Assignment 5: Data Visualization

Grace Randall

Fall 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. 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.

```
#1
#bring in packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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 errors
```

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

```
## here() starts at /home/guest/module1/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

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

```
#load data
peterpaul_chem <- read.csv(
  here("Data","Processed_KEY",
       "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)
litter <- read.csv(
  here("Data","Processed_KEY",
       "NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
#fix dates for lakes
class(peterpaul_chem$sampledate)
```

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

```
peterpaul_chem$sampledate <- ymd(peterpaul_chem$sampledate)

#fix months for lakes
peterpaul_chem$month <-
  factor(
    peterpaul_chem$month,
    levels=c("1","2","3","4","5","6","7","8","9","10","11","12"),
                               labels = month.abb)

#filter litter for needles
litter <- filter(litter, functionalGroup == "Needles")
#fix litter dates
litter$collectDate <- ymd(litter$collectDate)
```

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_base()  +
  theme(
    #line =              element_line(),
    #rect =              element_rect(),
    #text =              element_text(),

    # Modified inheritance structure of text element
    plot.title =        element_text(color = "midnightblue"),
    axis.title.x =      element_text(color = "midnightblue"),
    axis.title.y =      element_text(color = "midnightblue"),
    #axis.text =         element_text(),

    # Modified inheritance structure of line element
    #axis.ticks =        element_line(),
    panel.grid.major =  element_line(color="white"),
    #panel.grid.minor =  element_blank(),

    # Modified inheritance structure of rect element
    #plot.background =   element_rect(),
    panel.background =  element_rect(fill = "lightskyblue1"),
    #legend.key =        element_rect(),

    # Modifiying legend.position
    #legend.position = 'top',

    #complete = TRUE
    )
```

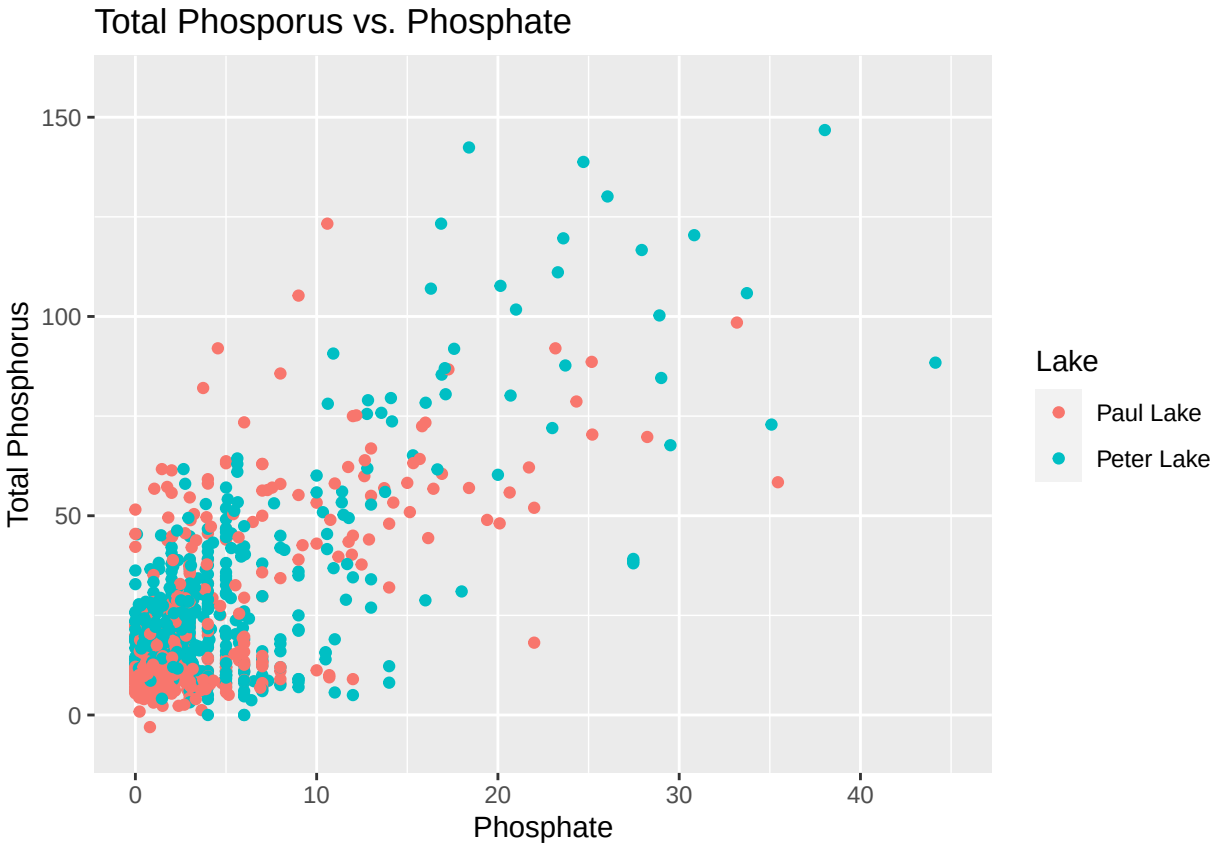
Create graphs

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
ggplot(peterpaul_chem, aes(x=po4,y=tp_ug,color=lakename))+
  xlim(0,45)+ #get rid of extreme values
  labs(title = "Total Phosporus vs. Phosphate", color = "Lake")+ #set labels
  xlab("Phosphate")+
  ylab("Total Phosphorus")+
  geom_point()
```

```
## 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: * Recall the discussion on factors in the previous section as it may be helpful here. * 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

#making temperature plot
 temp<-ggplot(peterpaul_chem)+
  theme(legend.position="none")+
  geom_boxplot(aes(x=month,y=temperature_C,color=lakename))+
  ylab("Temp (c)")+
  scale_x_discrete(drop=FALSE)

#making tp plot
 tp<-ggplot(peterpaul_chem)+
  theme(legend.position="none")+
  ylab("Total Phosphorus (um)")+
  geom_boxplot(aes(x=month,y=tp_ug,color=lakename))+
  scale_x_discrete(drop=FALSE)

 #making tn plot
 TN<-ggplot(peterpaul_chem)+
  theme(legend.position="none")+
  ylab("Total Nitrogen (um)")+
```

```
 geom_boxplot(aes(x=month,y=tn_ug,color=lakename))+
 scale_x_discrete(drop=FALSE)

#grabing the legend
shared_legend <-get_legend(TN + theme(legend.position = "bottom"))
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

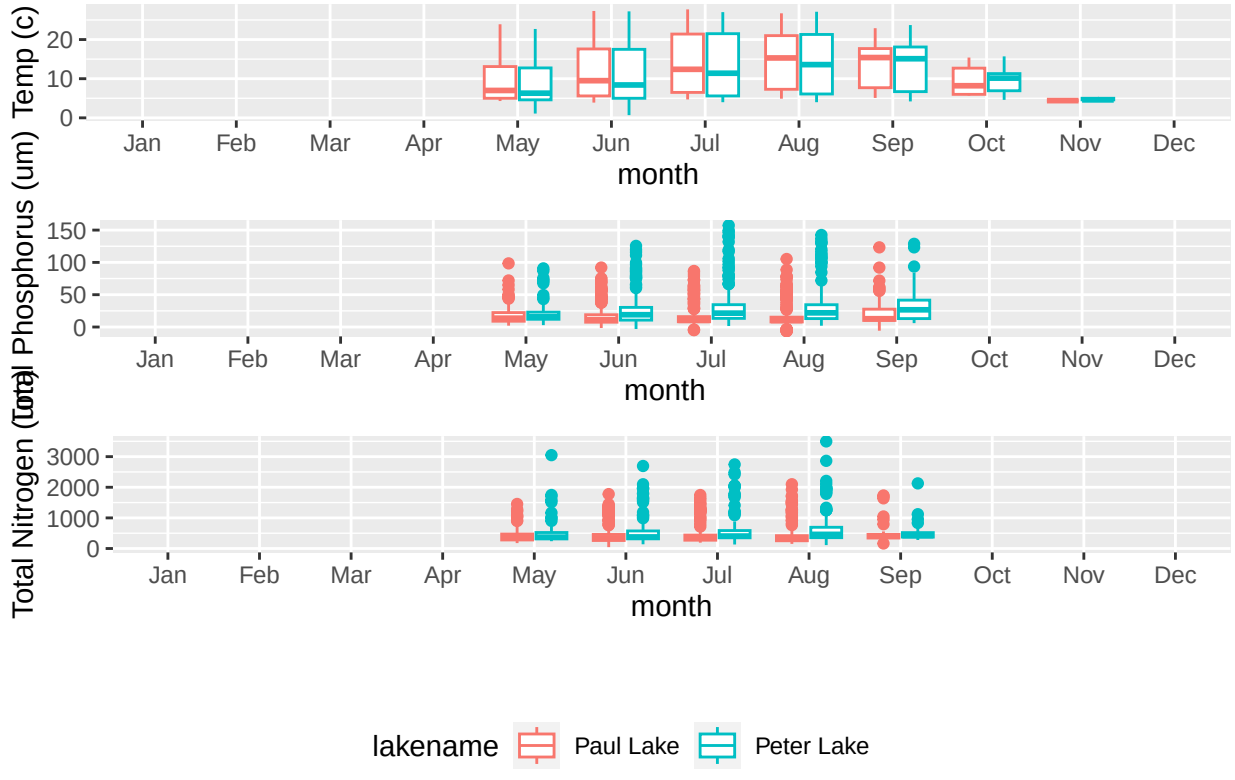
```
#plotting all the parts
plot_grid(temp,tp,TN,shared_legend,nrow=4,align = 'h')
```

```
## 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()`).

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



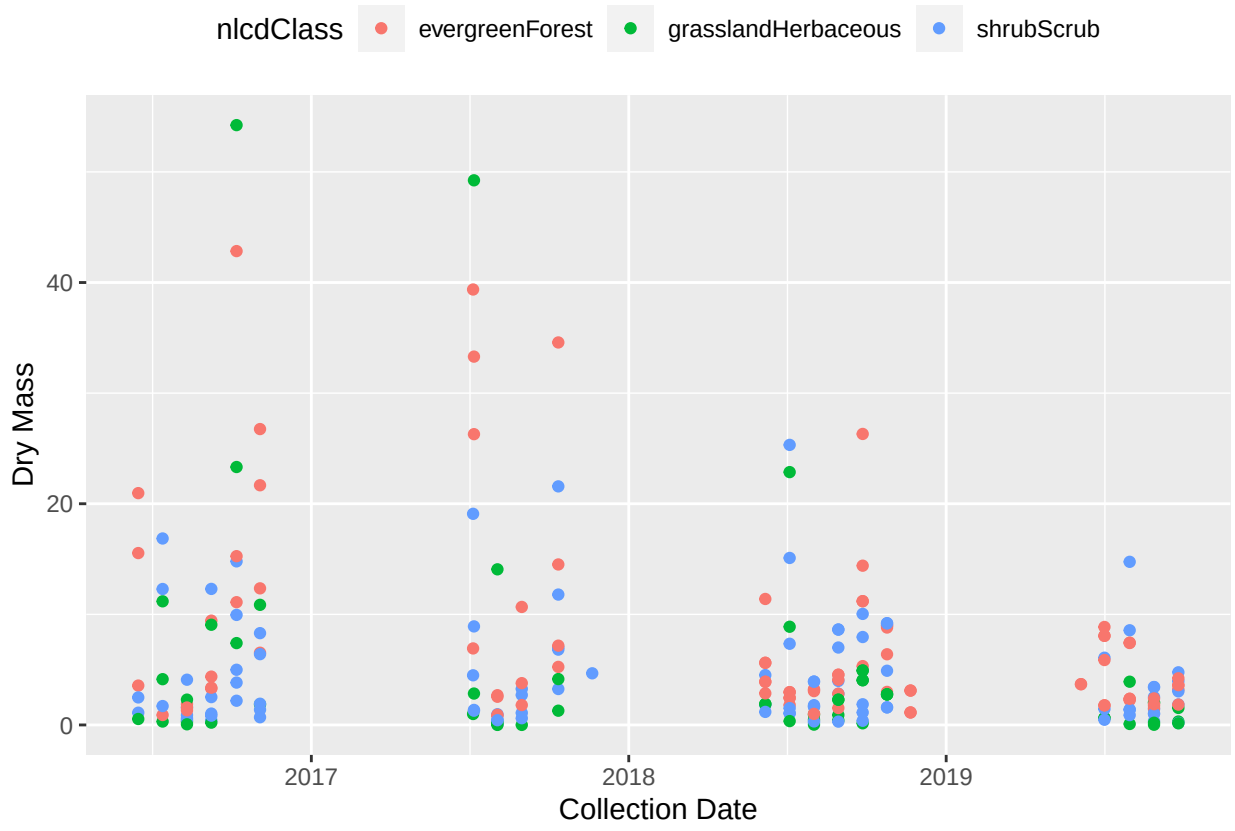
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: temperatures are clearly higher in both lakes in the summer months. there is not as clear of a seasonal trend in the nutrients but both Nitrogen and Phosphorus are higher in peter lake.

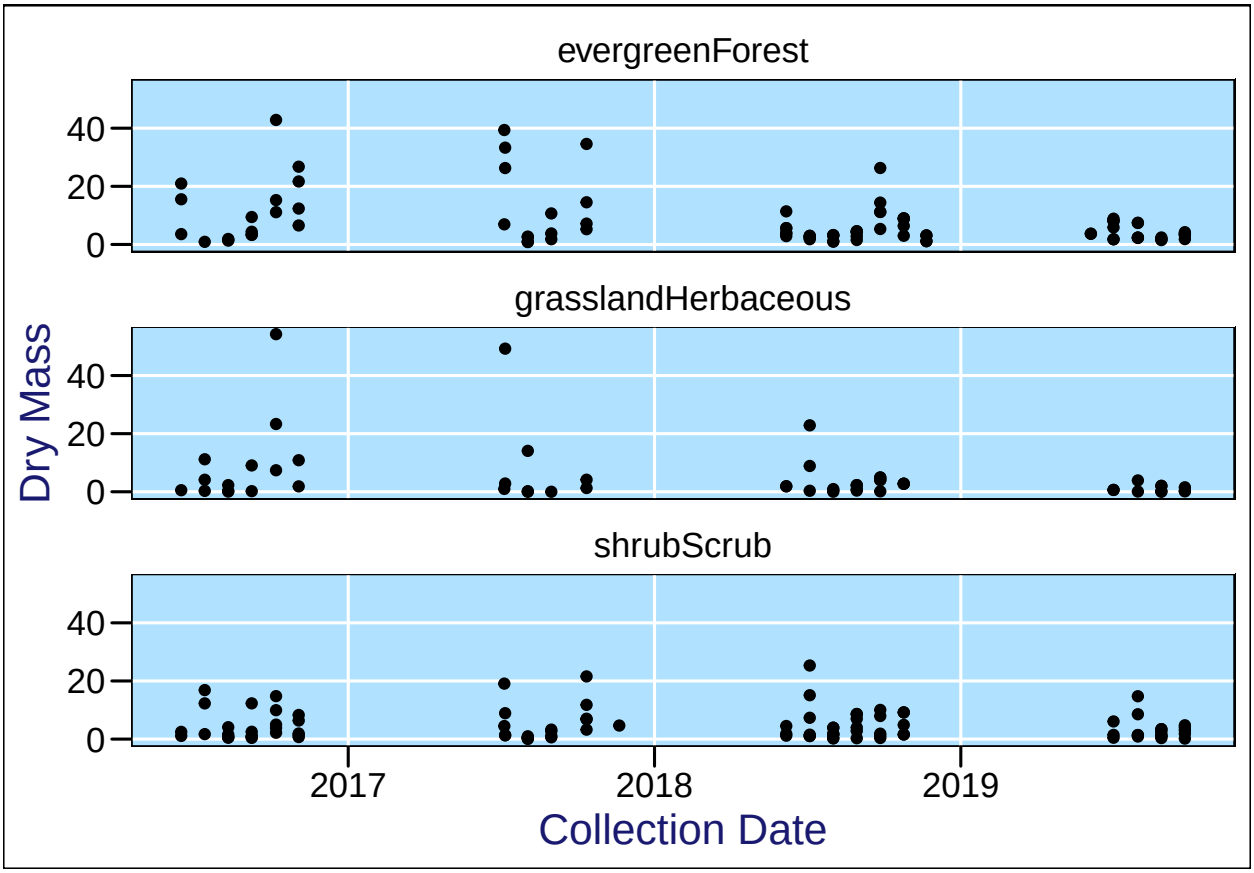
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
ggplot(litter,aes(x=collectDate,y=dryMass,color=nlcdClass))+
  xlab("Collection Date")+
  ylab("Dry Mass")+
  theme(legend.position = "top")+
  geom_point()
```




```
#7
ggplot(litter,aes(x=collectDate,y=dryMass))+
  xlab("Collection Date")+
  ylab("Dry Mass")+
  my_theme+
  facet_wrap(vars(nlcdClass),nrow=3)+
      geom_point()
```



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

Answer: I think that 7 is more effective because it is very difficult to see what is happening with each of the ecosystems when they are all plotted on top of each other.