Assignment 5: Data Visualization

Judy Hua Zhu

Fall 2024

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
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/872 Projects/EDE_Fall2024
```

```
library(cowplot)
```

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

```
library(ggplot2)
here()
```

```
## [1] "/home/guest/872 Projects/EDE_Fall2024"
```

```
processed_data = "./Data/Processed_KEY"

PeterPaul.chem.nutrients <- read.csv(
  here(processed_data,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

Litter <- read.csv(
  here(processed_data,"NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate)
Litter$collectDate <- as.Date(Litter$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
my_theme <- theme_bw() +
  theme(
    line = element_line(color='purple',linewidth = 2),
    axis.text = element_text(color = "black"),
    legend.position = "right",
    title.position = "top",
```

```
    plot.title.size = 2,
    points.size = 0.5,
    panel.grid.major =  element_line(linewidth=0.5),
    panel.grid.minor =  element_line(linewidth=0.5)
    )
```

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 line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

Q4 <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug,
                                           color = lakename)) +
  geom_point() +                                # Scatter plot
  geom_smooth(method = "lm", se = FALSE) +      # Line of best fit without confidence interval shading
  xlim(0, 50) +                                 # Adjust x-axis limits to hide extreme values
  ylim(0, 150) +                                # Adjust y-axis limits to hide extreme values
  labs(title = "Total Phosphorus vs. Phosphate by Lake",
       x = "Phosphate (po4)",
       y = "Total Phosphorus (tp_ug)",
       color = "Lake name") +
  my_theme

Q4
```

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

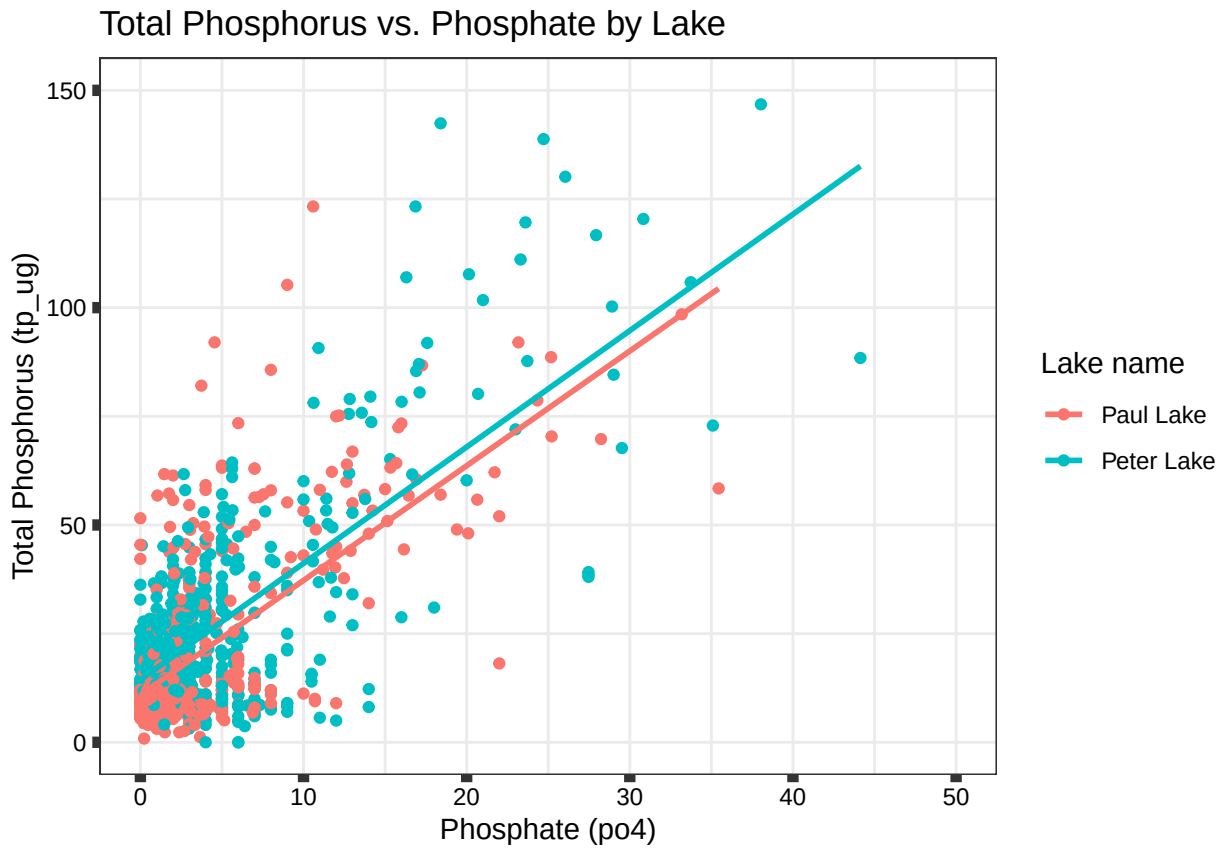
```
## Warning: Removed 21948 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning in plot_theme(plot): The `title.position` theme element is not defined
## in the element hierarchy.
```

```
## Warning in plot_theme(plot): The `plot.title.size` theme element is not defined
## in the element hierarchy.
```

```
## Warning in plot_theme(plot): The `points.size` theme element is not defined in
## the element hierarchy.
```

```
## Warning: Removed 21948 rows containing missing values or values outside the scale range
## (`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.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

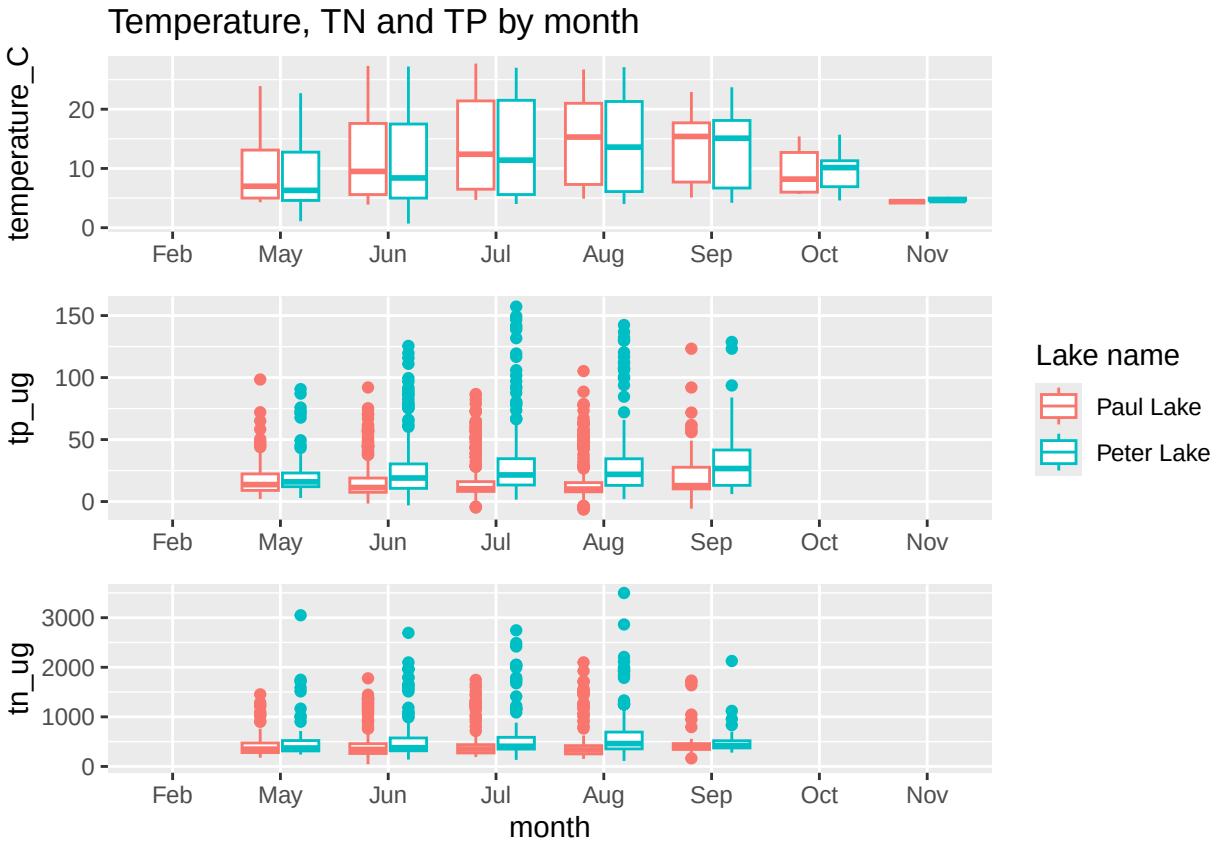
```
#5
PeterPaul.chem.nutrients$month <- factor(PeterPaul.chem.nutrients$month,
       levels=1:12,
       labels = month.abb)

Q5_temp <- ggplot(PeterPaul.chem.nutrients, aes(color=lakename, x= month))+
  geom_boxplot (aes(y = temperature_C))+
  theme(legend.position = "none",
        axis.title.x = element_blank())+
  labs(title = "Temperature, TN and TP by month")

Q5_tp <- ggplot(PeterPaul.chem.nutrients, aes(color=lakename, x= month))+
  geom_boxplot(aes(y = tp_ug)) +
  theme(axis.title.x = element_blank())+
  labs(color = "Lake name")
```

```
Q5_tn <- ggplot(PeterPaul.chem.nutrients, aes(color=lakename, x= month))+
  geom_boxplot(aes(y = tn_ug))+
  theme(legend.position = "none")

plot_grid (Q5_temp, Q5_tp, Q5_tn, nrow = 3, align = "v")
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



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

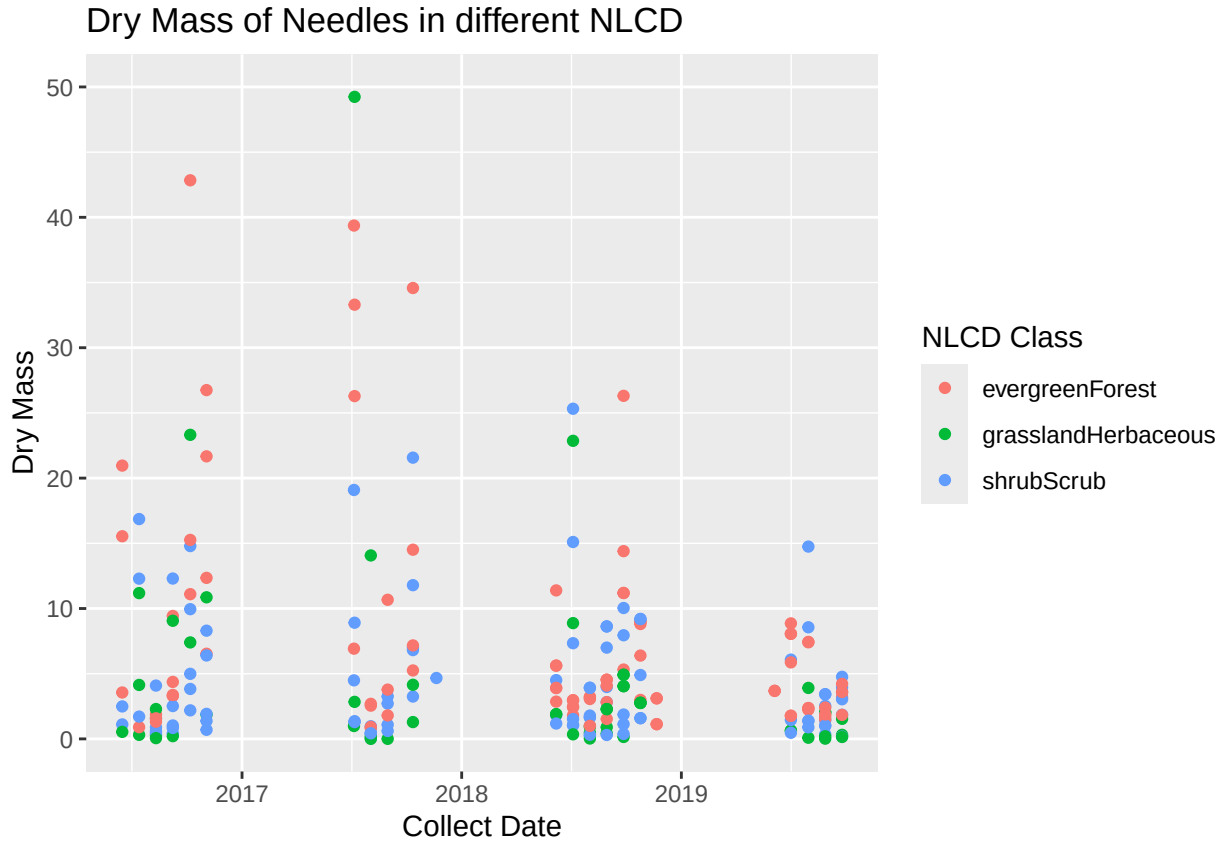
> Answer: Peter lake has lower summer temperature but higher TP and TN values than Paul lake. Also, TN and TP is a bit higher in summer than spring and fall, which are the only data available in this dataset.

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 Date not shown in x-axis;
Q6 <- Litter %>%
  filter (functionalGroup=="Needles") %>%
  ggplot (aes(x = collectDate, y = dryMass, color= nlcdClass)) +
  geom_point()+
  ylim(0, 50) +
  scale_x_date(date_labels = "%Y")+
  ylab("Dry Mass")+
  xlab("Collect Date") +
  labs(title = "Dry Mass of Needles in different NLCD",color = "NLCD Class")

Q6
```

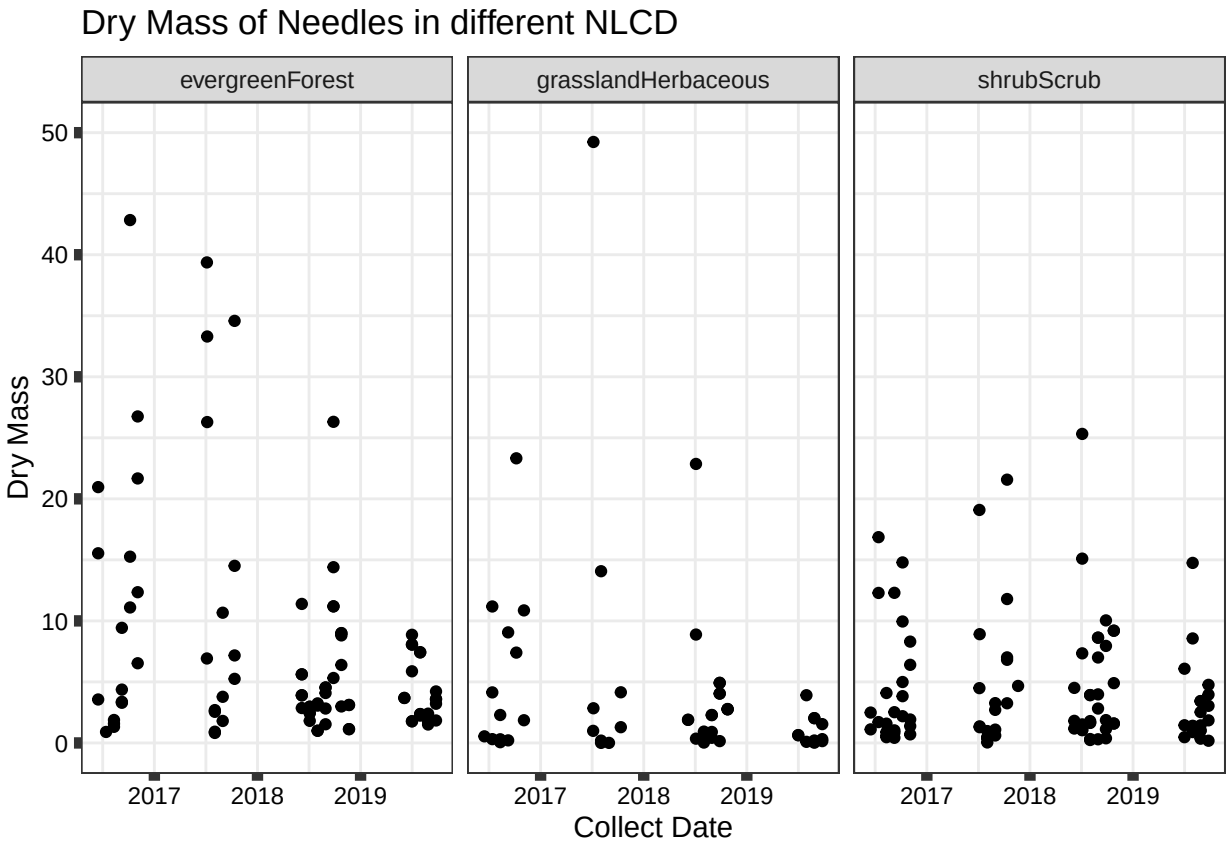
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
#7
Q7 <- Litter %>%
  filter (functionalGroup=="Needles") %>%
  ggplot (aes(x = collectDate, y = dryMass)) +
  geom_point()+
  ylim(0, 50) +
```

```
  scale_x_date(date_labels = "%Y")+
  ylab("Dry Mass")+
  xlab("Collect Date") +
  labs(title = "Dry Mass of Needles in different NLCD",color = "NLCD Class")+
  facet_wrap(vars(nlcdClass),ncol = 3) +
  my_theme

Q7
```

```
## Warning in plot_theme(plot): The ‘title.position‘ theme element is not defined
## in the element hierarchy.

## Warning in plot_theme(plot): The ‘plot.title.size‘ theme element is not defined
## in the element hierarchy.

## Warning in plot_theme(plot): The ‘points.size‘ theme element is not defined in
## the element hierarchy.

## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_point()‘).
```



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

Answer: I think plot 7 is more effective, as it shows the comparison of data in different NLCD more clearly by separating the data. Plot 6 is hard to compare the NLCDs since the data is

cluttered together and is hard for color-weak or even normal people. However, if we care more on the trend of dry mass change with time instead of the NLCD class, plot 6 is a good represent.