Assignment 5: Data Visualization

Aidan Power

Fall 2024

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. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

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

```r
#1
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```r
library(cowplot)
```

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

```r
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
Peter.Paul.chem.nutr <-
  read.csv(here("Data","Processed",
                "Processed_KEY",
                "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)
Neon.Niwo <-
  read.csv(here("Data","Processed","Processed_KEY",
                "NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)
#2
class(Peter.Paul.chem.nutr$sampledate)
```

```
## [1] "factor"
```

```r
class(Neon.Niwo$collectDate)
```

```
## [1] "factor"
```

```r
Peter.Paul.chem.nutr$sampledate <- ymd(Peter.Paul.chem.nutr$sampledate)
Neon.Niwo$collectDate <- ymd(Neon.Niwo$collectDate)
class(Peter.Paul.chem.nutr$sampledate)
```

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

```r
class(Neon.Niwo$collectDate)
```

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

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
my_theme <-
  theme(
    plot.background = element_rect(
      color='purple',
      fill='gold'
    ),
    plot.title = element_text(
      color='tan'
    )

  )
theme_set(my_theme)
```

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.

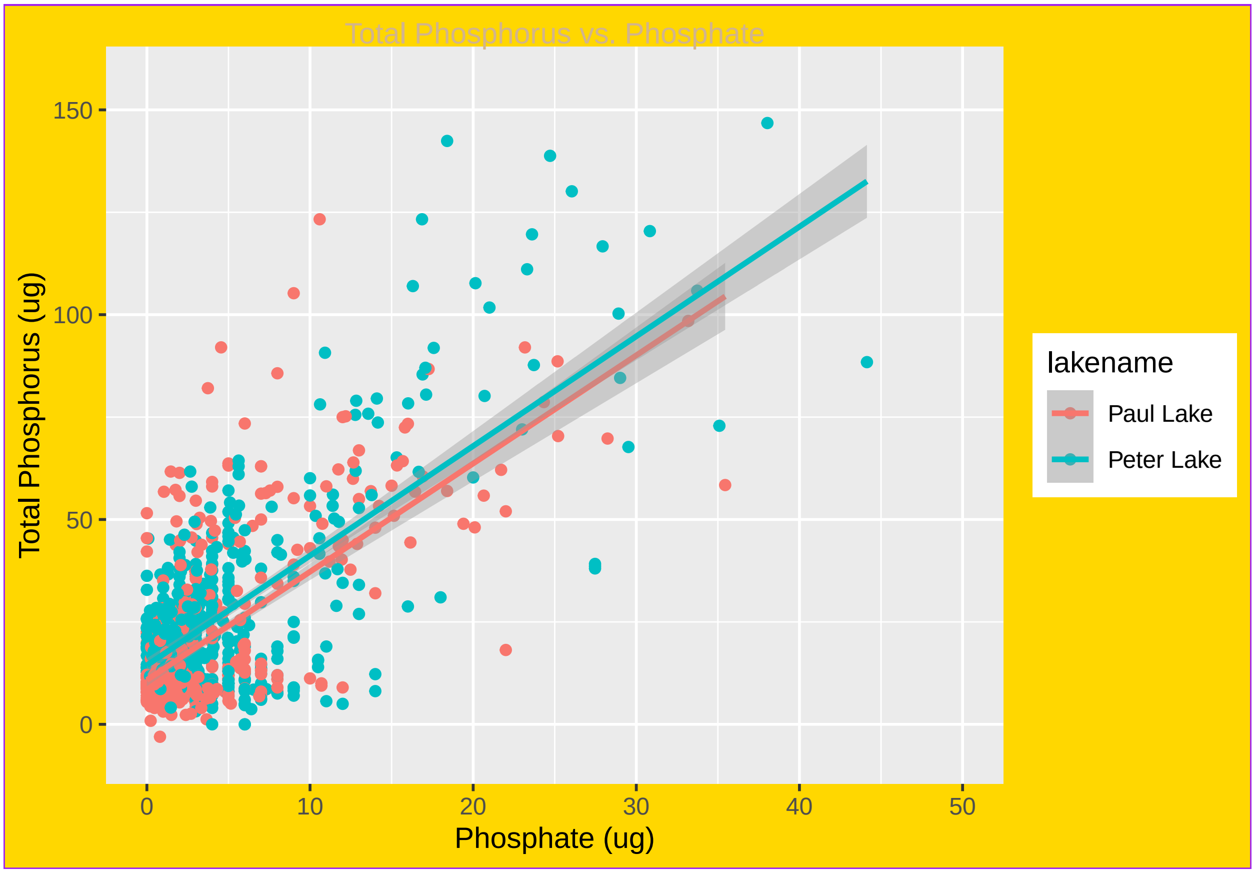
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Peter.Paul.chem.nutr %>%
  ggplot(aes(x=po4,y=tp_ug,color=lakename)) +
  geom_point()+
  geom_smooth(method="lm")+
  xlim(0,50) +
  labs(title="Total Phosphorus vs. Phosphate",
       x="Phosphate (ug)",
       y="Total Phosphorus (ug)")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 21947 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (‘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.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
Peter.Paul.chem.nutr$month <- factor(Peter.Paul.chem.nutr$month)
levels(Peter.Paul.chem.nutr$month) <- c(levels(Peter.Paul.chem.nutr$month),
                                        "1","2","3","4","5","9","10","11","12")
levels(Peter.Paul.chem.nutr$month)

Peter.Paul.chem.nutr$month <-
  factor(Peter.Paul.chem.nutr$month,
         levels = c("1","2","3","4","5","6","7","8","9","10","11","12"))
#Made sure that I had all 12 months



Peter.Paul.chem.nutr %>%
  mutate(month = factor(month,levels = 1:12, labels = month.abb))
tp1 <- Peter.Paul.chem.nutr %>%
  ggplot() +
  geom_boxplot(aes(
    x=month,
    y=temperature_C,
    color=lakename),show.legend=F) +
    scale_x_discrete(drop = FALSE)
tp2 <- Peter.Paul.chem.nutr %>%
  ggplot() +
  geom_boxplot(aes(
    x=month,
    y=tp_ug,
    color=lakename)) +
     scale_x_discrete(drop = FALSE)
 tp3 <- Peter.Paul.chem.nutr %>%
  ggplot() +
  geom_boxplot(aes(
    x=month,
    y=tn_ug,
    color=lakename), show.legend=F) +
    scale_x_discrete(drop = FALSE)

 tp1_revised <- tp1 +
   labs (title = NULL) +
   theme(legend.position = "none") +
   theme(plot.margin = unit(c(1,0,1,0), "cm")) +
   theme(axis.text.x = element_text(size = 4),
         axis.text.y = element_text(size = 4),
         axis.title.y = element_text(size = 6),
         axis.title.x = element_text(size=6))

  tp2_revised <- tp2 +
   labs (title = NULL) +
   theme(legend.position = "none") +
   theme(plot.margin = unit(c(1,0,1,0), "cm")) +
   theme(axis.text.x = element_text(size = 4),
         axis.text.y = element_text(size = 4),
         axis.title.y = element_text(size = 6),
         axis.title.x = element_text(size=6))

   tp3_revised <- tp3 +
   labs (title = NULL) +
   theme(legend.position = "none") +
   theme(plot.margin = unit(c(1,0,1,0), "cm")) +
   theme(axis.text.x = element_text(size = 4),
         axis.text.y = element_text(size = 4),
         axis.title.y = element_text(size = 6),
         axis.title.x = element_text(size=6))
 #Prepare for one legend and one title

legend <- get_legend(tp2_revised + theme(legend.position = "right",
                                         legend.title = element_text(size = 6),
                                         legend.text = element_text(size=4)))
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use `return_all = TRUE`.
```

```
#new legend
plot_all <- plot_grid(tp1_revised,tp2_revised,tp3_revised,legend,
                      nrow=1,align = 'h',rel_widths = c(1,1,1,0.7)) +
  draw_label("Temp,TP & TN vs. Month for Peter and Paul Lake",
             size=10,x=0.5,y=0.95,hjust=0.5) #revised cowplot with new title
```

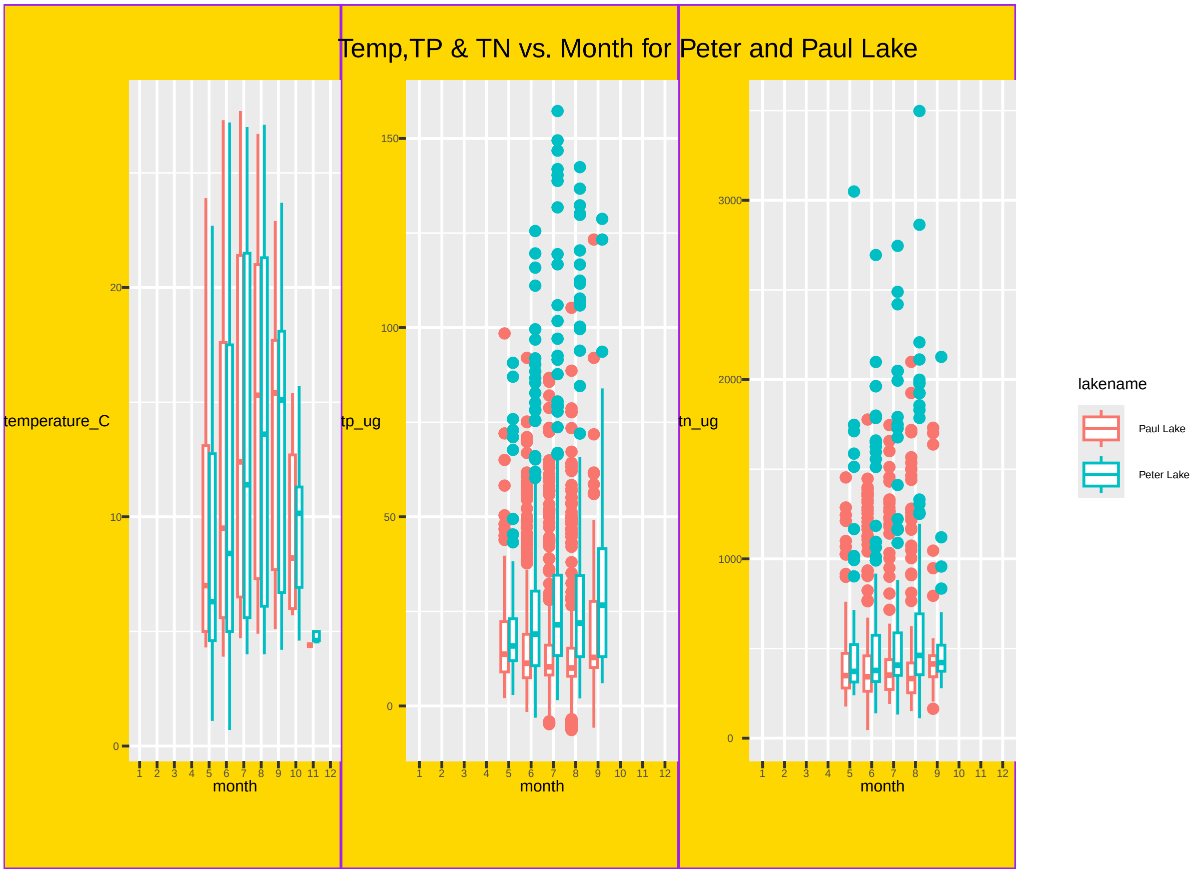
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
show(plot_all)
```

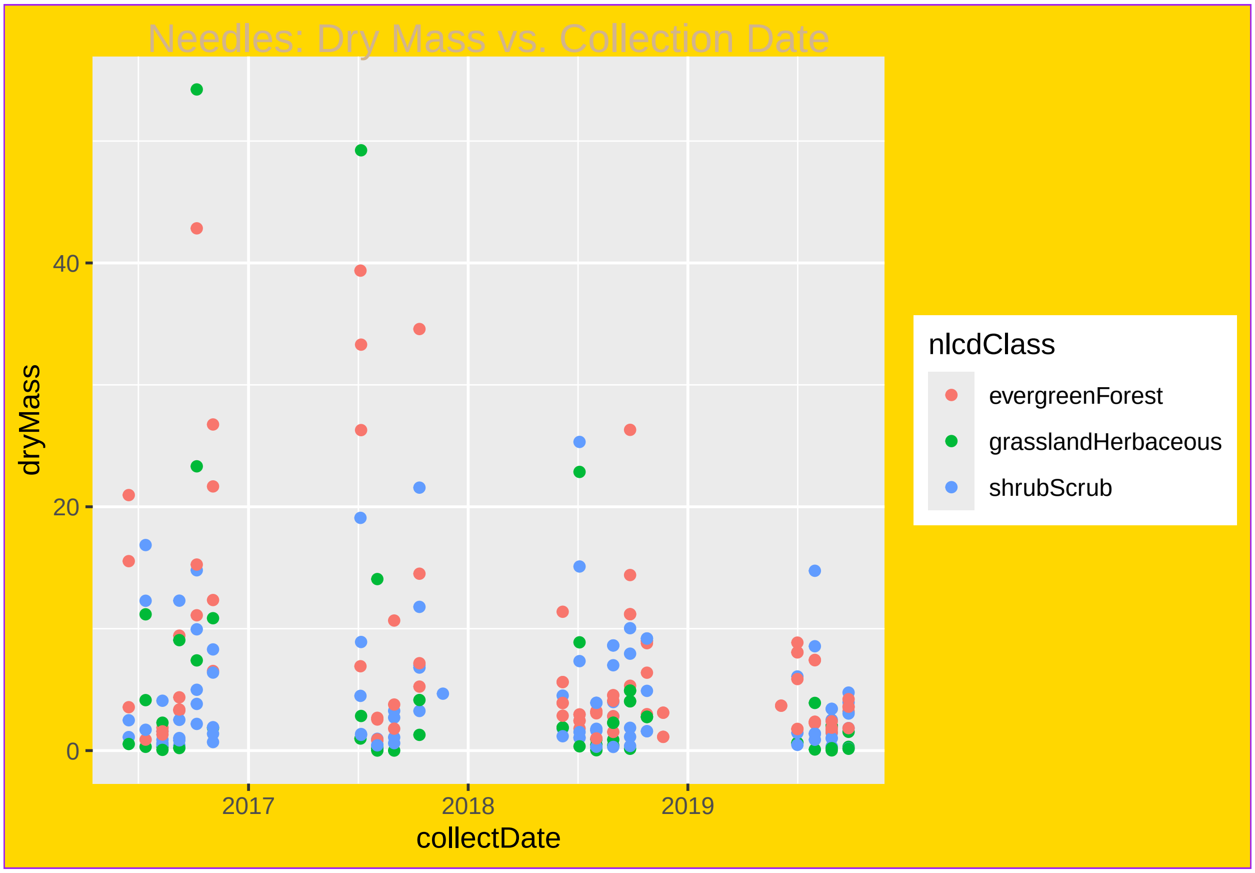


Question: What do you observe about the variables of interest over seasons and between lakes?

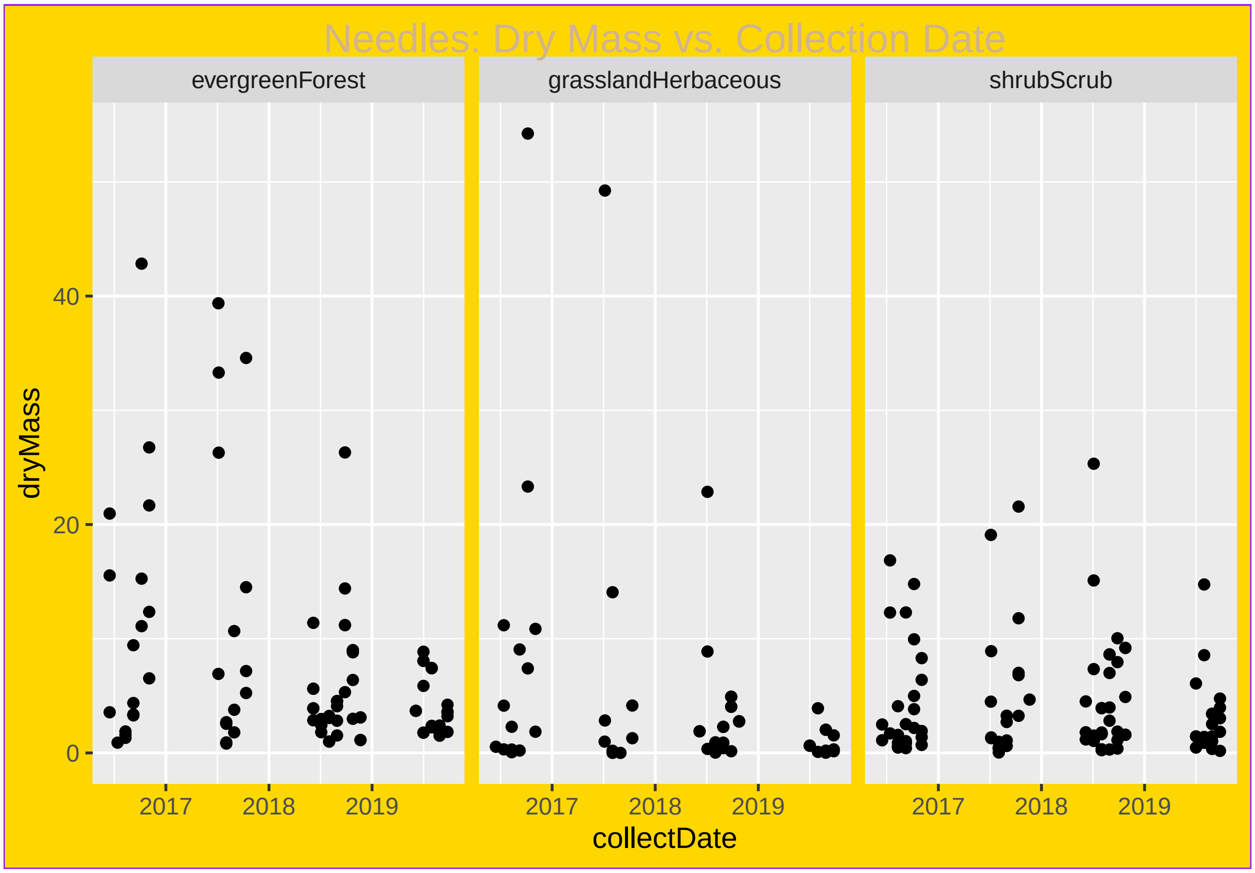
Answer: I observed that Peter lake is higher for both total nutrients while temeprature data is relatively even between the two. This makes sense as they are both lakes in summertime.

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
ggplot(subset(Neon.Niwo, functionalGroup == "Needles")) +
  geom_point(aes(x=collectDate,y=dryMass, color = nlcdClass)) +
  labs(title = "Needles: Dry Mass vs. Collection Date") +
  theme(plot.title = element_text(size=15))
```



```
#7
ggplot(subset(Neon.Niwo, functionalGroup == "Needles")) +
  geom_point(aes(x=collectDate,y=dryMass)) +
  facet_wrap(vars(nlcdClass),nrow=1) +
  labs(title = "Needles: Dry Mass vs. Collection Date") +
  theme(plot.title = element_text(size=15))
```



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

Answer: I believe that the second one is more effective due to how visually appealing it makes the analysis between the classes.