Assignment 5: Data Visualization

David Robinson

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

#Load necessary packages

#install.packages("tidyverse")
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
```

```
#install.packages("lubridate")
library(lubridate)
#install.packages("here")
library(here)
```

```
## here() starts at C:/Users/dhr20/OneDrive - Duke University/1 - Academics/1 - First Year/1 - Fall 2023
```

```
#install.packages("cowplot")
library(cowplot)
```

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

```
#install.packages("ggthemes")
library(ggthemes)
```

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

```
#2

#Read in the data sets in question
Nutrients <- read.csv(file = here('Data','Processed_KEY', 'NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_

Litter <- read.csv(file = here('Data','Processed_KEY', 'NEON_NIWO_Litter_mass_trap_Processed.csv'), str

#Make sure R is reading the dates as date format
Nutrients$sampledate <- ymd(Nutrients$sampledate)
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

#Customize axis ticks / gridlines
#Customize legend
#Customize background fill
mytheme <- theme_base() +
  theme(
    line = element_line(
      color = 'blue',
      linewidth = 1
    ),
    legend.background = element_rect(
      color = 'black',
    ),
    legend.title = element_text(
      color = 'purple'
    ),
    axis.text = element_text(
      size = 10
    ),
    plot.background = element_rect(
      color = 'grey', fill = 'grey'
    ))

#Apply / set the theme that was just defined
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

#Create scatter plot of TP by PO4 per the instructions
ggplot(Nutrients,
       aes(
         y = po4,
         x = tp_ug,
         color = lakename
       )
       ) +
  geom_point() +
  xlim(0, 150) +
  ylim(0, 50) +
```

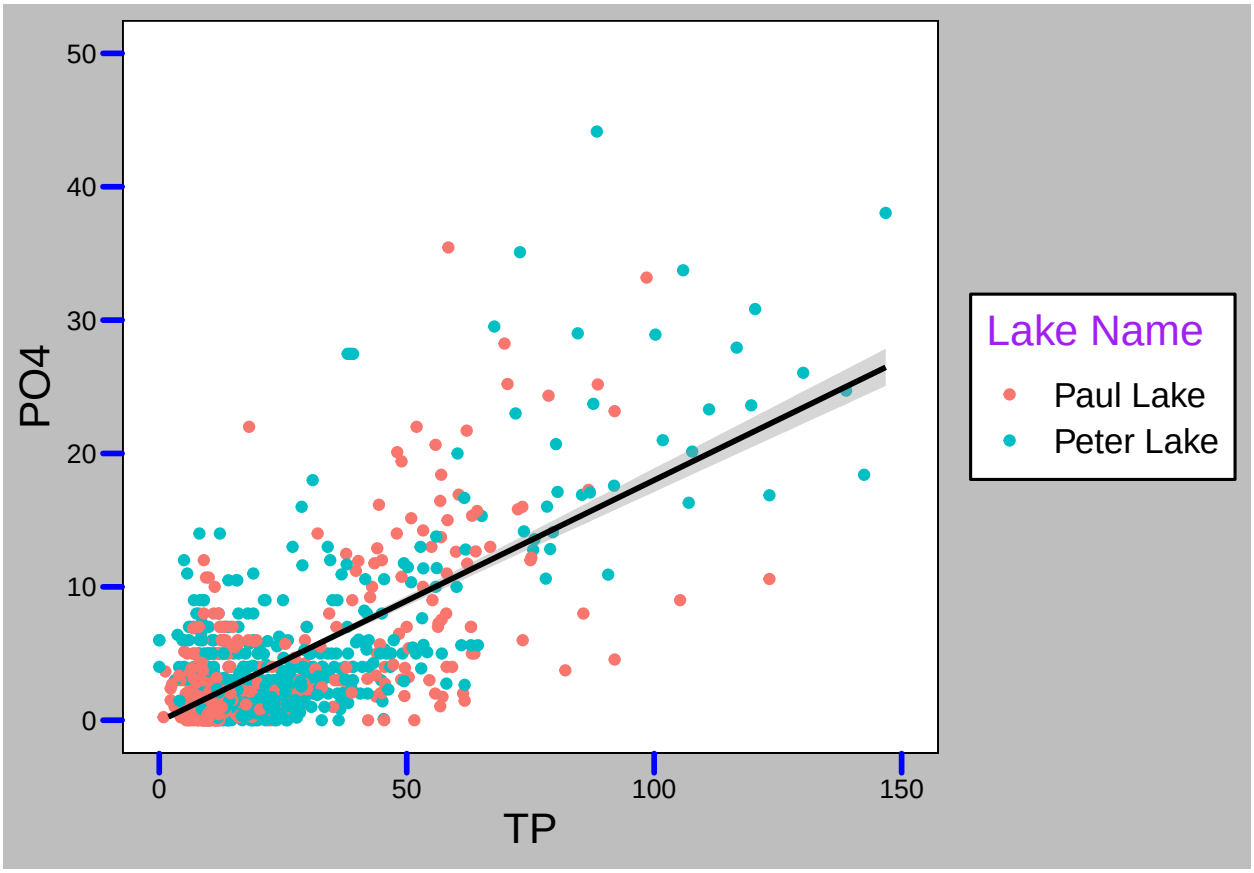
```
  geom_smooth(method= lm, color = "black") +
  labs(y = 'PO4', x = 'TP', color = 'Lake Name')
```

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

```
## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21948 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 1 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: * 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

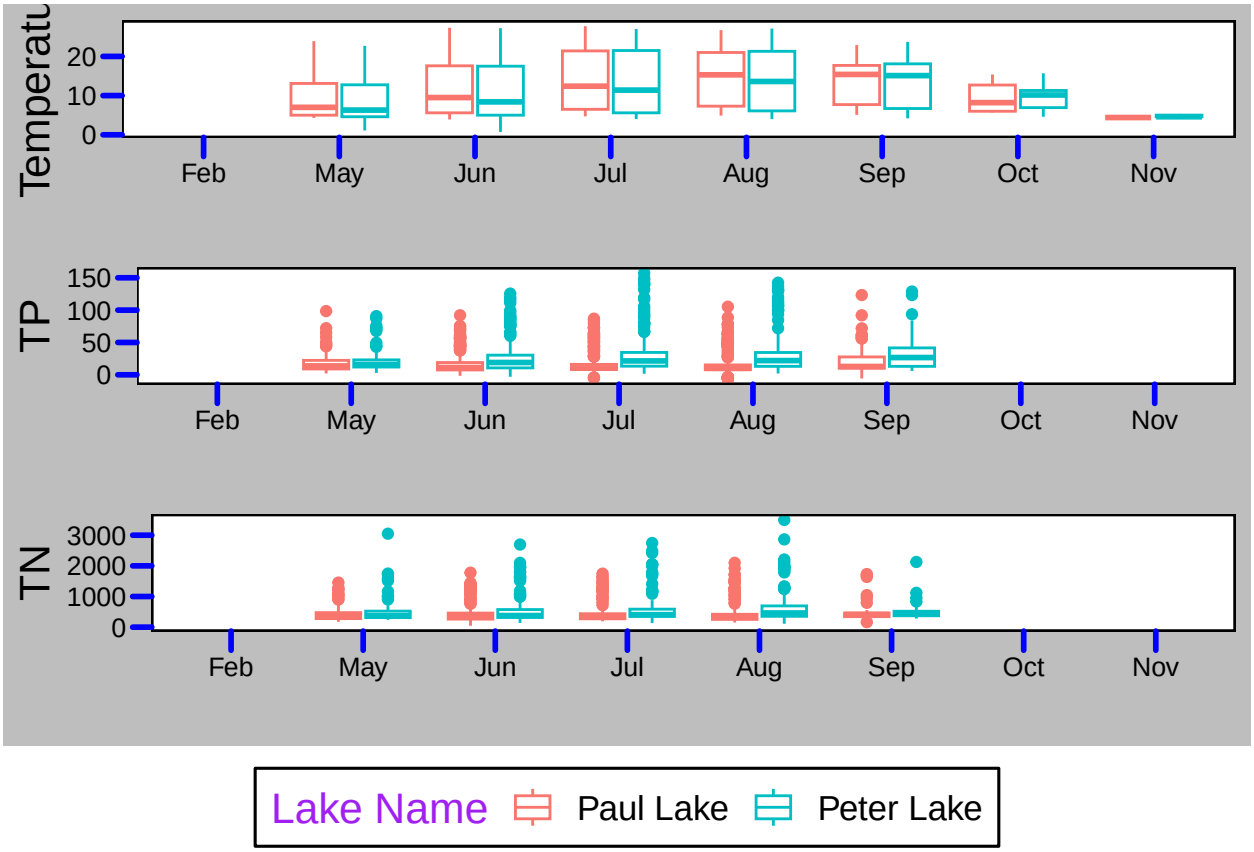
```
#5

#Create boxplot for temperature per the instructions -- note the legend is positioned for the
#later get_legend() call in the cowplot
boxplot_temp <- ggplot(Nutrients,
```

```
       aes(x = factor(month,levels = 1:12, labels = month.abb),
           y = temperature_C
           )) +
         geom_boxplot(aes(color = lakename)) +
  labs(y = 'Temperature', x = '', color = 'Lake Name') +
  theme(legend.position='bottom')

#Create boxplot for TP per the instructions
boxplot_TP <- ggplot(Nutrients,
       aes(x = factor(month,levels = 1:12, labels = month.abb),
           y = tp_ug
           )) +
         geom_boxplot(aes(color = lakename)) +
  labs(y = "TP", x = "")

#Create boxplot for TN per the instructions
boxplot_TN <- ggplot(Nutrients,
       aes(x = factor(month,levels = 1:12, labels = month.abb),
           y = tn_ug
           )) +
         geom_boxplot(aes(color = lakename)) +
  labs(y = "TN",x = "")

#Create cowplot to align the three boxplots
plot_grid(boxplot_temp + theme(legend.position = 'none'),
          boxplot_TP + theme(legend.position = 'none'),
          boxplot_TN + theme(legend.position = 'none'),
          get_legend(boxplot_temp),
          ncol = 1,
          align = 'vh',
          axis = 'b',
          rel_heights = c(1, 1, 1, .5))
```

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



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

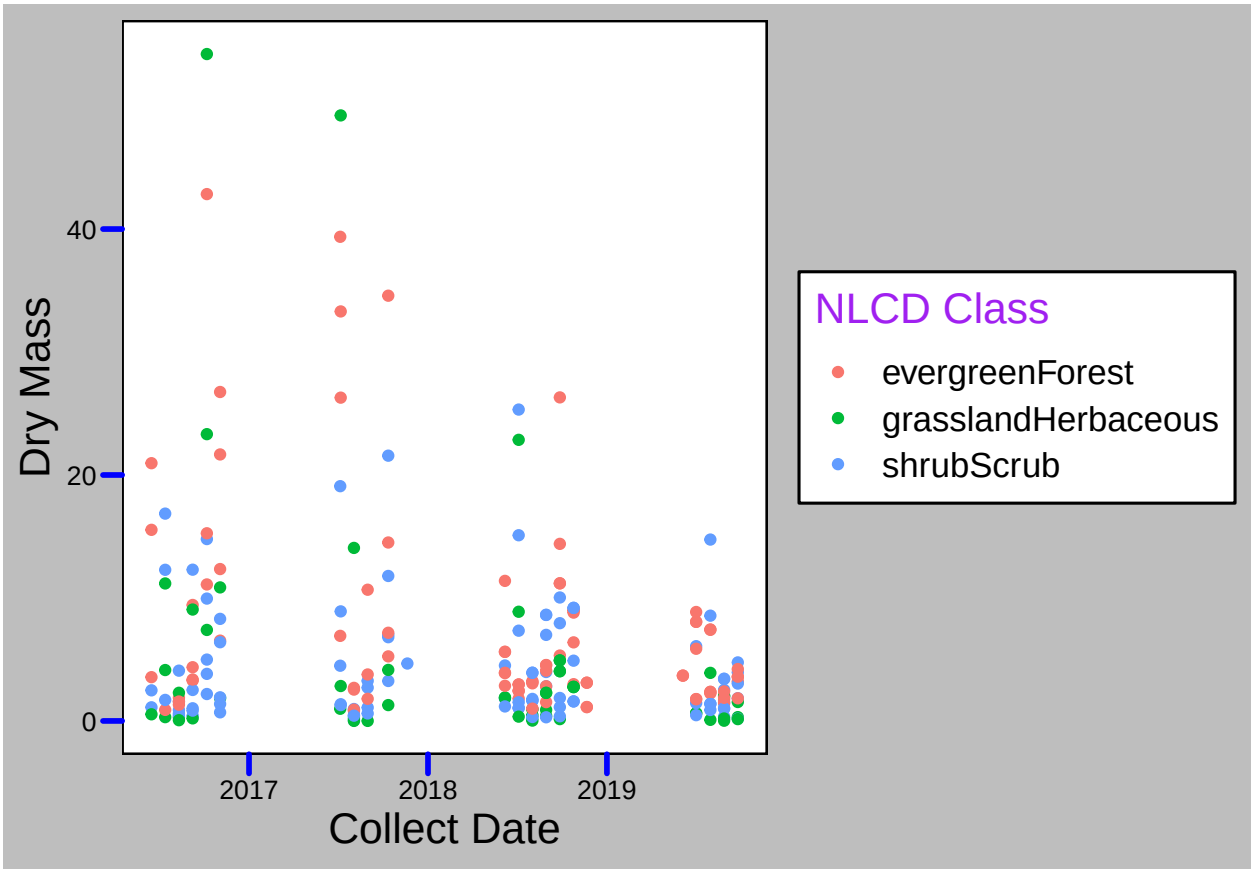
> Answer: Over seasons, temperatures tend to fluctuate as we might expect – in alignment with air temperatures (warmer in the summer months, cooler in the spring and fall months). TP and TN are pretty similar across seasons. Between lakes, Paul Lake has higher median temperatures than Peter Lake. Paul Lake has lower median values than Peter Lake of both TP and TN across all seasons.

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)

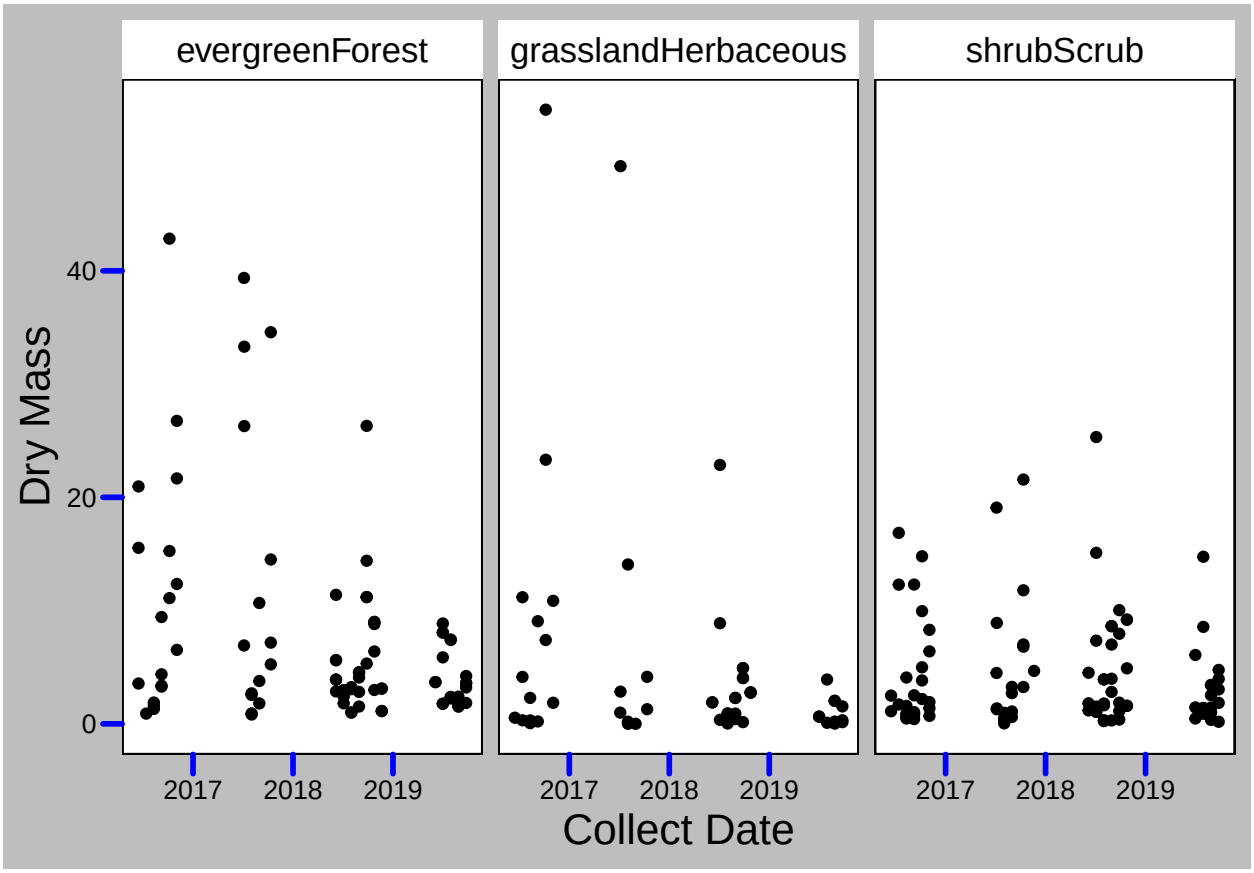
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

#Filter the dataset per the instructions
Litter_needles <- Litter %>%
  filter(functionalGroup == "Needles")

#Create scatterplot per the instructions
ggplot(Litter_needles,
       aes(
         y = dryMass,
         x = collectDate,
```

```
        color = nlcdClass
      )
      ) +
  geom_point() +
  labs(x = "Collect Date", y = "Dry Mass", color = "NLCD Class")
```




```
#7

#Create scatterplot per the instructions including facets by NLCD Class
ggplot(Litter_needles,
       aes(
         y = dryMass,
         x = collectDate
       )
       ) +
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(x = "Collect Date", y = "Dry Mass")
```



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

Answer: I think the plot on #7 is more effective – the facets by NLCD Class enable an easier digestion of the data rather than colors by data point given the large number of data points.