lake. Peter Lake also generally had greater phosphorus levels. The nitrogen levels were also generally higher at Peter Lake. The nitrogen levels were more consistent across seasons. There are also many outliers at high concentrations for the phosphorus and nitrogen levels, indicating there are certain variable events that drastically increases the availability.

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

Ex6 <- ggplot(subset(Litter, functionalGroup == "Needles"), aes(x = dryMass, y = collectDate,
    color = nlcdClass)) + geom_point() + ylab("Date Collected") + xlab("Dry Mass") +
    labs(color = "NLCD Class") + theme(legend.position = "top")

print(Ex6)
```




```
# 7

Ex7 <- ggplot(subset(Litter, functionalGroup == "Needles"), aes(x = dryMass, y = collectDate)) +
    geom_point() + facet_wrap(vars(nlcdClass), nrow = 3) + ylab("Date Collected") +
    xlab("Dry Mass")

print(Ex7)
```



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

> Answer: The second faceted plot is the most effective. Due to how much the datapoints overlap and the magnitude of datapoints, it is difficult to identify patterns in a single color while being distracted by other colors. The plot in exercise 7 shows each pattern individually without the interference of other patterns. It therefore makes distinquishing patterns far easier.