Assignment 5: Data Visualization

Natalie Holsclaw

Spring 2025

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 Load required packages
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 1.9.4     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(cowplot)
```

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

```
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2025
```

```
library(ggplot2)

# check working directory
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
# Load in data
PeterPaul_processed <- read.csv(
  here("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

Niwot_processed <- read.csv(
  here("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2 Check date formats
class(PeterPaul_processed$sampledate)
```

```
## [1] "factor"
```

```
class(Niwot_processed$collectDate)
```

```
## [1] "factor"
```

```
# Change to date format
PeterPaul_processed$sampledate <- ymd(PeterPaul_processed$sampledate)
Niwot_processed$collectDate <- ymd(Niwot_processed$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 Build default theme
mytheme <- theme(panel.background = element_rect(fill = "cornsilk"),
        legend.title = element_text(color = "darkseagreen4"),
        axis.text.x = element_text(angle = 45),
        panel.grid.major = element_line(color = "white"))
```

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.

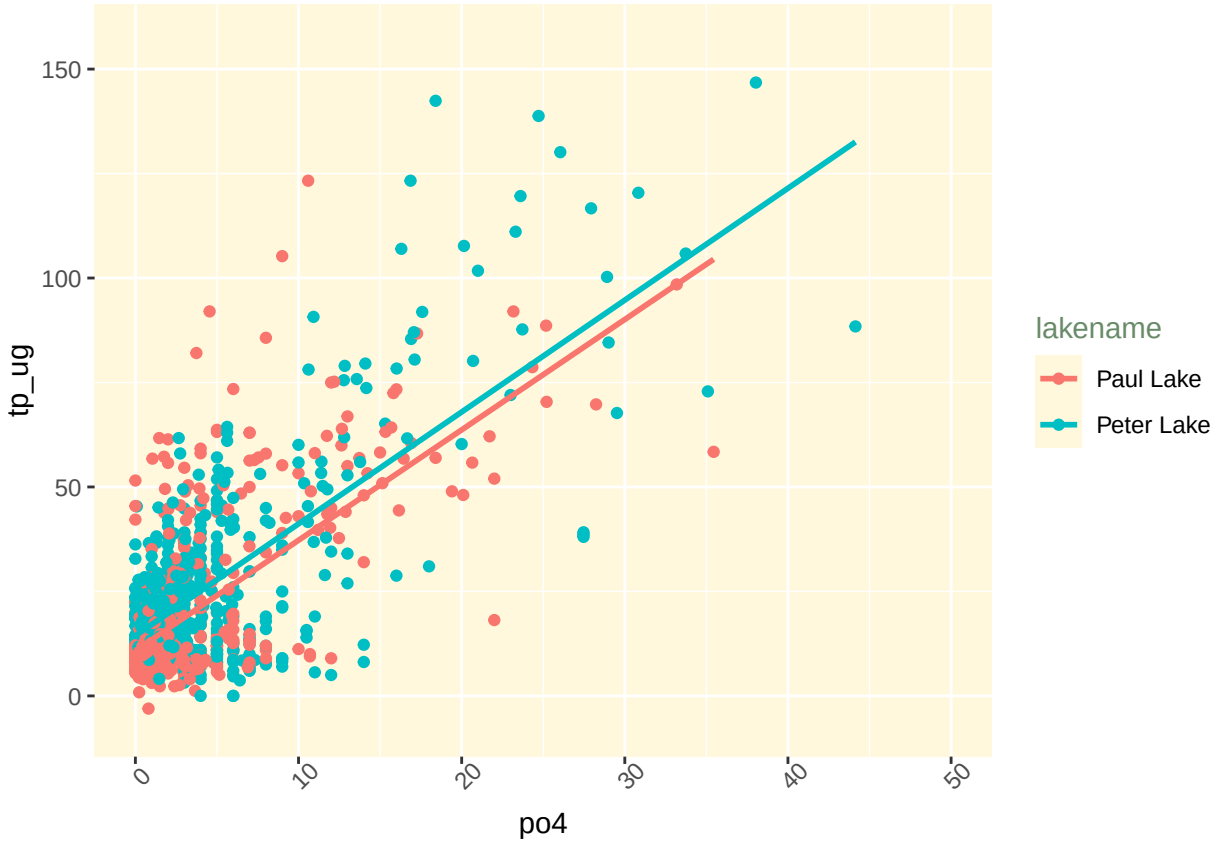
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4 Phosphorus vs phosphate
PeterPaul_processed %>%
  ggplot(aes(x = po4, y = tp_ug, color = lakename))+
  geom_point()+
  xlim(x = c(0, 50))+
  geom_smooth(method = lm, se = FALSE)+
  mytheme
```

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

```
## Warning: Removed 21947 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21947 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 Temperature boxplot
temp_plot <- PeterPaul_processed %>%
  ggplot(aes(x = factor(month), y = temperature_C, color = lakename))+
  geom_boxplot()+
  scale_color_manual(values = c("darkseagreen4", "darkslategray3"))+
  theme(axis.title.x = element_blank(),
        legend.position = 'none')

# TP boxplot
tp_plot <- PeterPaul_processed %>%
  ggplot(aes(x = factor(month), y = tp_ug, color = lakename))+
  geom_boxplot()+
  scale_color_manual(values = c("darkseagreen4", "darkslategray3"))+
  theme(axis.title.x = element_blank())

# TN boxplot
```

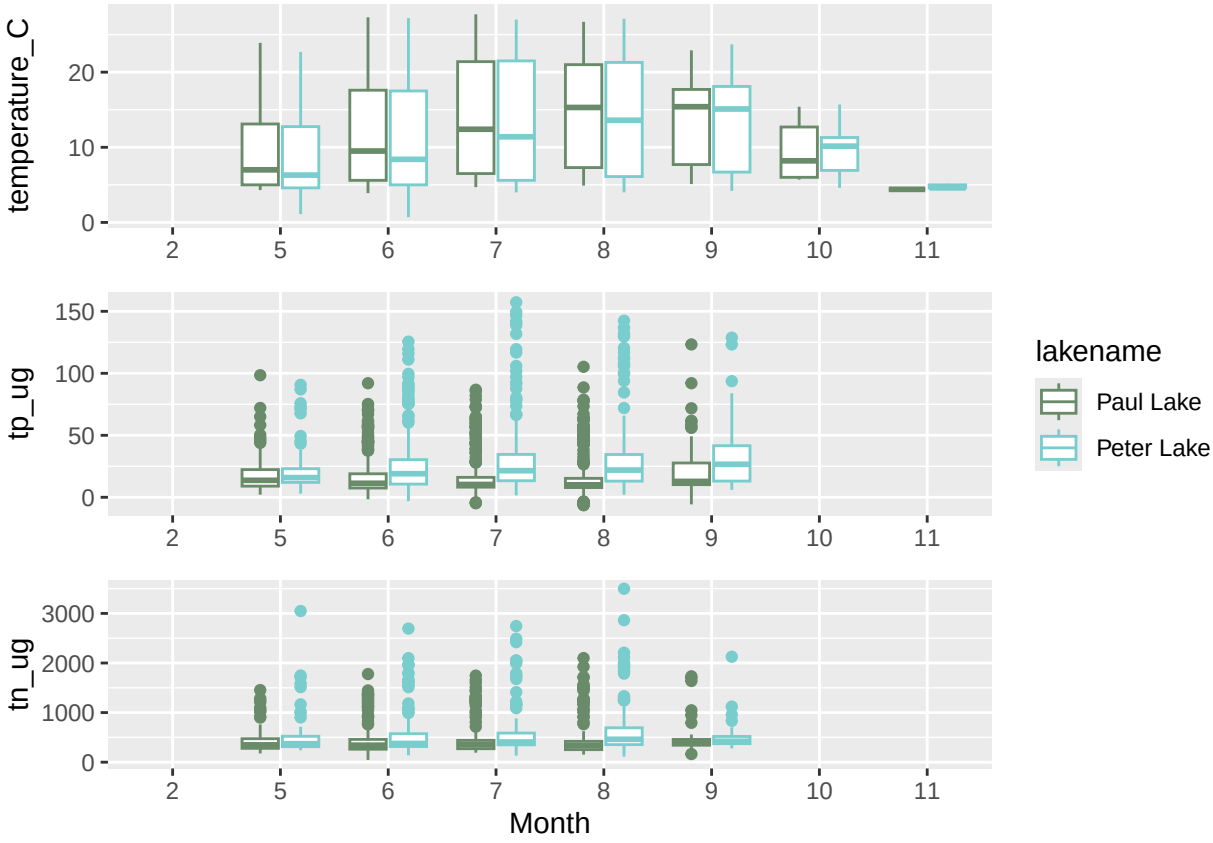
```
tn_plot <- PeterPaul_processed %>%
  ggplot(aes(x = factor(month), y = tn_ug, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  scale_color_manual(values = c("darkseagreen4", "darkslategray3"))+
  theme(legend.position = 'none')

# Combine plots
plot_grid(temp_plot, tp_plot, tn_plot, 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?

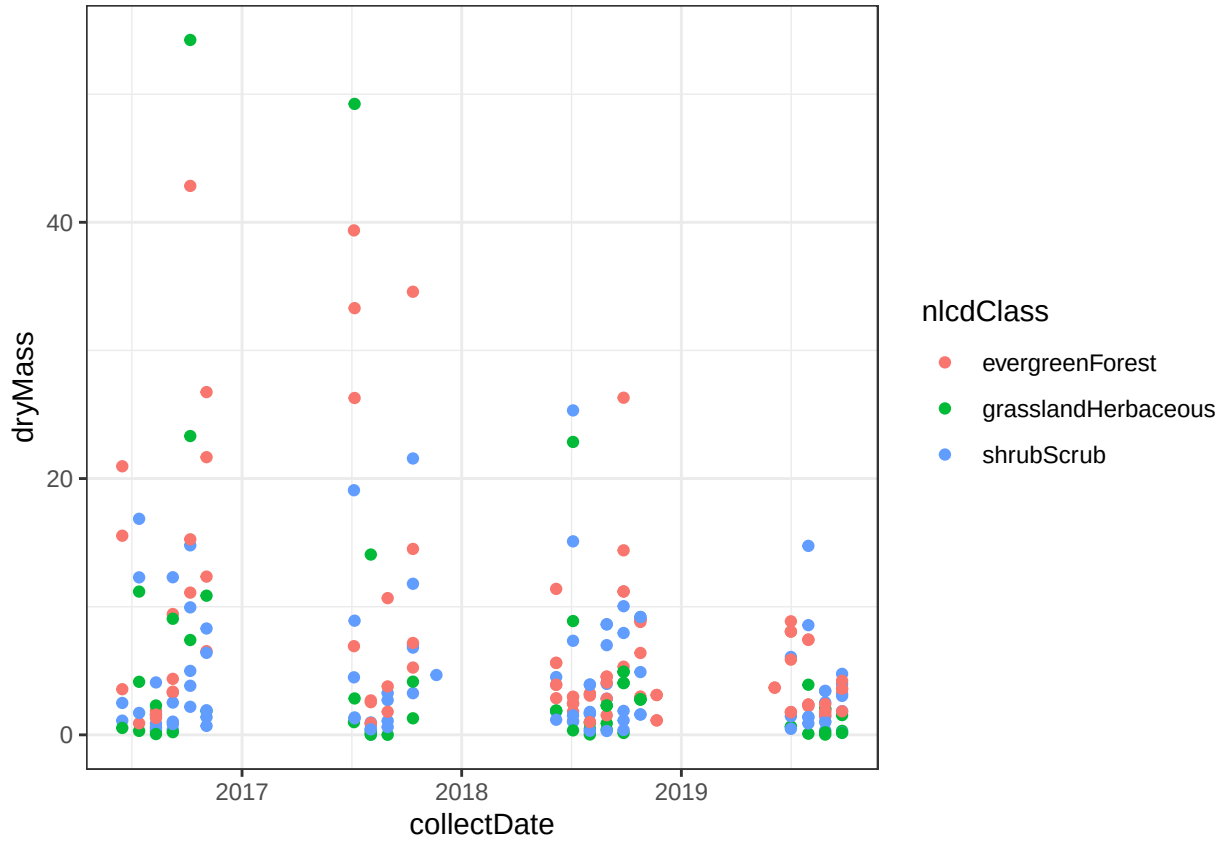
> Answer: There isn't a lot of variation in nitrogren and phosphorus over the months but temperatures vary widely especially in the winter months. Nitrogen and phosphorus may have slighly higher concentrations in Peter Lake and lake temperatures are fairly similar.

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.

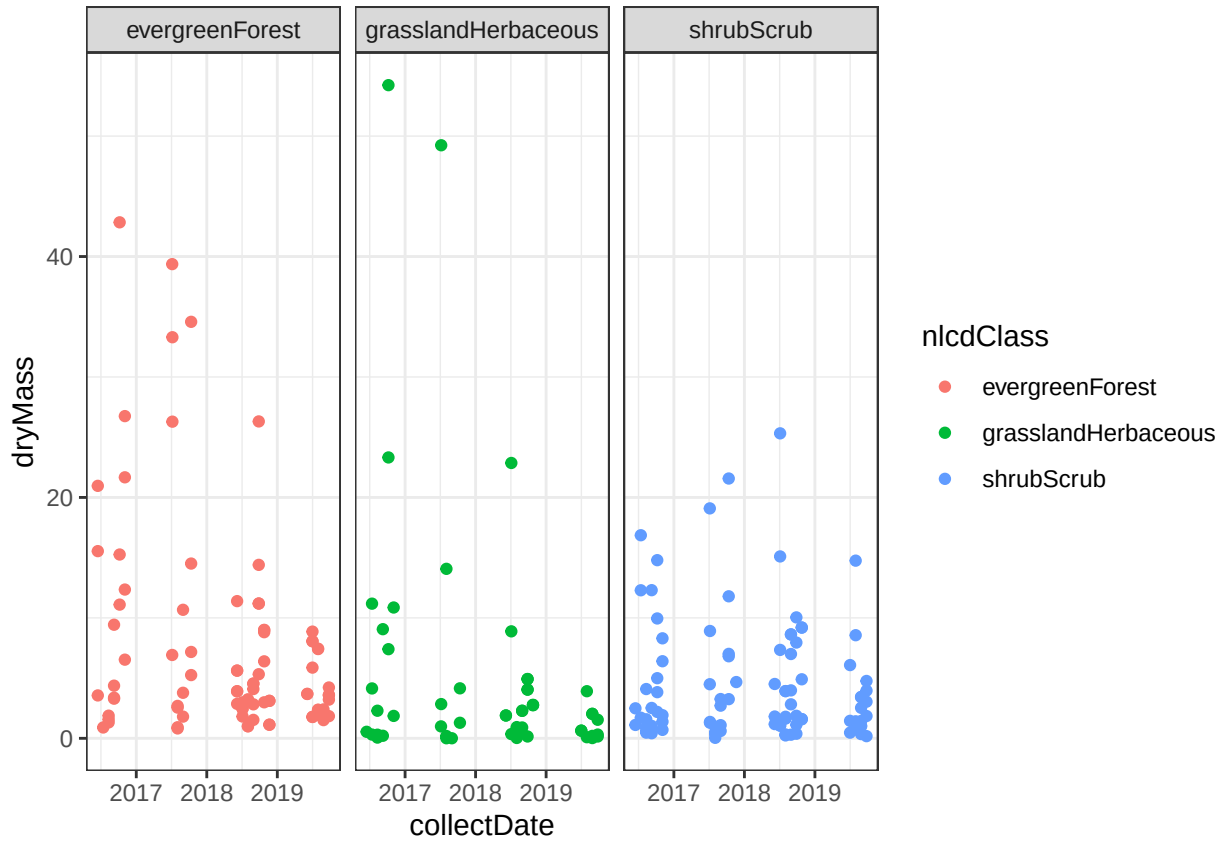
```
#6 Dry mass of needles
needles_plot <- Niwot_processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point()+
  theme_bw()

needles_plot
```




```
#7 Dry mass of needles with facet
needles_plot2 <- Niwot_processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point()+
  theme_bw()+
  facet_wrap(vars(nlcdClass))

needles_plot2
```



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

Answer: I think plot 7 is more effective when trying to view differences in NLCD classes. It is hard to observe patterns in the data when they are plotted together.