Assignment 5: Data Visualization

Xianhang Xie

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
# Load the necessary packages
library(tidyverse)
```

```
## Warning:   'tidyverse' R 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
```

```
## Warning:   'ggplot2' R 4.1.3
```

```
## Warning:   'tibble' R 4.1.3

## Warning:   'tidyr' R 4.1.3

## Warning:   'readr' R 4.1.3

## Warning:   'purrr' R 4.1.3

## Warning:   'dplyr' R 4.1.3

## Warning:   'stringr' R 4.1.3

## Warning:   'forcats' R 4.1.3

## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Warning:   'lubridate' R 4.1.3

##      timechange

## Warning:   'timechange' R 4.1.3

##
##     'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(here)
```

```
## Warning:   'here' R 4.1.2

## here() starts at C:/Users/11764/Desktop/EDA-Spring2023
```

```
library(ggplot2)
library(cowplot)
```

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

```
# Verify the home directory
getwd()
```

```
## [1] "C:/Users/11764/Desktop/EDA-Spring2023/Assignments"
```

```
#2
peter_paul <- read_csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
```

```
## Rows: 23008 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): lakename
## dbl  (13): year4, daynum, month, depth, temperature_C, dissolvedOxygen, irra...
## date  (1): sampledate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
niwot_litter <- read_csv("../data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
```

```
## Rows: 1692 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): plotID, trapID, functionalGroup, qaDryMass, nlcdClass, plotType, g...
## dbl  (5): dryMass, subplotID, decimalLatitude, decimalLongitude, elevation
## date (1): collectDate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(peter_paul$sampledate)
```

```
##  Date[1:23008], format: "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-
27" ...
```

```
str(niwot_litter$collectDate)
```

```
##  Date[1:1692], format: "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-
16" ...
```

From the result we know that R is reading dates as data format.

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
# Define the custom theme
custom_theme <- theme(
  plot.background = element_rect(fill = "White"),
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  panel.grid.major = element_line(colour = "gray", linetype = "dashed"),
  panel.grid.minor = element_line(colour = "gray", linetype = "dotted"),

  legend.title = element_text(face = "bold")
)

# Set the custom theme as the default theme
theme_set(custom_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 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()`).

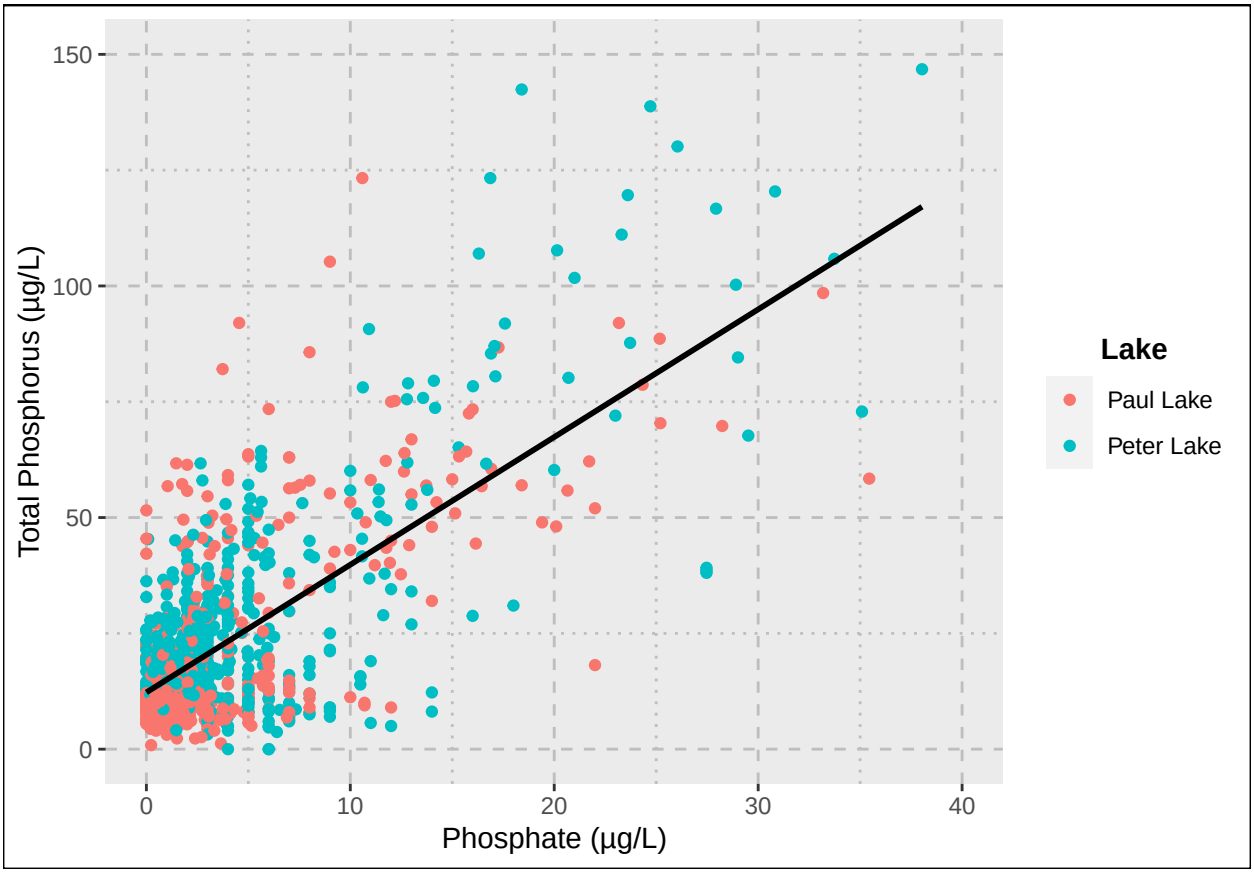
```
#4
tp_po4_plot <- ggplot(peter_paul, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(x = "Phosphate (µg/L)", y = "Total Phosphorus (µg/L)", color = "Lake")

# Adjust the axes to hide extreme values
tp_po4_plot + xlim(0, 40) + ylim(0, 150)
```

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

## Warning: Removed 21949 rows containing non-finite values (stat_smooth).

## Warning: Removed 21949 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
peter_paul$monthabb = month.abb[peter_paul$month]
peter_paul$monthabb = factor(peter_paul$monthabb, levels = month.abb)
temp_plot <- ggplot(peter_paul, aes(x = monthabb, y = temperature_C, fill = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = "Temperature (°C)", fill = "Lake")

tp_plot <- ggplot(peter_paul, aes(x = monthabb, y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = "Total Phosphorus (µg/L)", fill = "Lake")

tn_plot <- ggplot(peter_paul, aes(x = monthabb, y = tn_ug, fill = lakename)) +
  geom_boxplot() +
  labs(x = "Month", y = "Total Nitrogen (µg/L)", fill = "Lake")

# Combine the three plots
legend <- get_legend(
  temp_plot
)
```

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

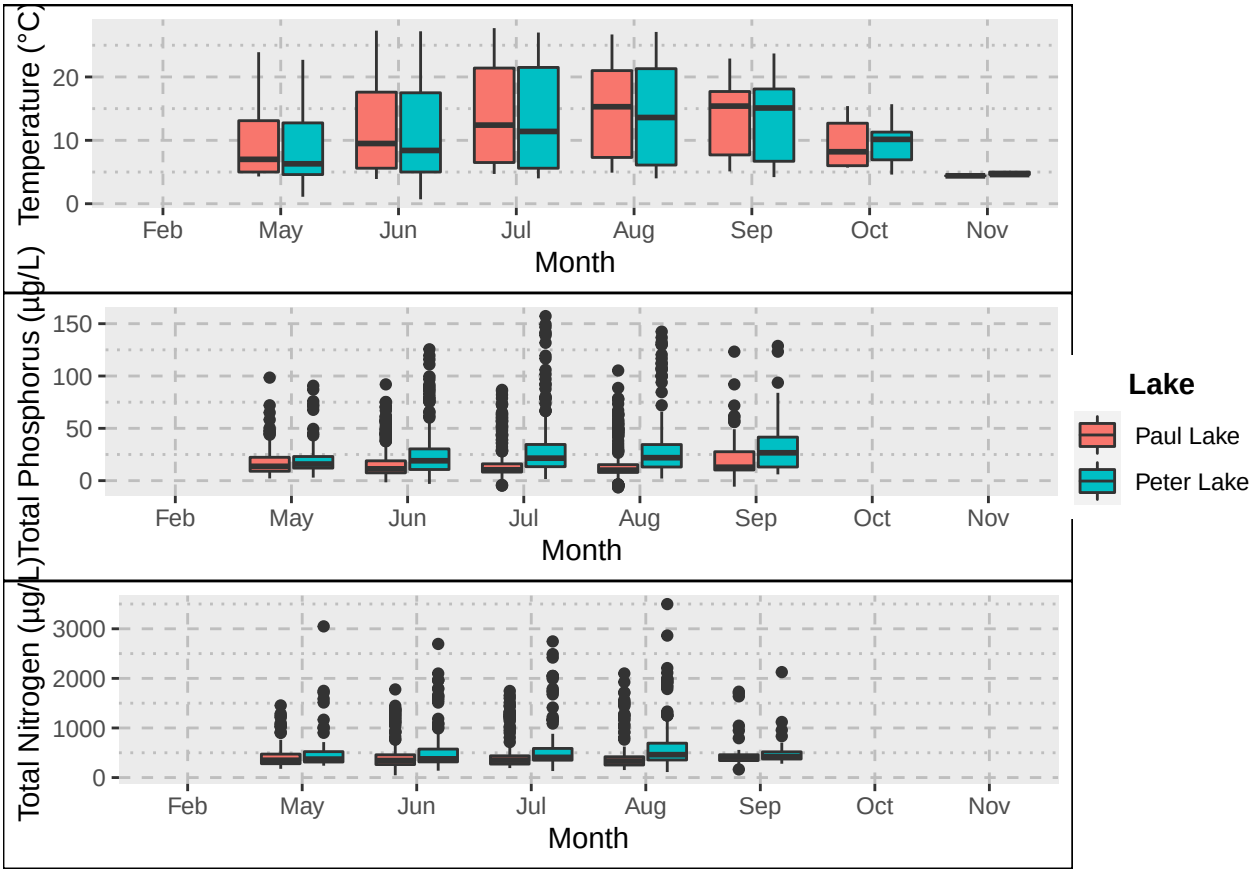
```
combined_plot <- cowplot::plot_grid(
  temp_plot + theme(legend.position="none"),
  tp_plot + theme(legend.position="none"),
  tn_plot + theme(legend.position="none"),
  nrow = 3, align = "h", axis = "lr")
```

```
## 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).
```

```
plot_grid(combined_plot, legend, rel_widths = c(3, .5))
```



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

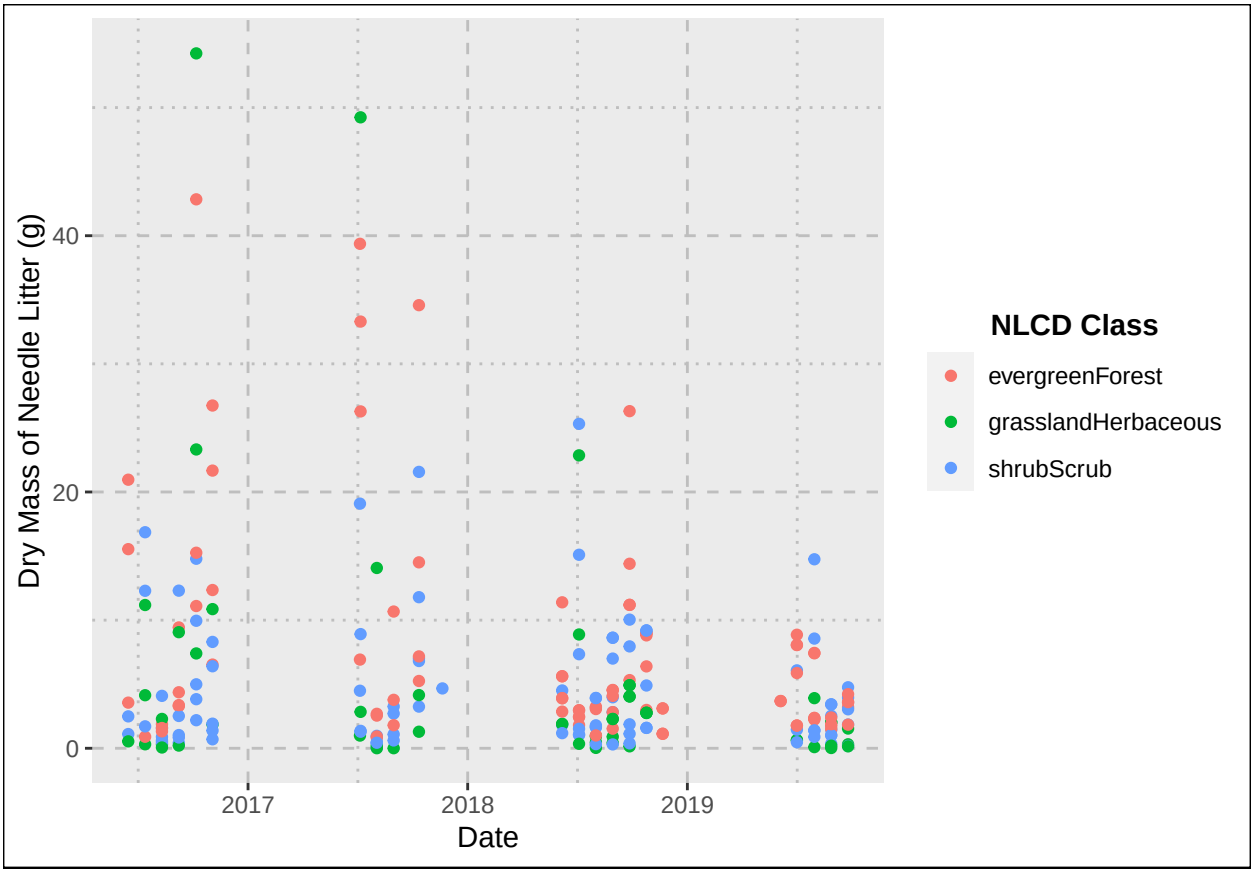
> Answer: For the variable temperature: it tends to be higher in Paul Lake than in Peter Lake throughout May to August, similar in september, and lower then Peter lake in Oct and Nov. The highest temperature is in september.
>
> For the variable Total Phosphorus: it tends to be higher in Peter Lake than in Paul Lake throughout the year. There is a little bit increase from July, Augest to September.

For the variable Total Nitrogen: it tends to be higher in Peter Lake than in Paul Lake throughout the year. There is no significant trend throught the year.

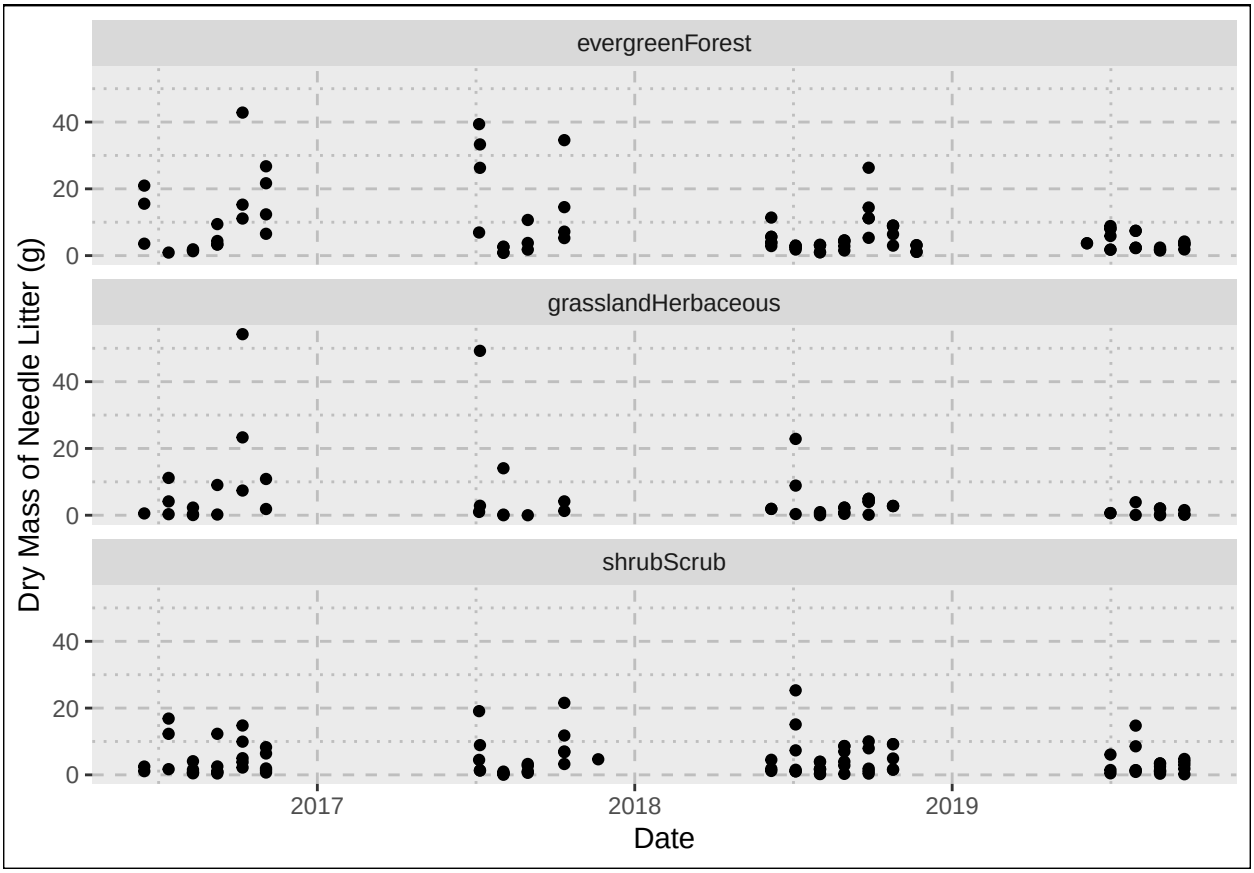
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
needles <- filter(niwot_litter, functionalGroup == "Needles")
ggplot(needles, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(x = "Date", y = "Dry Mass of Needle Litter (g)", color = "NLCD Class")
```




```
#7
ggplot(needles, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  labs(x = "Date", y = "Dry Mass of Needle Litter (g)", color = "NLCD Class") +
  facet_wrap(~ nlcdClass, nrow = 3)
```



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

Answer: I think the plot 7 is more effective. Because with NLCD classes separated by color (plot 6), it is effective in showing how the dry mass of needle litter varies over time across different land cover types. But the result shows there is no apparent difference between different land cover types, so the color cue cannot support anything.

But for the plot 7, with NLCD classes separated into three facets, is effective in showing how the dry mass of needle litter varies over time within each NLCD class. And the plot 7 clearly shows that trend.