

Assignment 5: Data Visualization

Jinglin Zhang

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)

## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.0      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2     3.4.1      ✓ tibble     3.1.8
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts ————— tidyverse_co
```

```

nflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag() masks stats::lag()
## i Use the ]8;;http://conflicted.r-lib.org/conflicted-package]8;; to
force all conflicts to become errors

library(lubridate)
library(here)

## here() starts at C:/Users/wwwla/Documents/EDA-Spring2023

library(cowplot)

##
## 载入程辑包: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
## stamp

getwd()

## [1] "C:/Users/wwwla/Documents/EDA-Spring2023"

lake.chem.nut <- read.csv("~/EDA-Spring2023/Data/Processed_KEY/NTL-LTER
_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",stringsAsFactors = T
RUE)
litter.mass <- read.csv("~/EDA-Spring2023/Data/Processed/NEON_NIWO_Litt
er_mass_trap_Processed.csv",stringsAsFactors = TRUE)
#2
is.Date(lake.chem.nut$sampdate)

## [1] FALSE

is.Date(litter.mass$collectDate)

## [1] FALSE

lake.chem.nut$sampdate <- ymd(lake.chem.nut$sampdate)
litter.mass$collectDate <- ymd(litter.mass$collectDate)
is.Date(lake.chem.nut$sampdate)

## [1] TRUE

is.Date(litter.mass$collectDate)

## [1] TRUE

```

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(
  plot.background = element_rect(fill = "lightgray"),
  plot.title = element_text(size = 16, face = "bold"),
  axis.text = element_text(color = "black"),
  axis.line = element_line(colour = "black", size = 0.5),
  legend.position = "top",
  legend.text = element_text(size = 10)
)

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

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()`).

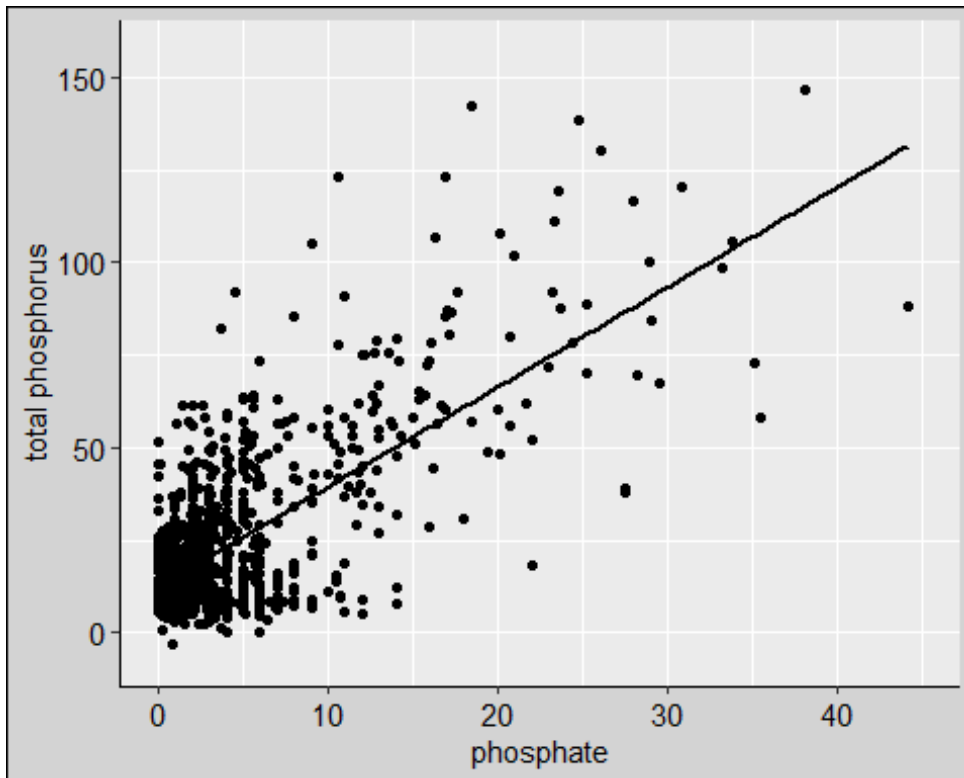
#4

```
ggplot(lake.chem.nut, aes(x = po4, y = tp_ug)) +
  geom_point() +
  xlim(0,45) +
  labs(x = "phosphate", y = "total phosphorus") +
  geom_smooth(method = lm, se = FALSE, colour = "black")

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

```
#5
lake.T <- ggplot(lake.chem.nut,
  aes(x = factor(month, levels = 1:12, labels = month.abb),
      y = temperature_C, color = lakename)) +
  scale_x_discrete(drop = FALSE) +
  geom_boxplot() +
  labs(x = "month", y = "Temperature")

lake.TP <- ggplot(lake.chem.nut,
  aes(x = factor(month, levels = 1:12, labels = month.abb),
      y = tp_ug, color = lakename)) +
  scale_x_discrete(drop = FALSE) +
  geom_boxplot() +
  labs(x = "month", y = "Total Phosphorus") +
  theme(legend.position = "none")
```

```

lake.TN <- ggplot(lake.chem.nut,
                 aes(x = factor(month, levels = 1:12, labels = month.ab
b),
                    y = tn_ug, color = lakename)) +
  scale_x_discrete(drop = FALSE) +
  geom_boxplot() +
  labs(x = "month", y = "Total Nitrogen") +
  theme(legend.position = "none")

legend <- get_legend(
  lake.T +
  guides(color = guide_legend(nrow = 2)) +
  theme(legend.position = "right")
)

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

plot1 <- plot_grid(lake.T + theme(legend.position = "none"),
                  lake.TP + theme(legend.position = "none"),
                  lake.TN + theme(legend.position = "none"),
                  nrow = 3)

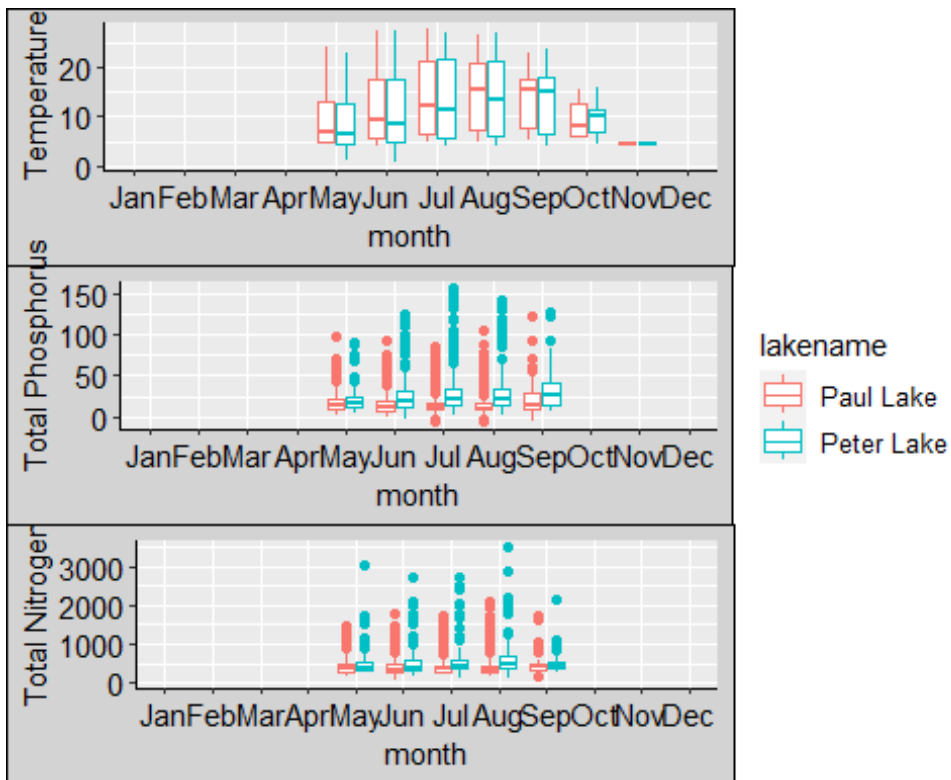
## Warning: Removed 3566 rows containing non-finite values (`stat_boxpl
ot()`).

## Warning: Removed 20729 rows containing non-finite values (`stat_boxp
lot()`).

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

plot_grid(plot1, legend, nrow = 1, rel_widths = c(3, 1))

```



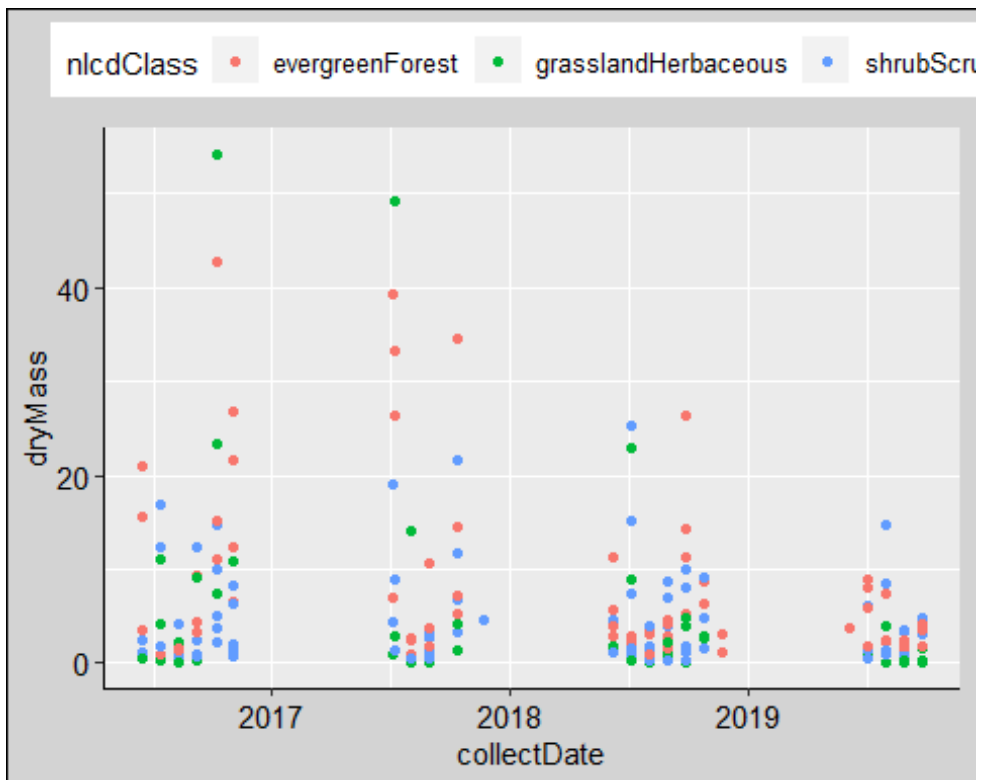
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: during the summer, the temperature of two lakes are similar, but TP and TN in Peter Lake are generally higher than them in Paul 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.

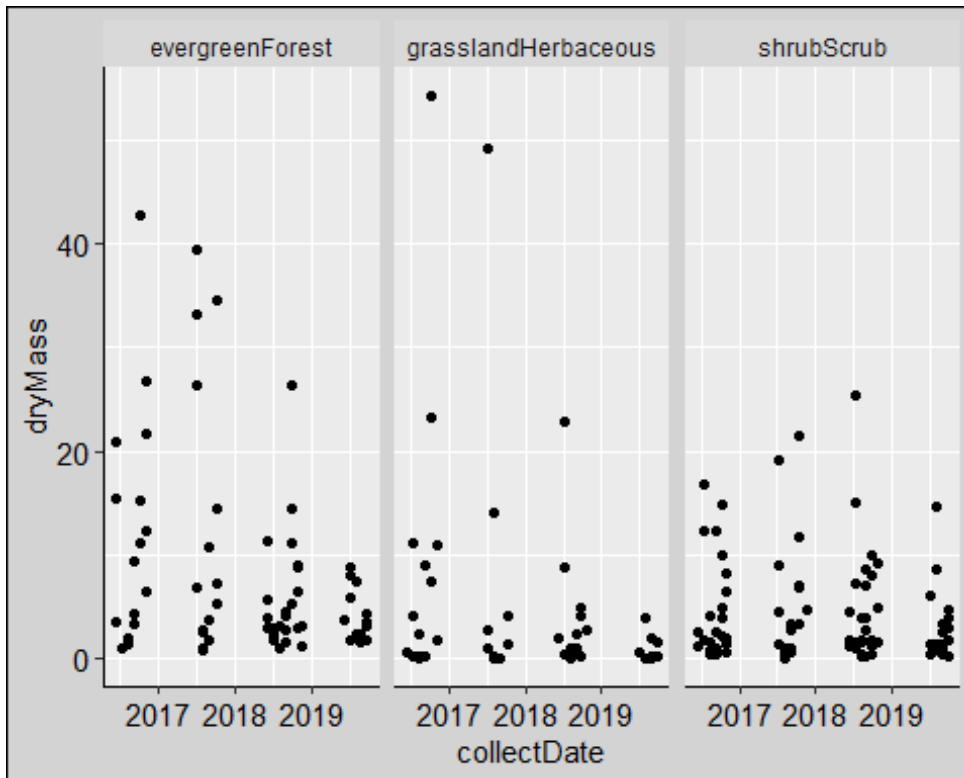
#6

```
litter.mass %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point()
```



#7

```
litter.mass %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass))
```



Question:

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

Answer: the second one, because it shows the trend of drymass more clearly than the 6th, there are too many points appear in a single graph that make me hard to catch informations.