

# Assignment 5: Data Visualization

Griffin Bird

Spring 2023

## OVERVIEW

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

## Directions

1. Rename this file `<FirstLast>_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. 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
setwd("C:/Users/griff/OneDrive/Documents/EDA-Spring2023")
getwd()
```

```
## [1] "C:/Users/griff/OneDrive/Documents/EDA-Spring2023"
```

```
library(knitr)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr  1.1.0
```

```
## v tidyr 1.3.0 v stringr 1.5.0
## v readr 2.1.3 v forcats 1.0.0
## -- 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(here)
```

```
## here() starts at C:/Users/griff/OneDrive/Documents/EDA-Spring2023
```

```
library(cowplot)
```

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

```
library(ggplot2)
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
## theme_map
```

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=80), tidy=TRUE)
```

```
PeterPaul.data <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
NIWOT.litter.data <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors=FALSE)
```

```
#2
NIWOT.litter.data$collectDate <- as_date(NIWOT.litter.data$collectDate)
PeterPaul.data$sampldate <- as_date(PeterPaul.data$sampldate)
```

## 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_wsj() + theme(plot.title = element_text(hjust = 0.5),
  panel.grid.minor = element_line(color = 2,
    size = 0.25,
    linetype = 1),
  legend.position = "bottom",
  axis.text = element_text(face="plain", size = 12),
  axis.title=element_text(size=14),
  title = element_text(size=14))
```

```
## Warning: The 'size' argument of 'element_line()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
```

```
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

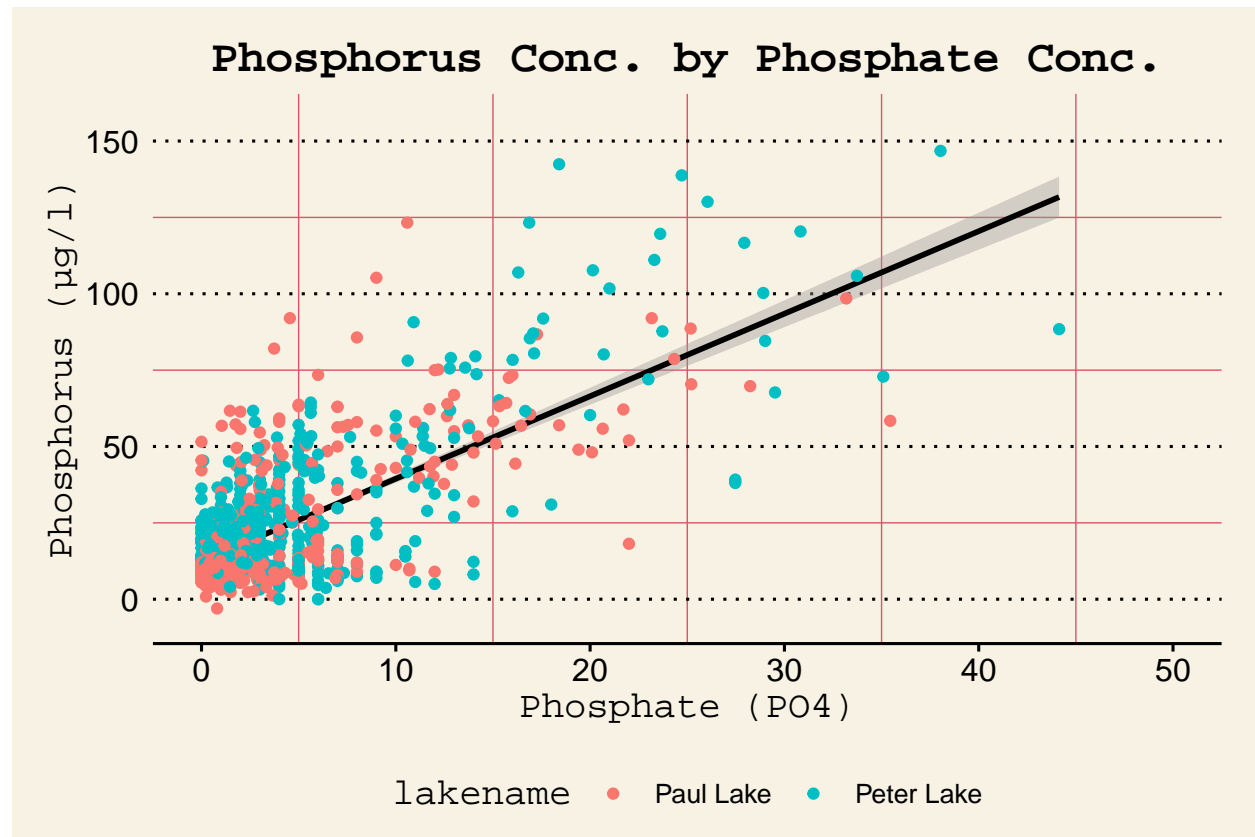
```
phos.plot <- ggplot(PeterPaul.data, aes(x = po4, y = tp_ug, color = lakename)) +
  xlim(0, 50) +
  labs(title = "Phosphorus Conc. by Phosphate Conc.",
    x = "Phosphate (P04)",
    y = "Phosphorus (µg/l)") +
  geom_smooth(method = lm, color = "black") +
  geom_point()

print(phos.plot)
```

```
## '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.

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

```
#5

temp.plot <- ggplot(PeterPaul.data, aes(x = month.abb[month], y = temperature_C, color = lakename)) +
  labs(x = "",
       y = "Temperature (C)") +
  theme(legend.position = "none") +
  scale_x_discrete(limits = month.abb[5:11]) +
  geom_boxplot()

TP.plot <- ggplot(PeterPaul.data, aes(x = sort(month.abb[month]), y = tp_ug, color = lakename)) +
  labs(x = "",
       y = "Total Phosphorus (µg/l)") +
  theme(legend.position = "none") +
  scale_x_discrete(limits = month.abb[5:11]) +
  geom_boxplot()

TN.plot <- ggplot(PeterPaul.data, aes(x = month.abb[month], y = tn_ug, color = lakename)) +
  labs(x = "Month",
```

```
    y = "Total Nitrogen (µg/l)" +  
    scale_x_discrete(limits = month.abb[5:11]) +  
    geom_boxplot()
```

```
plot_grid(temp.plot, TP.plot, TN.plot, ncol=1, align = "vh")
```

```
## Warning: Removed 16 rows containing missing values ('stat_boxplot()').
```

```
## Warning: Removed 3550 rows containing non-finite values ('stat_boxplot()').
```

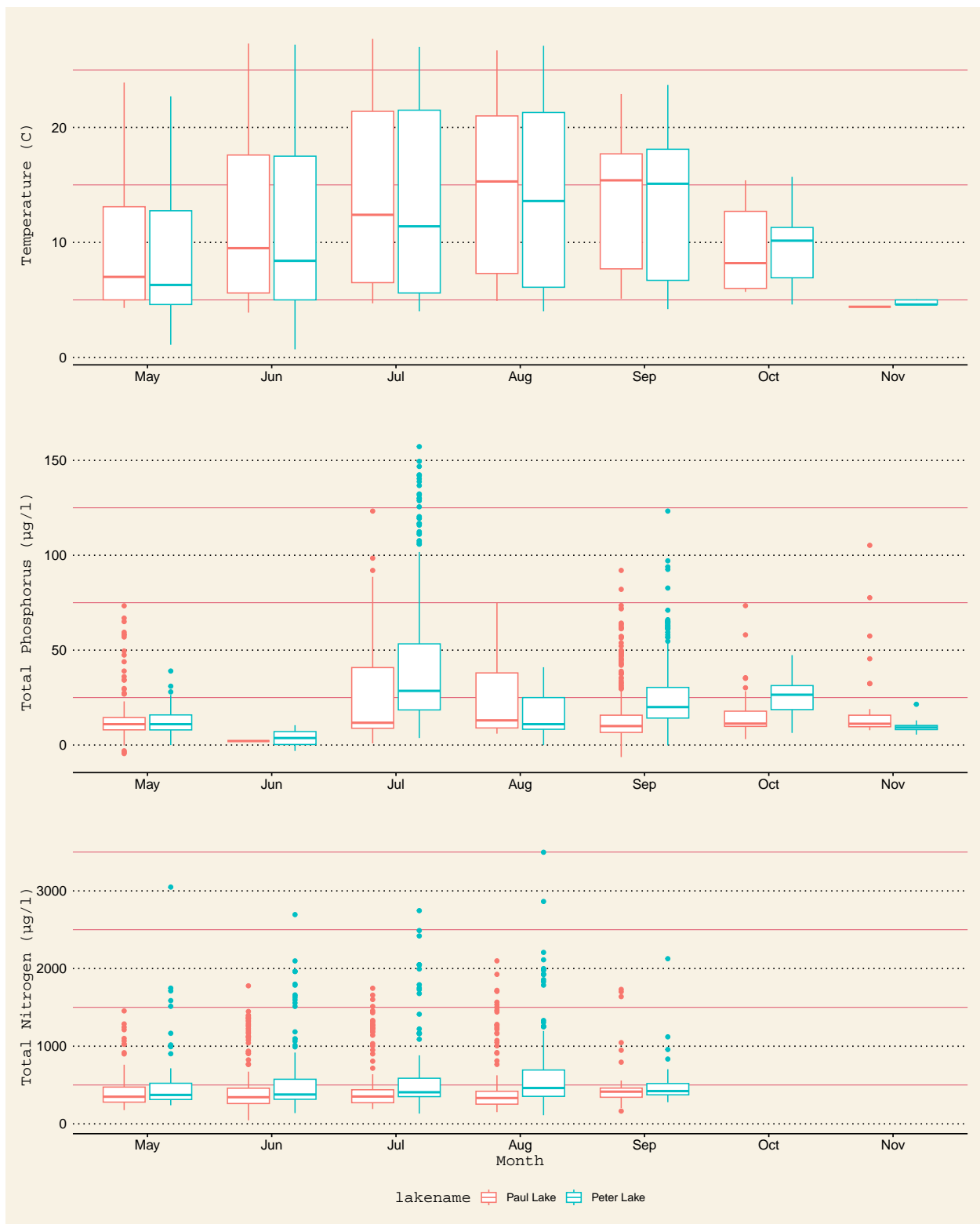
```
## Warning: Removed 16 rows containing missing values ('stat_boxplot()').
```

```
## Warning: Removed 20715 rows containing non-finite values ('stat_boxplot()').
```

```
## Warning: Removed 16 rows containing missing values ('stat_boxplot()').
```

```
## Warning: Removed 21567 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?

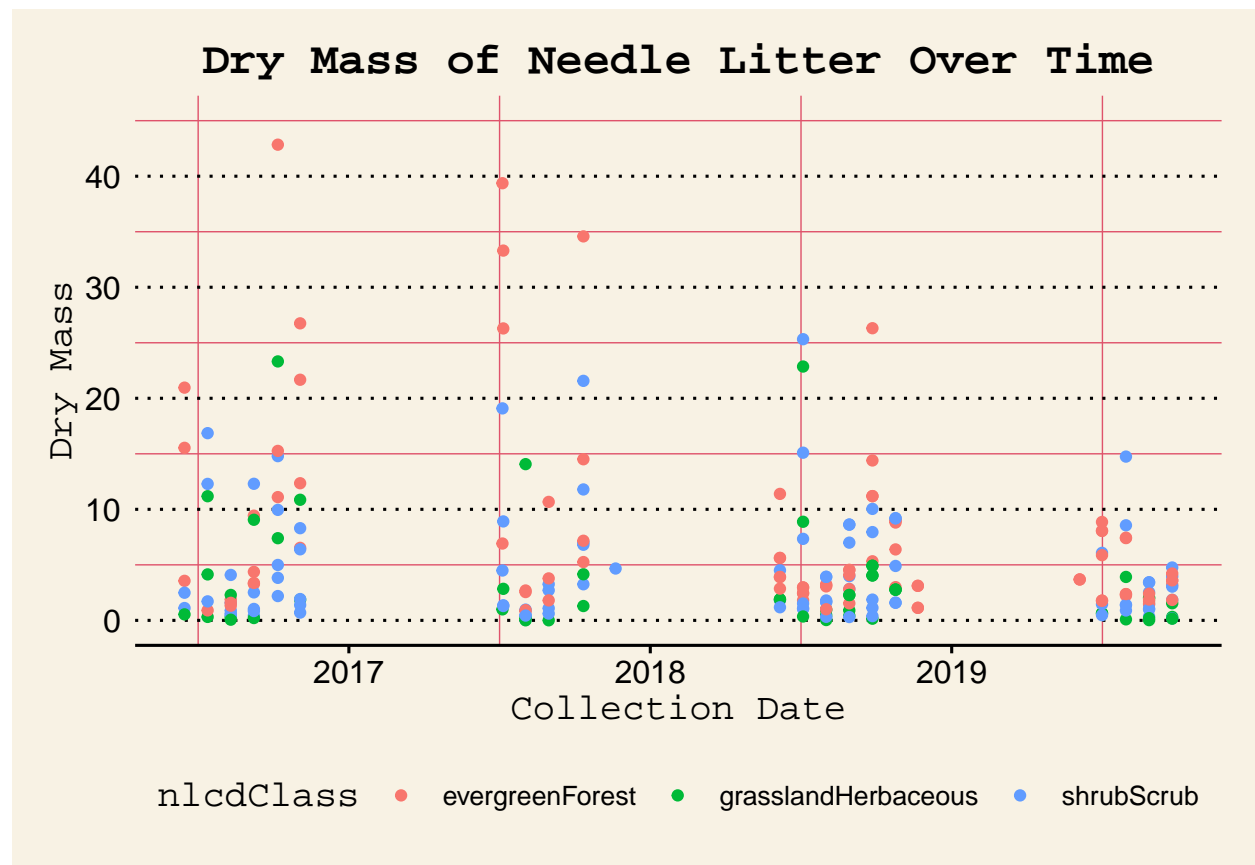
Answer: All of the variables trend upwards over the warmer summer months. There isn't much temperature difference between the lakes. But it does look like Peter lake might receive more phosphorus and nitrogen pollution than Paul lake.

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

```
NIWOT.litter.data %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  ylim(0,45) +
  labs(title = "Dry Mass of Needle Litter Over Time",
       x = "Collection Date",
       y = "Dry Mass") +
  geom_point()
```

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

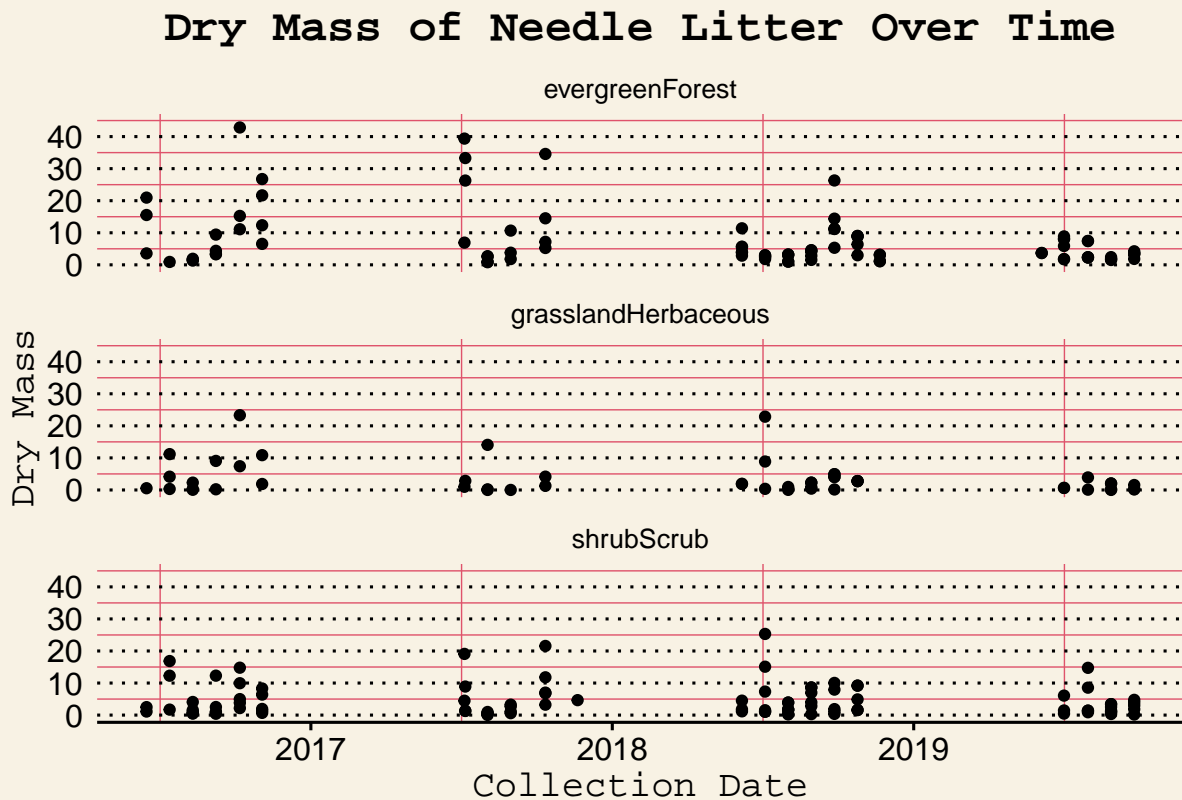


#7

```
NIWOT.litter.data %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  ylim(0,45) +
```

```
labs(title = "Dry Mass of Needle Litter Over Time",
     x = "Collection Date",
     y = "Dry Mass") +
facet_wrap(vars(nlcdClass), nrow = 3) +
geom_point()
```

## Warning: Removed 2 rows containing missing values ('geom\_point()').



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

Answer: The faceted plot is more effective. The dry mass data is so clustered together that its hard to see detail when nlcd classes are differentiated by color. But the faceted graph gives the data room to spread out and is easier to interpret.