Assignment 5: Data Visualization

Jiahuan Li

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 packages, vertify home directory, & import data

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

```
library(here)
```

```
## here() starts at D:/Users/Lijh/Desktop/872 R & data analytics/ENV872
```

```
library(cowplot)
```

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

```
here()
```

```
## [1] "D:/Users/Lijh/Desktop/872 R & data analytics/ENV872"
```

```
lakes <- read.csv (file = here(
  "Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
litter <- read.csv (file = here(
  "Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2 check and transform the date format
class(lakes$sampledate)
```

```
## [1] "character"
```

```
class(litter$collectDate)
```

```
## [1] "character"
```

```
lakes$sampledate <- ymd(lakes$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 create and set the default theme
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'

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

```
my_theme <- theme_base() +
  theme(
    plot.background = element_rect(
      fill = "lightblue",
      colour = "lightblue",
      size = 0.5, linetype = "solid"),
    plot.title = element_text(
      color="black",
      size=14,
      hjust = 0.5),
    line = element_line(
      color='black',
      linewidth =1
    ),
    legend.background = element_rect(
      color='grey',
      fill = 'white'
    ),
    legend.title = element_text(
      color='black'
    ),
    axis.title = element_text(size = 10)
  )
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
```

```
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 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()`).

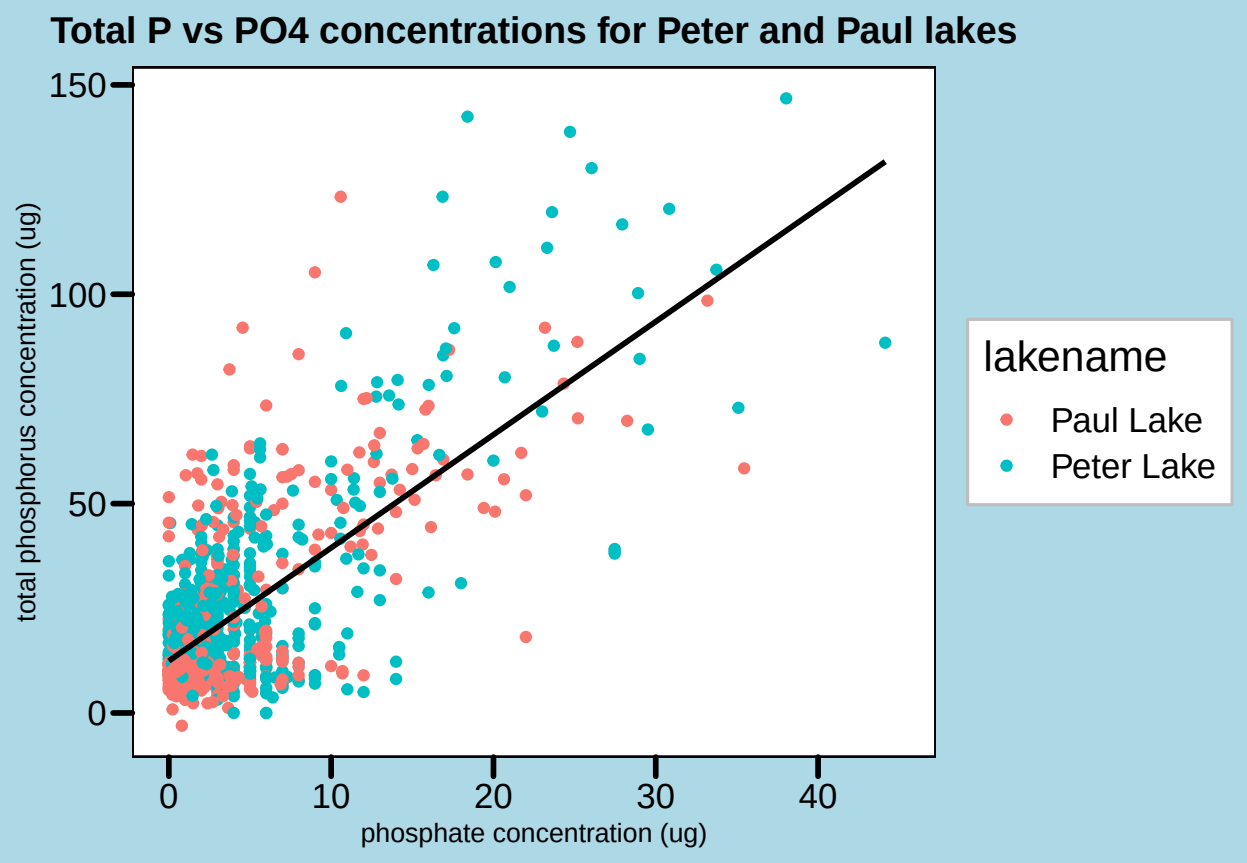
```r
#4
library(ggplot2)
plot1 <- lakes %>%
  drop_na(tp_ug) %>%
  drop_na(po4) %>%
  ggplot(aes(x=po4, y=tp_ug,color=lakename)) +
  geom_point() +
  geom_smooth(method='lm', se=FALSE, color = 'black') +
  xlim(0,45) +
  xlab("phosphate concentration (ug)") +
  ylab("total phosphorus concentration (ug)") +
  ggtitle("Total P vs PO4 concentrations for Peter and Paul lakes")

print(plot1)
```

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

## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 1 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.

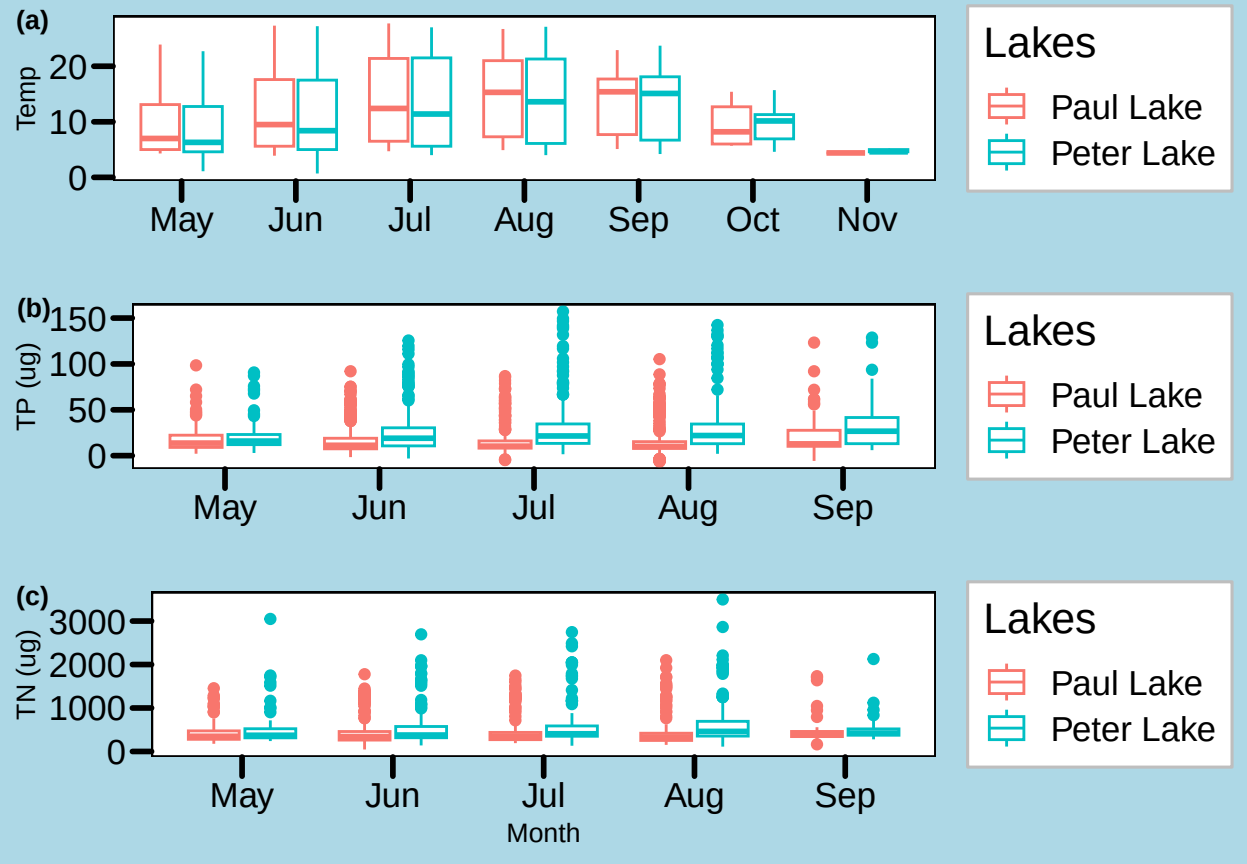
Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
plot2a <- lakes %>%
  drop_na(temperature_C) %>%
  ggplot(aes(y=temperature_C,x=as.factor(month),color=as.factor(lakename))) +
  geom_boxplot() +
  labs(x='Lakes',color='Lakes') +
  scale_x_discrete(labels= month.abb[unique(lakes$month)]) +
  xlab("") +
  ylab("Temp")

plot2b <- lakes %>%
  drop_na(tp_ug) %>%
  ggplot(aes(y=tp_ug,x=as.factor(month),color=as.factor(lakename))) +
  geom_boxplot() +
  labs(x='Lakes',color='Lakes') +
  scale_x_discrete(labels= month.abb[unique(lakes$month)]) +
  xlab("") +
  ylab("TP (ug)")

plot2c <- lakes %>%
  drop_na(tn_ug) %>%
  ggplot(aes(y=tn_ug,x=as.factor(month),color=as.factor(lakename))) +
  geom_boxplot() +
  labs(x='Lakes',color='Lakes') +
  scale_x_discrete(labels= month.abb[unique(lakes$month)]) +
  xlab("Month") +
  ylab("TN (ug)")

plot_grid(plot2a, plot2b, plot2c,
          nrow = 3, align = 'h',
          labels = c('(a)', '(b)','(c)'), label_size = 10,
          hjust = -0.5) +
  theme(legend.text = element_text(size = 10),
        legend.title = element_text(size = 12))
```



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

> Answer: The temperature observations for both lakes exhibit a generally concentrated distribution with few outliers. Both lakes show similar temporal trends, with temperature increasing from May to August or September, followed by a decrease until November. The values for the same month are approximately equal for both lakes with no significant differences.
>
> However, the TN and TP observations for both lakes contain a large proportion of outliers, indicating imperfections in the measurement process. Comparing the two lakes, Peter lake generally exhibits higher nutrient concentrations than Paul lake in all months, suggesting a more severe state of eutrophication. The nutrient levels for both lakes do not vary significantly with changes in the month.

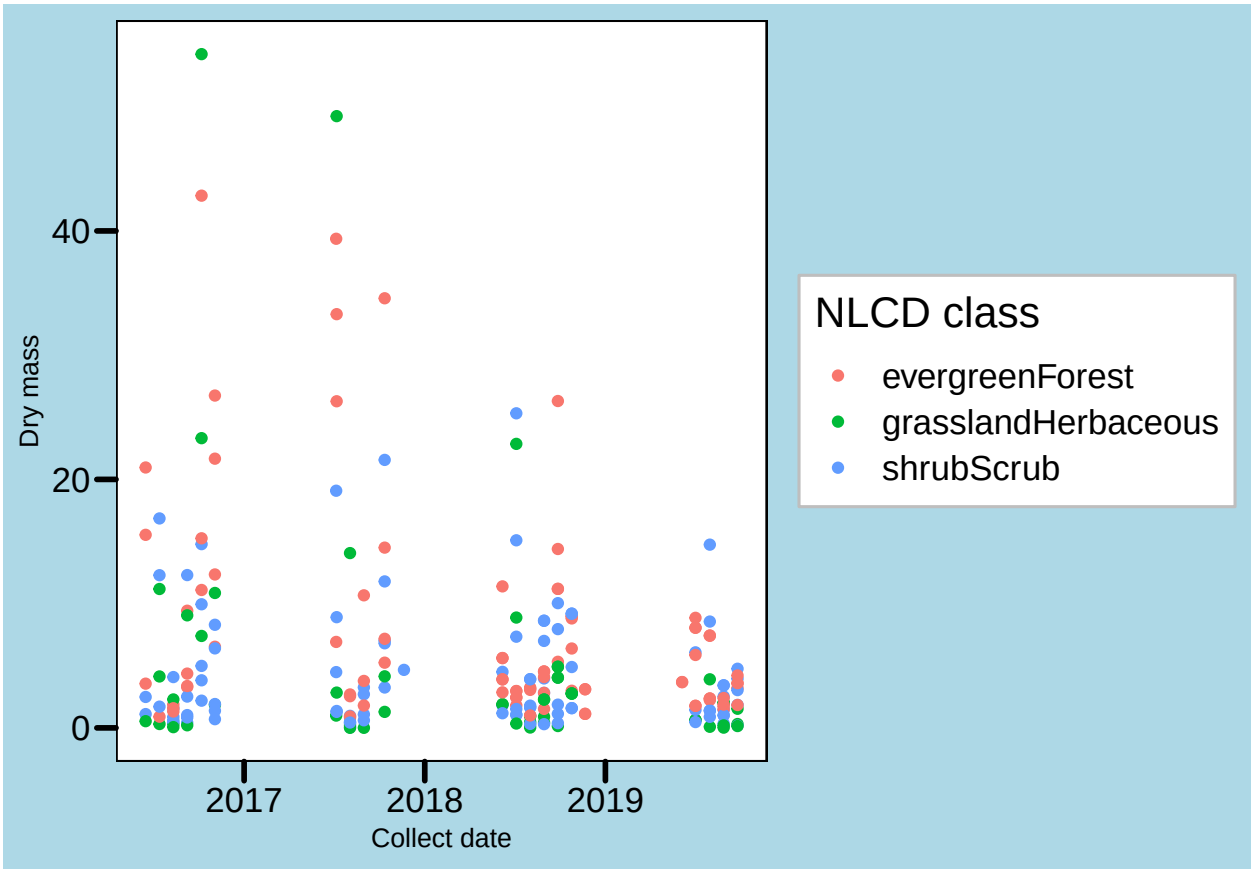
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
plot3 <- litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(y=dryMass,x=collectDate,color=nlcdClass)) +
  geom_point() +
  xlab("Collect date") +
  ylab("Dry mass") +
```

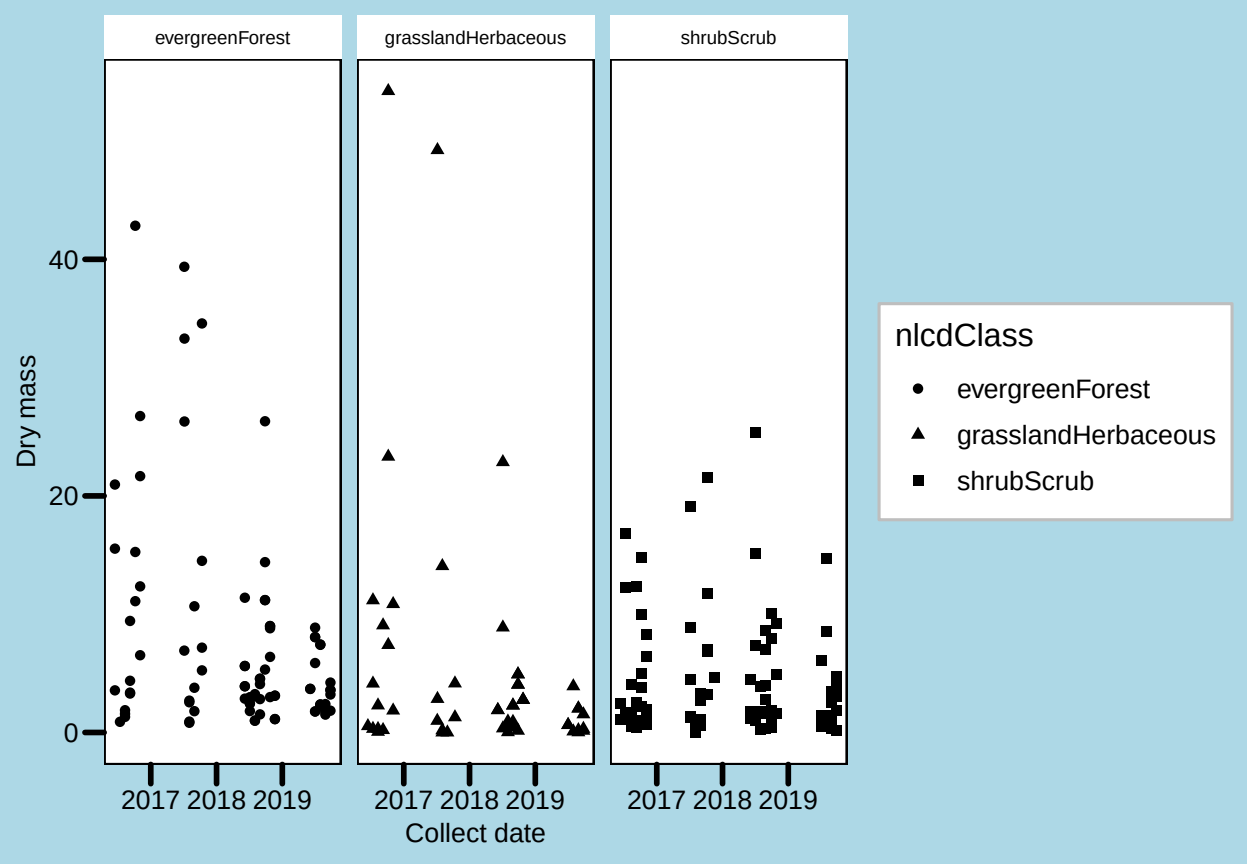
```
  guides(color = guide_legend(title = "NLCD class"))

print(plot3)
```




```
#7
plot4 <- litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(y=dryMass,x=collectDate,shape=nlcdClass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  xlab("Collect date") +
  ylab("Dry mass") +
  theme(axis.text = element_text(size = 10),
        strip.text = element_text(size = 7),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12))


print(plot4)
```



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

Answer: I think it depends on the types of information that are being concerned. To compare the dry mass value of different NLCD classes on the same date, the plot for question 6 is better suited. This method allows for better reflection of the differences among NLCD groups.

Conversely, showing the results in separated facets would be more effective for illustrating the temporal distribution and trend in each NLCD group. After that, we can intuitively know that if there are universal rules shown in different groups.