# Assignment 5: Data Visualization

Leonardo Rueda

## 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_PeterP`
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

getwd()
```

```
## [1] "C:/Users/leor9/OneDrive/Leonardo/MIDP Courses Fall 2022/R Class/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
```

```
library(cowplot)
```

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

```
NTL_LTER_processed_data <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Proces
    stringsAsFactors = TRUE)

Niwot_Ridge_litter_dataset <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)

# 2

NTL_LTER_processed_data$sampledate <- as.Date(NTL_LTER_processed_data$sampledate,
    format = "%Y-%m-%d")

Niwot_Ridge_litter_dataset$collectDate <- as.Date(Niwot_Ridge_litter_dataset$collectDate,
    format = "%Y-%m-%d")
```

## Define your theme

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

```
# 3

mytheme <- theme_classic(base_size = 11) + theme(axis.text = element_text(color = "black"),
    legend.position = "top")
```

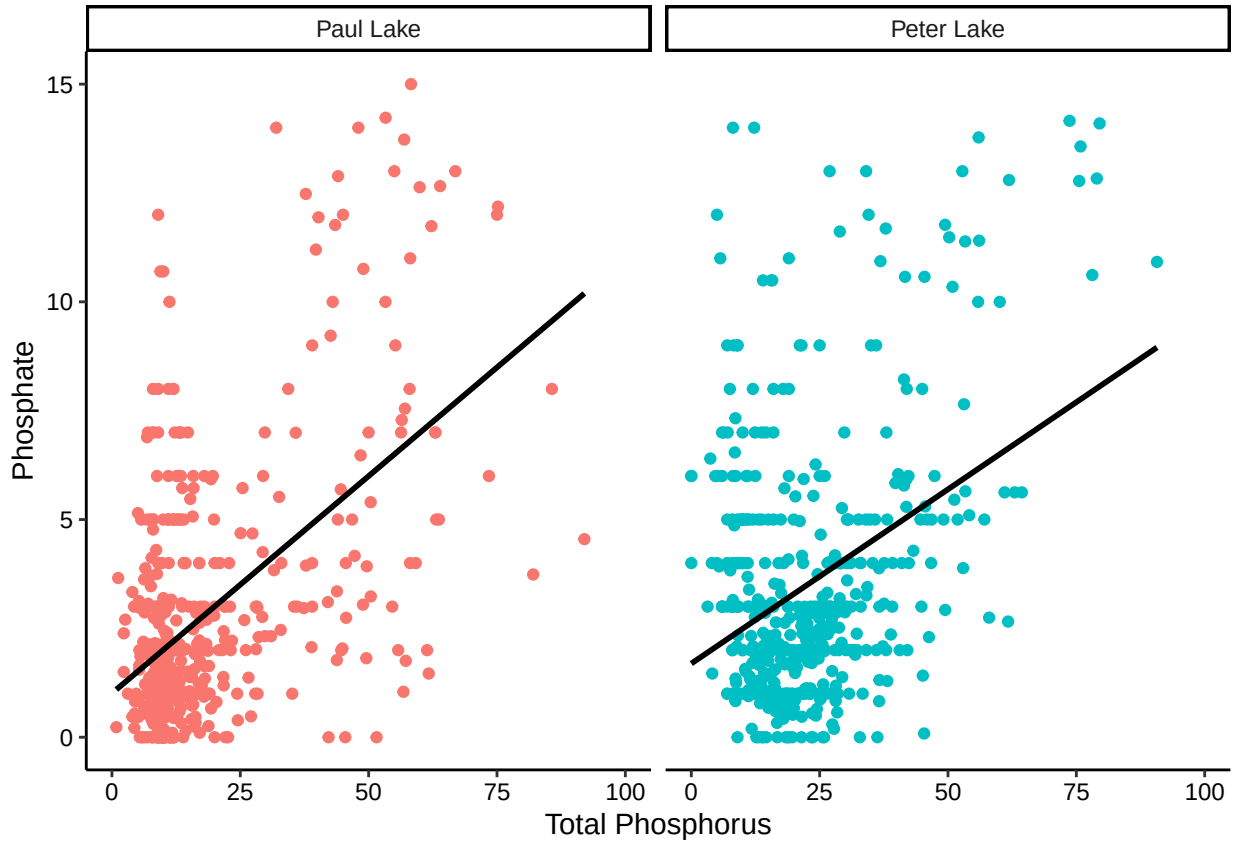
## Create graphs

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()`).

```r
# 4

tp_ug_po4_plot_by_lake <- ggplot(NTL_LTER_processed_data,
    aes(x = tp_ug, y = po4)) + geom_point(aes(colour = lakename),
    show.legend = FALSE) + mytheme + xlim(0, 100) +
    ylim(0, 15) + geom_smooth(method = "lm", se = FALSE,
    color = "black") + facet_wrap(vars(lakename)) +
    xlab("Total Phosphorus") + ylab("Phosphate")

print(tp_ug_po4_plot_by_lake)
```

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

## Warning: Removed 22007 rows containing non-finite values (stat_smooth).

## Warning: Removed 22007 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.

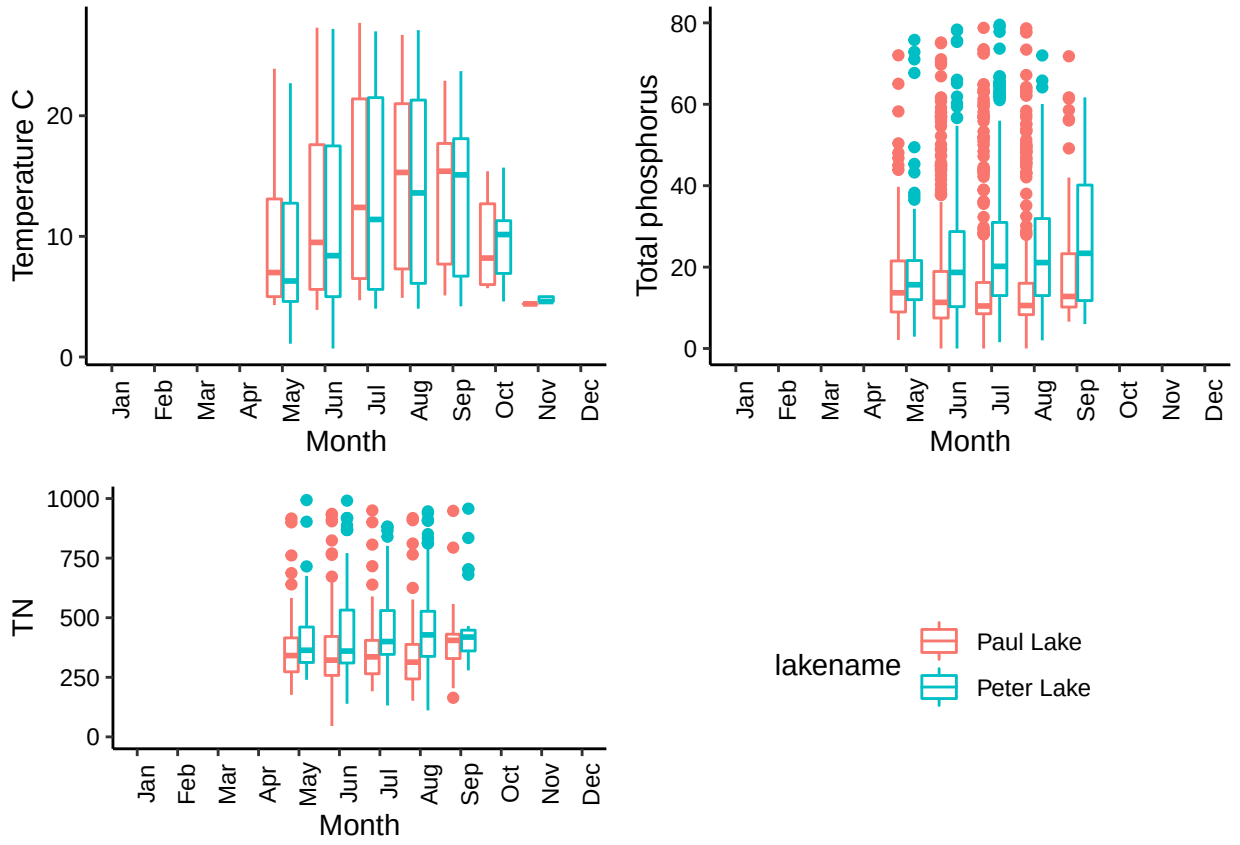
Tip: R has a build in variable called `month.abb` that returns a list of months; see https://r-lang.com/month-abb-in-r-with-example

```
# 5

temperature_plot <- ggplot(NTL_LTER_processed_data,
    aes(x = month.abb[month], y = temperature_C)) +
    geom_boxplot(aes(colour = lakename)) + mytheme +
    xlab("Month") + ylab("Temperature C") + scale_x_discrete(limits = month.abb) +
    theme(axis.text.x = element_text(angle = 90))

TP_plot <- ggplot(NTL_LTER_processed_data, aes(x = month.abb[month],
    y = tp_ug)) + geom_boxplot(aes(colour = lakename)) +
    mytheme + ylim(0, 80) + xlab("Month") + ylab("Total phosphorus") +
    scale_x_discrete(limits = month.abb) + theme(axis.text.x = element_text(angle = 90))

TN_plot <- ggplot(NTL_LTER_processed_data, aes(x = month.abb[month],
    y = tn_ug)) + geom_boxplot(aes(colour = lakename)) +
    mytheme + ylim(0, 1000) + xlab("Month") +
    ylab("TN") + scale_x_discrete(limits = month.abb) +
    theme(axis.text.x = element_text(angle = 90))

legend <- get_legend(temperature_plot + guides(color = guide_legend(ncol = 1)) +
    theme(legend.position = "bottom"))
```

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

```
plot_grid(temperature_plot + theme(legend.position = "none"),
    TP_plot + theme(legend.position = "none"),
    TN_plot + theme(legend.position = "none"),
    legend, nrow = 2, align = "h", rel_heights = c(1.25,
        1))
```

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

## Warning: Removed 20827 rows containing non-finite values (stat_boxplot).

## Warning: Removed 21730 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?

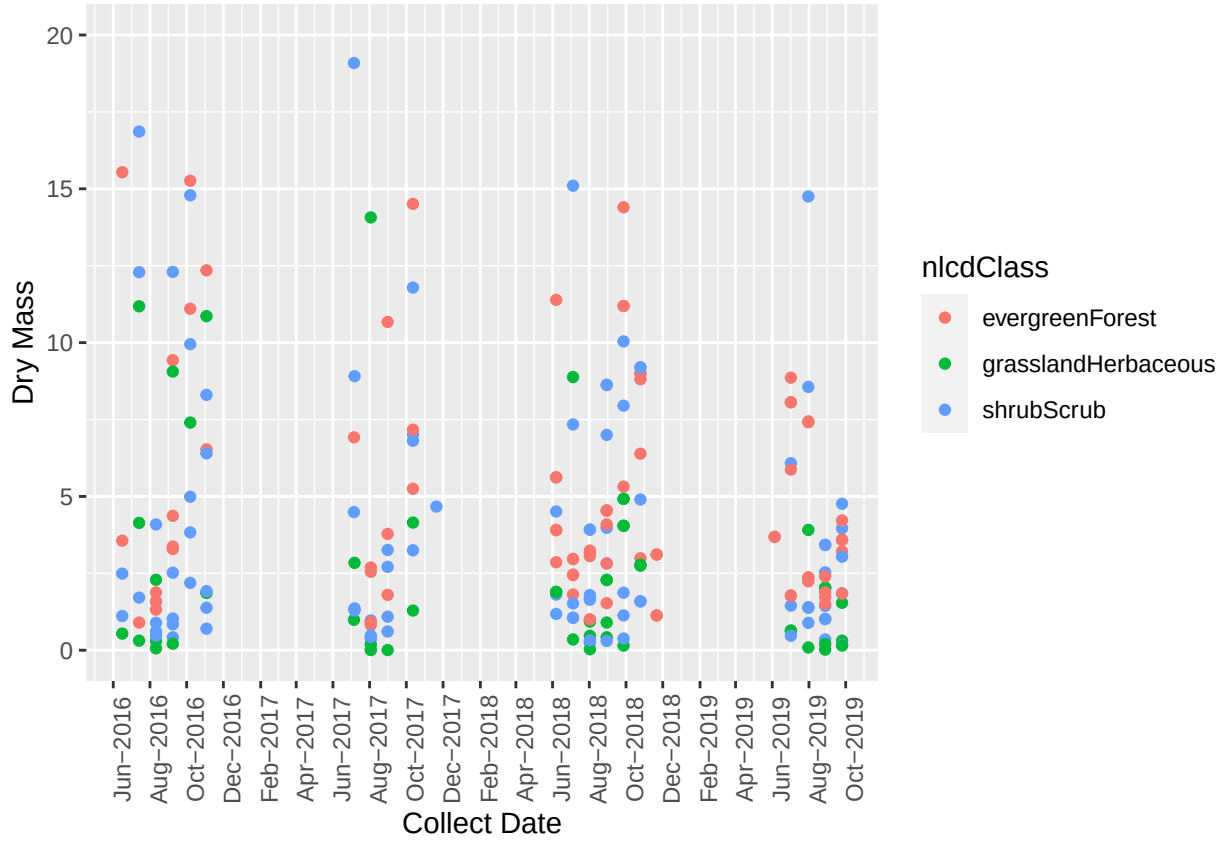
> **Answer:** For both lakes and for all variables, there are observations only for the warm season (between May and September for tp_ug and tn_ug, and between May and November for temperature). For the seasons with data, Paul Lake tends to have higher median temperatures compared to Peter Lake (except for October and November), while Peter Lake shows higher medium values for tp_ug and tn_ug than Paul Lake through the season.

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.

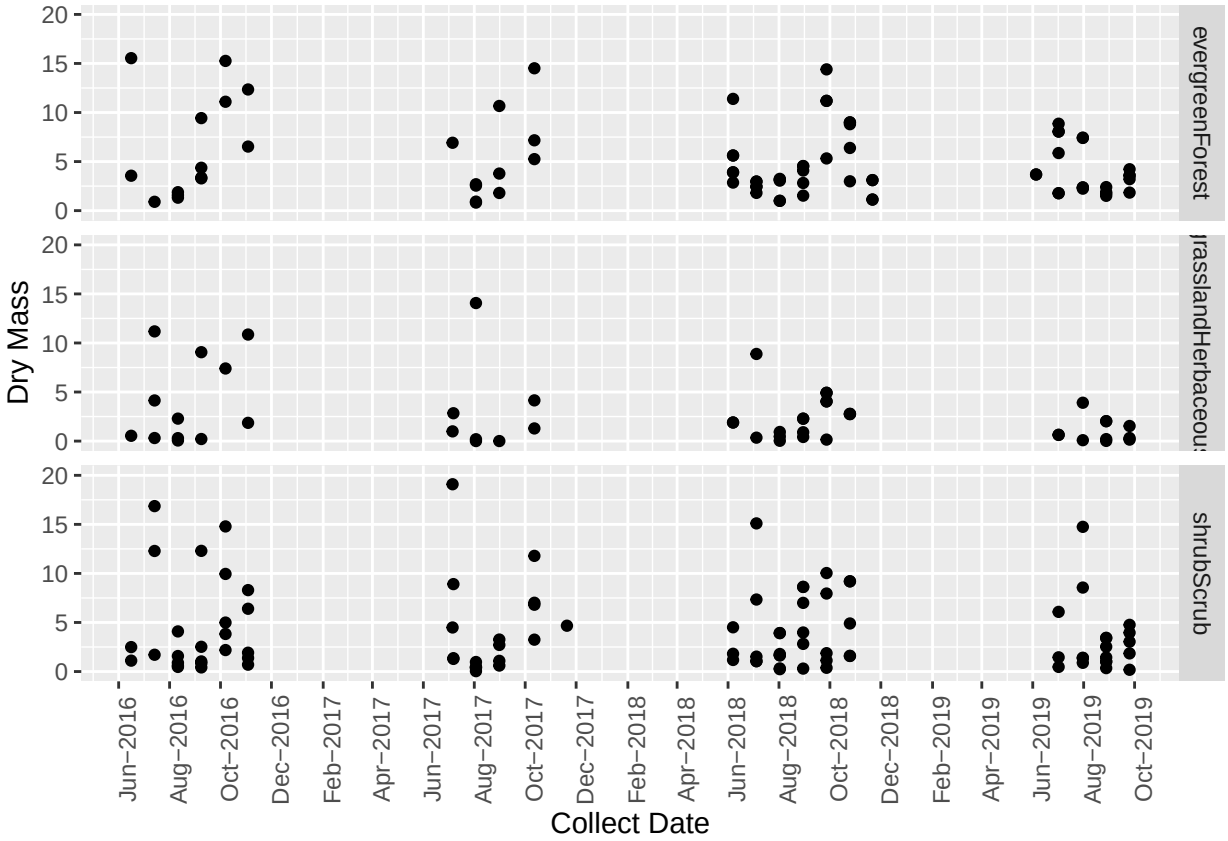
```
# 6

Needles_plot <- ggplot(subset(Niwot_Ridge_litter_dataset,
    functionalGroup == "Needles"), aes(x = collectDate,
    y = dryMass)) + geom_point(aes(colour = nlcdClass)) +
    xlab("Collect Date") + ylab("Dry Mass") +
    ylim(0, 20) + scale_x_date(date_breaks = "2 months",
    date_labels = "%b-%Y") + theme(axis.text.x = element_text(angle = 90))

print(Needles_plot)
```

```
## Warning: Removed 15 rows containing missing values (geom_point).
```




```
# 7

Needles_plot_faceted <- ggplot(subset(Niwot_Ridge_litter_dataset,
    functionalGroup == "Needles"), aes(x = collectDate,
    y = dryMass)) + geom_point() + facet_grid("nlcdClass") +
    xlab("Collect Date") + ylab("Dry Mass") +
    ylim(0, 20) + scale_x_date(date_breaks = "2 months",
    date_labels = "%b-%Y") + theme(axis.text.x = element_text(angle = 90))

print(Needles_plot_faceted)
```

```
## Warning: Removed 15 rows containing missing values (geom_point).
```




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

**Answer:** In my opinion, the most effective graph to show how dry mass evolves according to the collection date by nlcd class is the faceted one. In the first graph, although dots from different classes are differentiated by color, many of them overlap, so we are not seeing some observations that could be important to understand this relationship. On the other hand, in the second graph, we can clearly see the dots for each nlcd class and compare between them.