Assignment 5: Data Visualization

Yuxiang Ren

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
library(tidyverse);library(lubridate);library(here);library(cowplot)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     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   2.1.4     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
```

```
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
## here() starts at /Users/min/Desktop/EDA-Spring2023/R/EDA
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(ggplot2)
here() #verify home directory
```

```
## [1] "/Users/min/Desktop/EDA-Spring2023/R/EDA"
```

```
#raw data
Lake <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                 stringsAsFactors = T)
Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                   stringsAsFactors = T)

#2
#date format
Lake$sampledate <- ymd(Lake$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
mytheme <- theme_bw(base_size = 14) +
  theme(axis.text = element_text(size = 11, color = "black"),
        axis.title = element_text(size = 13, color = "black"))+
  theme(legend.title = element_text(size = 13, color = "black"))
```

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.

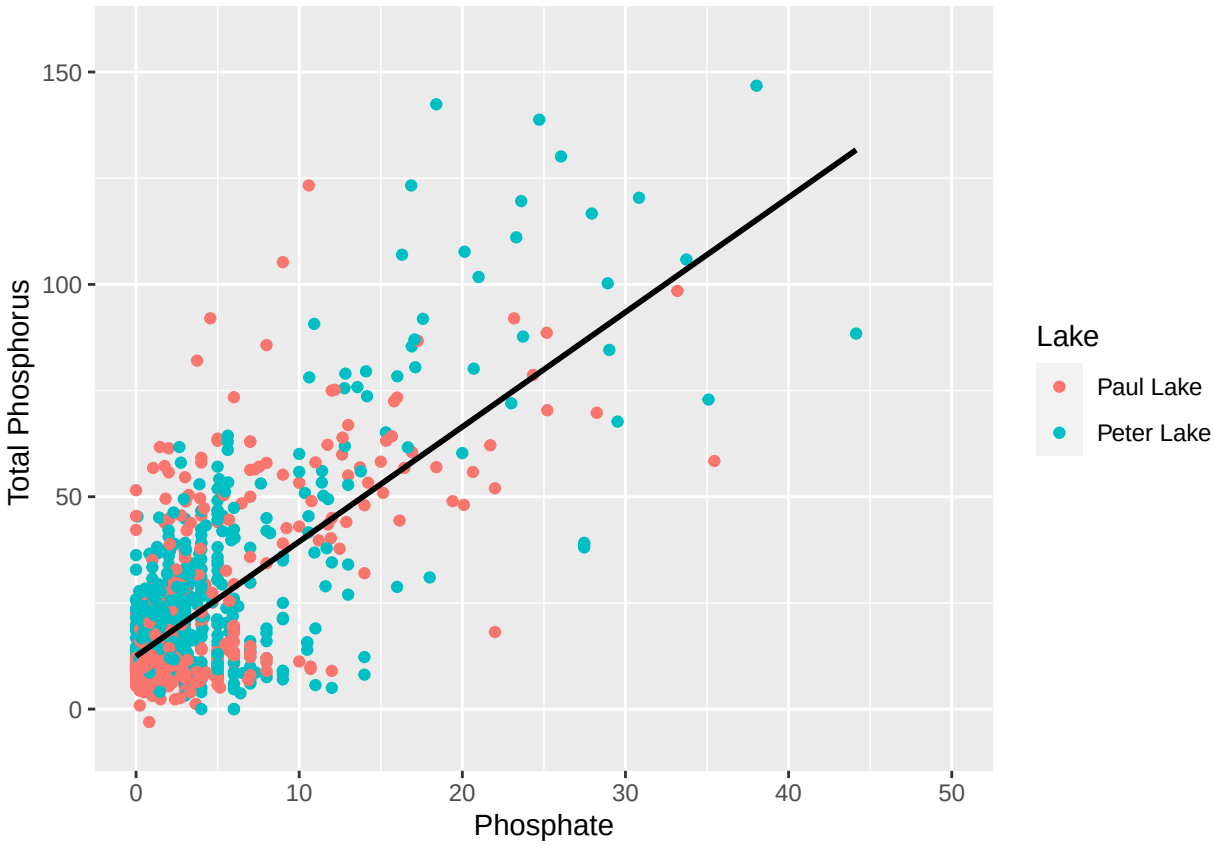
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Q4 <- ggplot(Lake, aes(x = po4,y = tp_ug, color = lakename))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE, color = "black")+
  labs(x = "Phosphate", y = "Total Phosphorus ",
       color = "Lake")+
  xlim(0,50)
print(Q4)
```

```
## `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 build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
str(Lake$month)
```

```
##  int [1:23008] 5 5 5 5 5 5 5 5 5 5 ...
```

```
Lake$month <- factor(Lake$month,levels = 1:12)
#temperature
boxplot1 <- ggplot(Lake, aes(x = factor(month, level = 1:12, labels = c(month.abb)), y = temperature_C,
  geom_boxplot()+
  labs(x = "Month", y = "Temperature", fill = "Lake")+
  xlab(NULL)

#TP
boxplot2 <- ggplot(Lake, aes(x = factor(month, level = 1:12, labels = c(month.abb)), y = tp_ug, fill = l
  geom_boxplot()+
  labs(x = "Month", y = "TP", fill = "Lake") +
  xlab(NULL)

#TN
boxplot3 <- ggplot(Lake, aes(x = factor(month, level = 1:12, labels = c(month.abb)), y = tn_ug, fill = l
  geom_boxplot()+
  labs(x = "Month", y = "TN", fill = "Lake")

#extract legend
legend_Q5 <- get_legend(boxplot1)
```

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

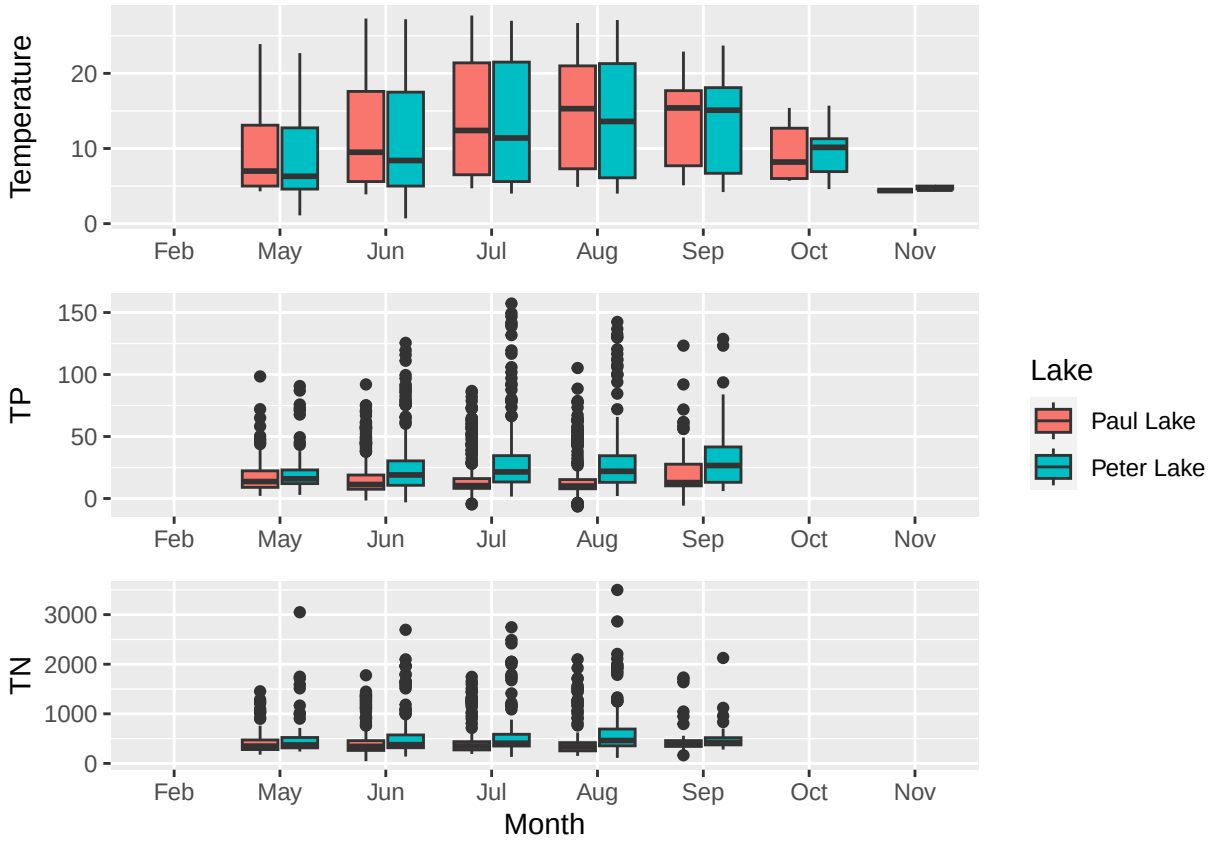
```
#no legend grid
plot_all_raw <- plot_grid(
  boxplot1 + theme(legend.position = "none"),
  boxplot2 + theme(legend.position = "none"),
  boxplot3 + theme(legend.position = "none"),
  align = 'v',
  ncol = 1
)
```

```
## 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()`).
```

```
# share legend
plot_grid(plot_all_raw, legend_Q5, rel_widths = c(4, 1))
```



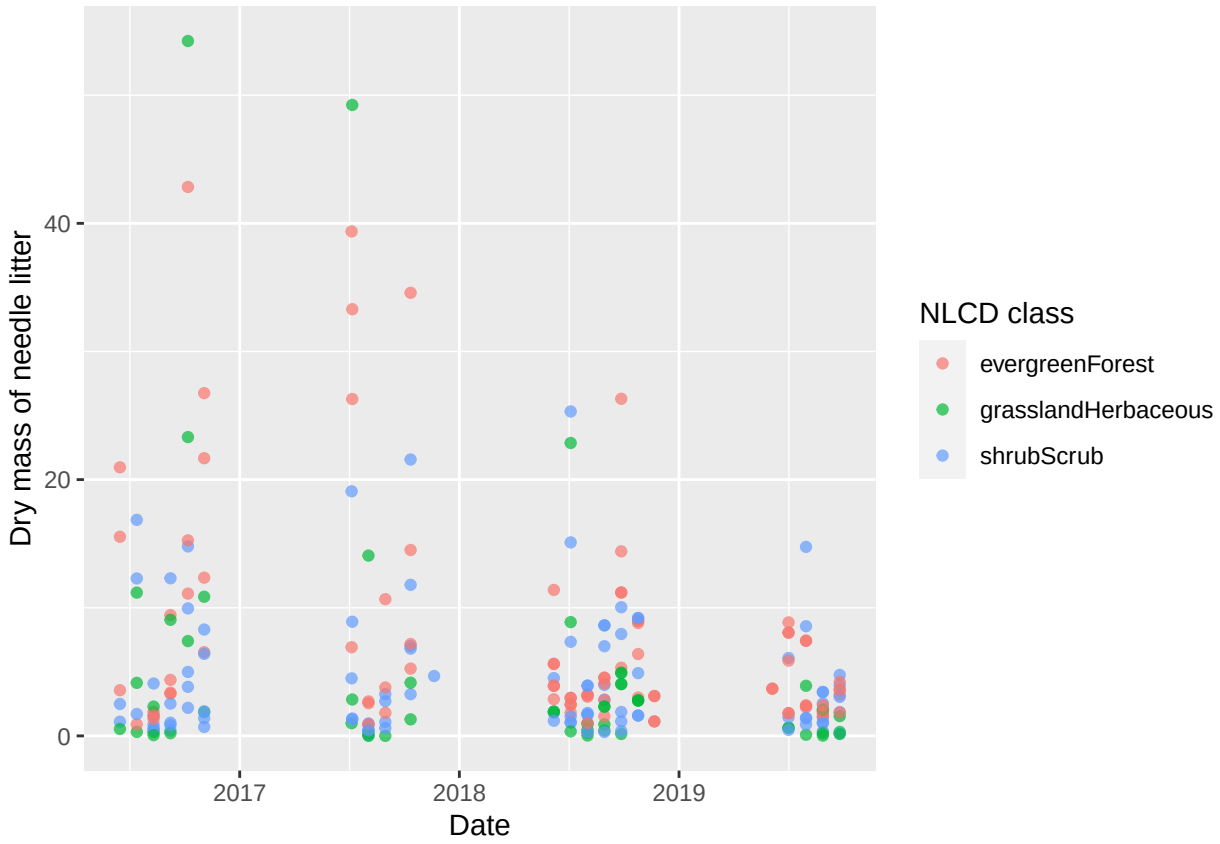
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: There is not much difference in temperature between the two lakes, and each month has a similar temperature range. It can be observed that in June, July, and September, the values of TP and TN in Lake Paul are higher than those in Lake Peter, and the maximum value of Lake PAUL is also higher.

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)

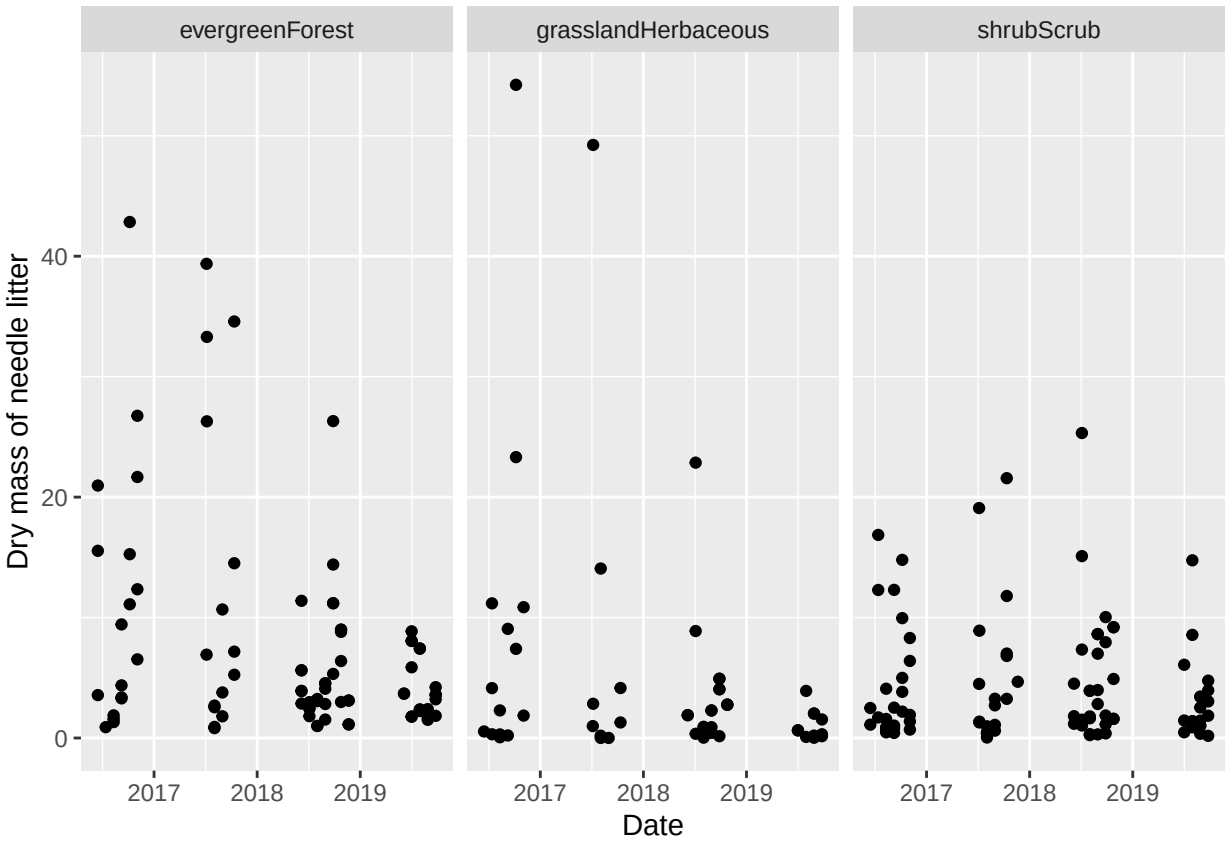
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Needles <- filter(Litter, functionalGroup == "Needles")
Plot_Q6 <- ggplot(Needles, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point(alpha = 0.7) +
  labs(x = "Date", y = "Dry mass of needle litter", color = "NLCD class")
print(Plot_Q6)
```




```
#7

Plot_Q7 <- ggplot(Needles, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 1)+
  labs(x = "Date", y = "Dry mass of needle litter")
print(Plot_Q7)
```



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

Answer: I think that Plot_Q7 is more effective because the different NLCD classes in Plot_Q7 are divided into different areas, which is easier to observe than the large number of overlapping points in Plot_Q6.