Assignment 5: Data Visualization

Jonathan Gilman

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.

```r
#1
#load 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 1.9.3     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/EDE_Fall2024
```

```
library(cowplot)
```

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

```
#check current working directory
here()
```

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

```
#upload datasets
nutrients <- read.csv(
  file = here('Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
  stringsAsFactors = T)

litter <- read.csv(
  file = here('Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv'),
  stringsAsFactors = T)

#2
# check class of date column
class(nutrients$sampledate)
```

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

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

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

```
# change the date columns to be date objects.
nutrients$sampledate <- as.Date(nutrients$sampledate, format = "%m/%d/%Y")
litter$collectDate <- as.Date(litter$collectDate, format = "%Y-%m-%d")
```

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 a custom theme
jpg_theme <- theme(
  # plot background
  panel.background = element_rect(fill = "skyblue4", color = "black", linewidth = 1),
  plot.background = element_rect(fill = "white", color = "black", linewidth = 1),
  # plot title
  plot.title = element_text(face = "bold", size = 16, hjust = 0.5, color = "olivedrab"))

# set as the default
theme_set(jpg_theme)
```

Create graphs

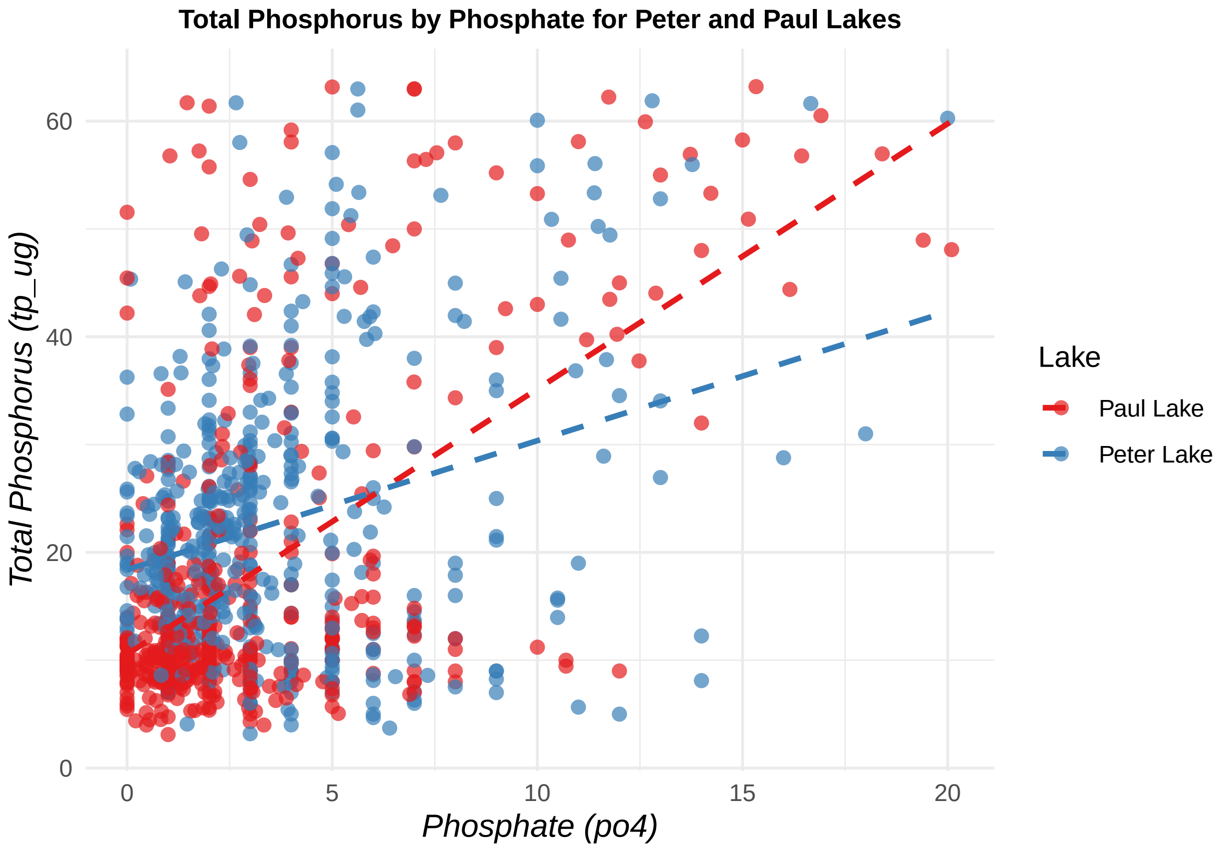
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add 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
# filter for Peter and Paul lakes
nutrients_PP <- nutrients %>%
  filter(lakename %in% c("Peter Lake", "Paul Lake"))

# make ggplot
ggplot(nutrients_PP, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point(size = 2, alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed") +
  labs( title = "Total Phosphorus by Phosphate for Peter and Paul Lakes",
    x = "Phosphate (po4)",
    y = "Total Phosphorus (tp_ug)",
    color = "Lake") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 10, hjust = 0.5),
    axis.title = element_text(size = 12, face = "italic"),
    legend.position = "right") +
  scale_color_brewer(palette = "Set1")+
  # remove extremes
  xlim(quantile(nutrients_PP$po4, probs = c(0.05, 0.95), na.rm = TRUE)) +
  ylim(quantile(nutrients_PP$tp_ug, probs = c(0.05, 0.95), na.rm = TRUE))
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 22020 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
## Warning: Removed 22020 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



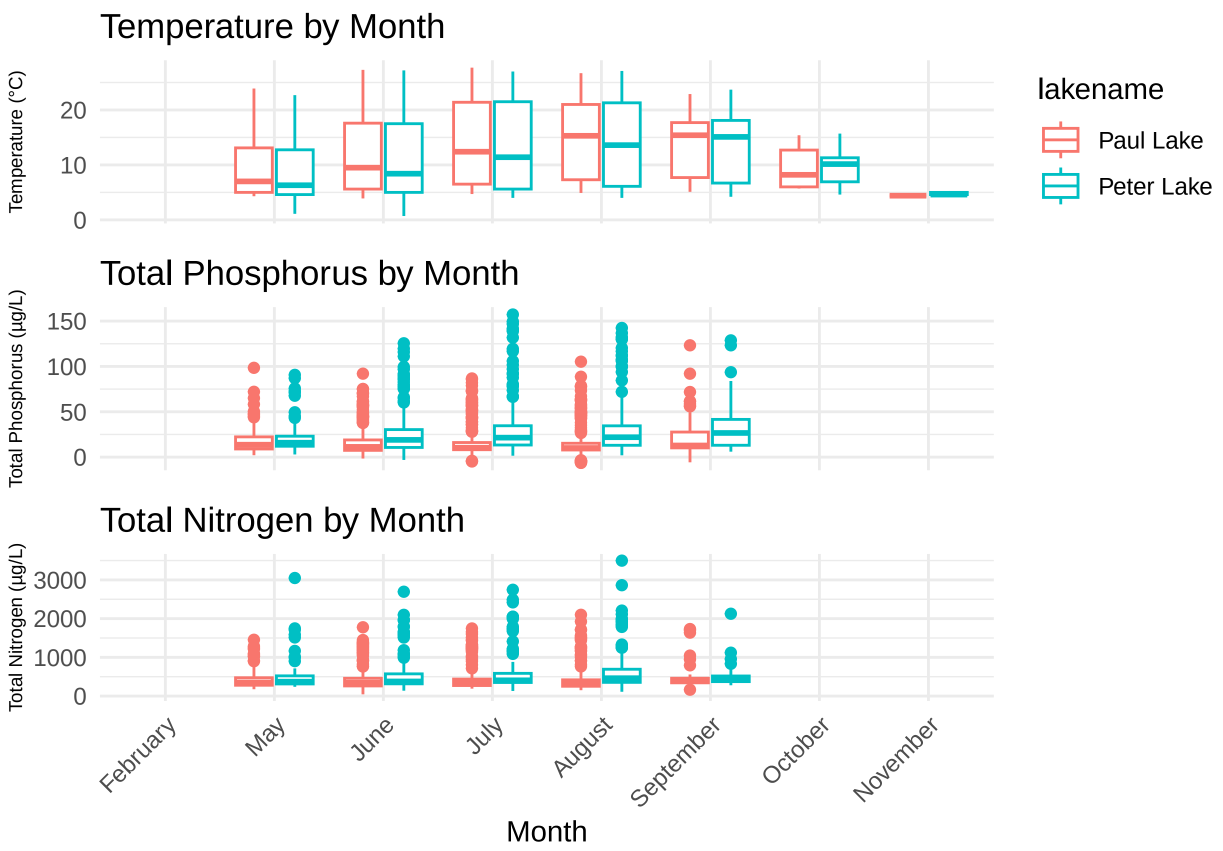
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
# make months as factors
nutrients$month <- factor(nutrients$month, levels = 1:12, labels = month.name)


# temp boxplot
temp_plot <- ggplot(nutrients, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  labs(title = "Temperature by Month", y = "Temperature (°C)") +
  theme_minimal() +
  theme( legend.position = "right",
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_text(size = 7))

# TP boxplot
tp_plot <- ggplot(nutrients, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(title = "Total Phosphorus by Month", y = "Total Phosphorus (µg/L)") +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_text(size = 7))

# TN boxplot
tn_plot <- ggplot(nutrients, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(title = "Total Nitrogen by Month", x = "Month", y = "Total Nitrogen (µg/L)") +
  theme_minimal() +
  theme(legend.position = "none",
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.y = element_text(size = 7))


# combine the plots using cowplot
combined_plot <- plot_grid(
  temp_plot, tp_plot, tn_plot,
  align = "v", ncol = 1, rel_heights = c(1, 1, 1.5))
```

```
## 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()‘).
```

```
print(combined_plot)
```

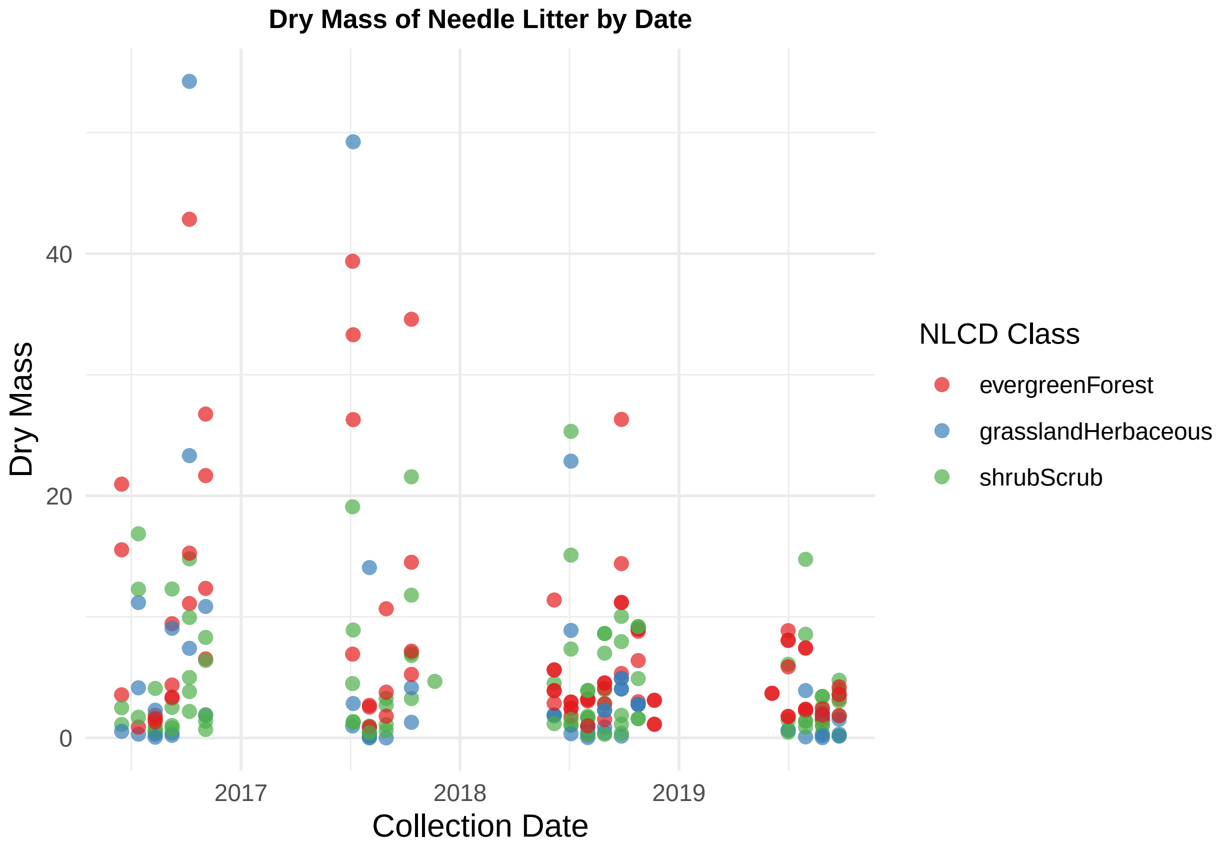


Question: What do you observe about the variables of interest over seasons and between lakes?

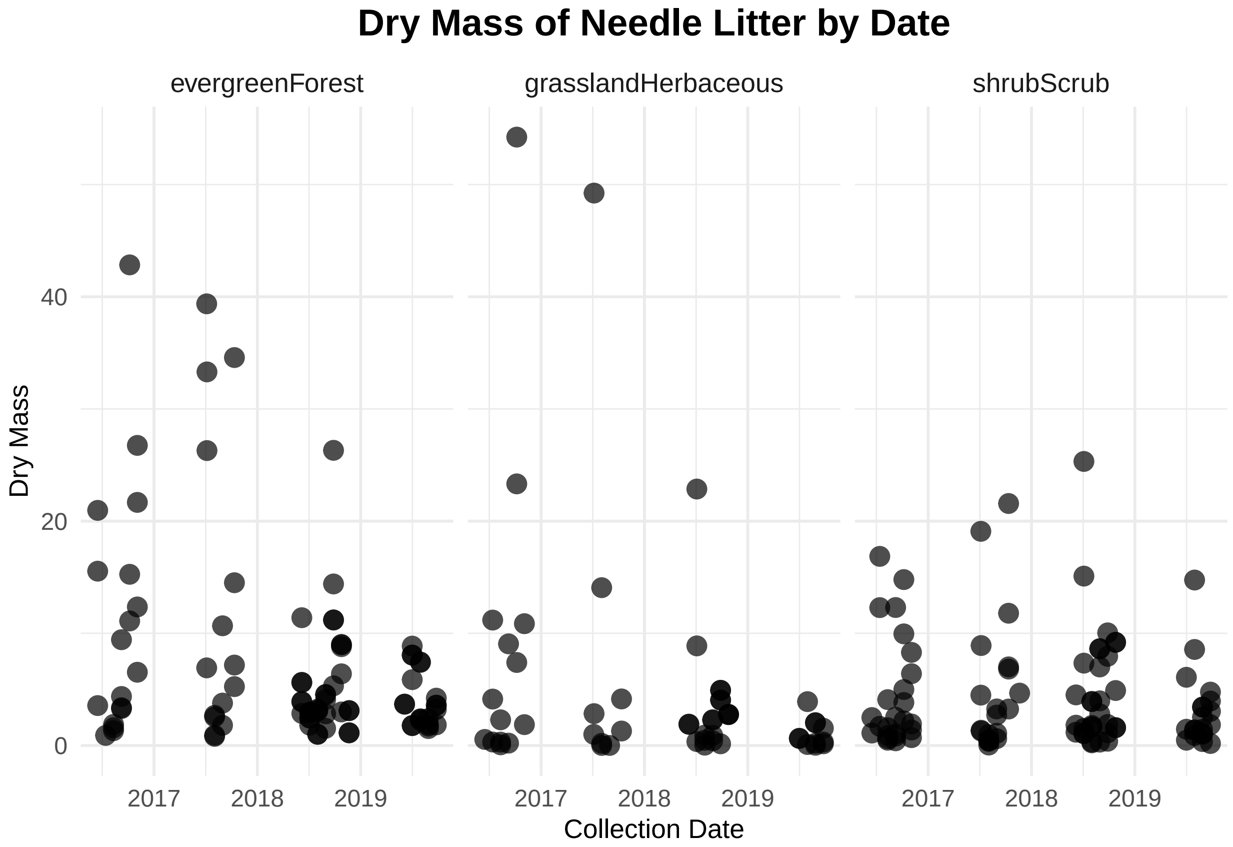
> Answer: I observe that the nutrients seem to increase in the warmer months – or at least the highest extremes occur in the warmer months. Also, it appears that Peter Lake tends to have higher nutrient concentrations.

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.

```r
#6
# filter for "Needles" functional group
needles_data <- litter %>%
  filter(functionalGroup == "Needles")

# Create the plot
ggplot(needles_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point(size = 2, alpha = 0.7) +
  labs( title = "Dry Mass of Needle Litter by Date",
    x = "Collection Date",
    y = "Dry Mass",
    color = "NLCD Class"
  ) +
  theme_minimal() +
  theme( plot.title = element_text(face = "bold", size = 10, hjust = 0.5),
    axis.title = element_text(size = 12)
  ) +
  scale_color_brewer(palette = "Set1")
```



```
#7
# filter for "Needles" functional group
needles_data <- litter %>%
  filter(functionalGroup == "Needles")

# make plot with facets
ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point(size = 3, alpha = 0.7) +
  labs( title = "Dry Mass of Needle Litter by Date",
    x = "Collection Date",
    y = "Dry Mass" ) +
  facet_wrap(~nlcdClass) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
    axis.title = element_text(size = 10),
    strip.text = element_text(size = 10) )
```



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

> Answer: I think the first plot #6 is more effective because it is more consolidated. Personally, I like seeing the three years once with NLCD class separated by color is very easy to view and understand. Plot #7 took me a second to get oriented, and it is less easy to compare classes by year.