Assignment 5: Data Visualization

Sydney Williams

Spring 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
#loading basic libraries
library(tidyverse);library(lubridate);library(here);library(cowplot)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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
## here() starts at /home/guest/EDA_Spring2024
##
##
## Attaching package: 'cowplot'
##
```

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

```
# verifying my home directory
here()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

```
#Assigning a variable to the processed data folder location
processed_data = "Data/Processed_KEY"

# Reading in the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul L
PeterPaul.chem.nutrients <- read.csv(
  here(processed_data, "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

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

#2 Making sure dates are in right format

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

class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(NIWO_Litter$collectDate)
```

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

```
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
NIWO_Litter$collectDate <- ymd(NIWO_Litter$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 a custom theme
my_theme <- theme_minimal() +
  theme(
    line = element_line(
      color='red',
      linewidth =2
    ),
```

```
    legend.background = element_rect(
      color='grey',
      fill = 'orange'
    ),
    legend.title = element_text(
      color='blue'
    )

  )

# Setting the default theme
theme_set(my_theme)
```

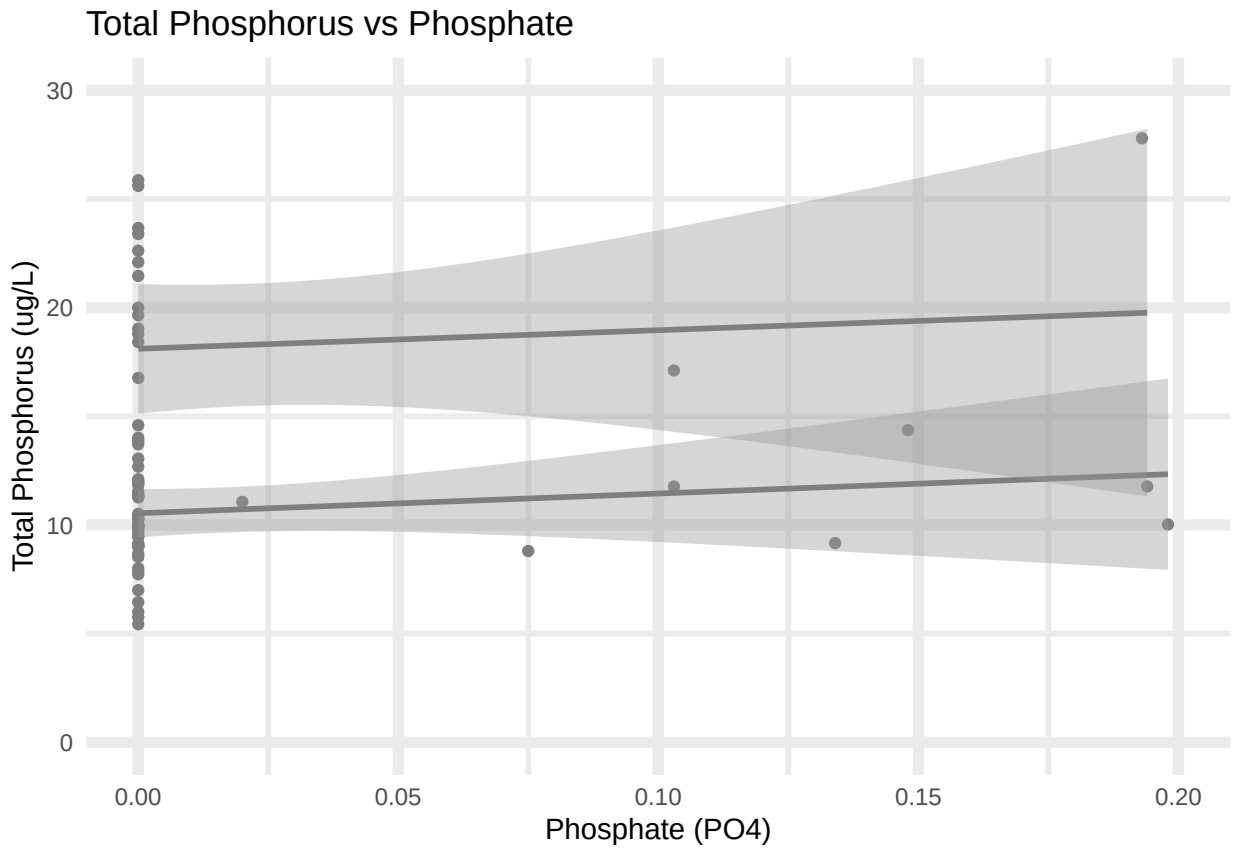
Create graphs

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 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
# Plot total phosphorus (tp_ug) by phosphate (po4) for Peter and Paul lakes
peter_paul_plot <-
  ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug, color = lakename )) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Total Phosphorus vs Phosphate",
       x = "Phosphate (PO4)",
       y = "Total Phosphorus (ug/L)") +
  xlim(0, 0.2)+
  ylim(0, 30) +
  scale_color_manual(values = c("Peter" = "green", "Paul" = "red"))

peter_paul_plot
```

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

## Warning: Removed 22942 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 22942 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.

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`.

```
# Creating boxplot for temperature
boxplot_temp <- PeterPaul.chem.nutrients %>%
  ggplot(mapping = aes(x = sampledate, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  labs(title = "Temperature by Month",
       x = "Month",
       y = "Temperature (°C)") +
  theme(axis.title.x = element_blank()) +
  theme(legend.position = "none")  # Remove legend from this plot

# Creating boxplot for tp
boxplot_tp <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x = sampledate, y = tp_ug , color = lakename)) +
  geom_boxplot() +
  labs(title = "TP by Month",
       x = "Month",
       y = "TP") +
  theme(axis.title.x = element_blank()) +
```

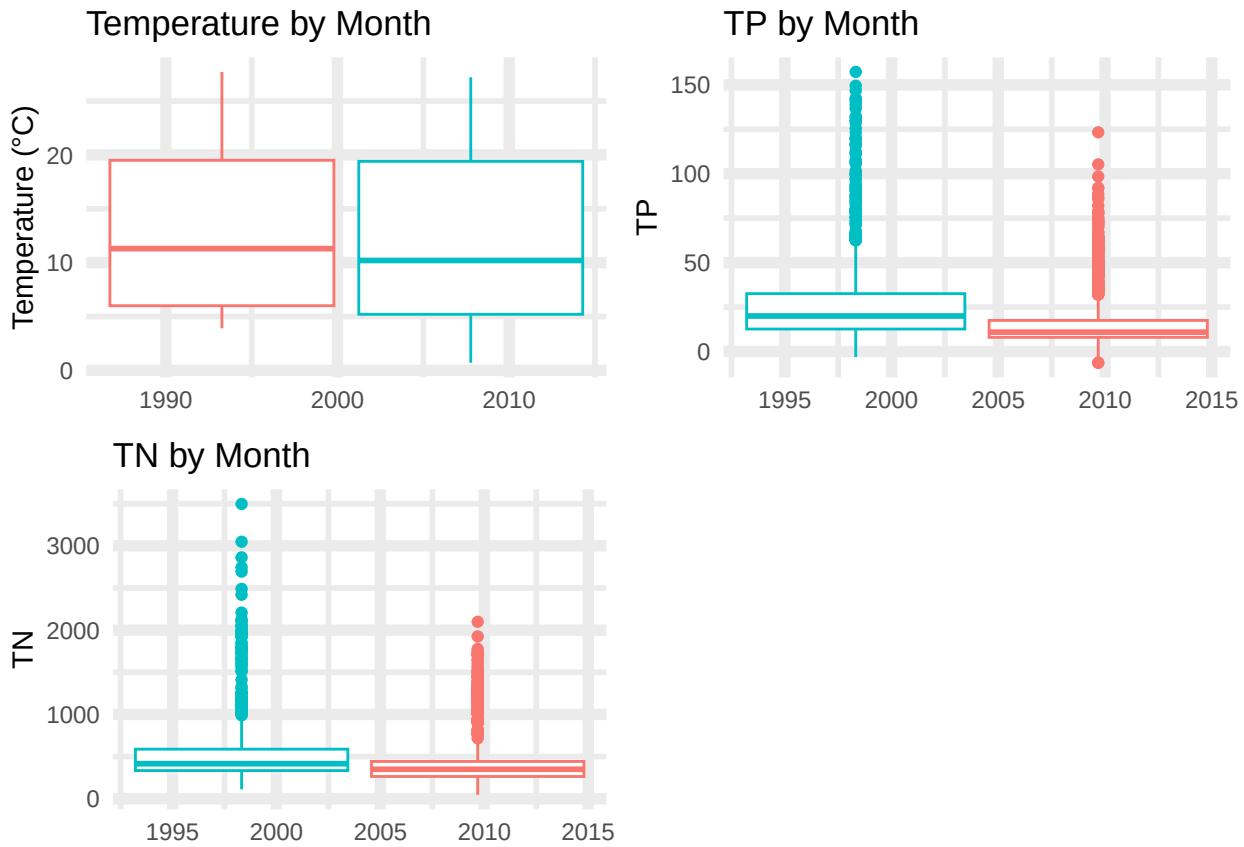
```
  theme(legend.position = "none")  # Remove legend from this plot

# Creating boxplot for tn
boxplot_tn <- PeterPaul.chem.nutrients %>%
  ggplot( aes(x = sampledate , y = tn_ug , color = lakename)) +
  geom_boxplot() +
  labs(title = "TN by Month",
       x = "Month",
       y = "TN") +
  theme(axis.title.x = element_blank()) +
  theme(legend.position = "none")  # Remove legend from this plot

# Combine the three boxplots using cowplot
combined_plot <- plot_grid(boxplot_temp, boxplot_tp, boxplot_tn)
```

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

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

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

```
# Adding a common legend
combined_plot <- combined_plot + theme(legend.position = "bottom")

# Displaying the combined plot
print(combined_plot)
```



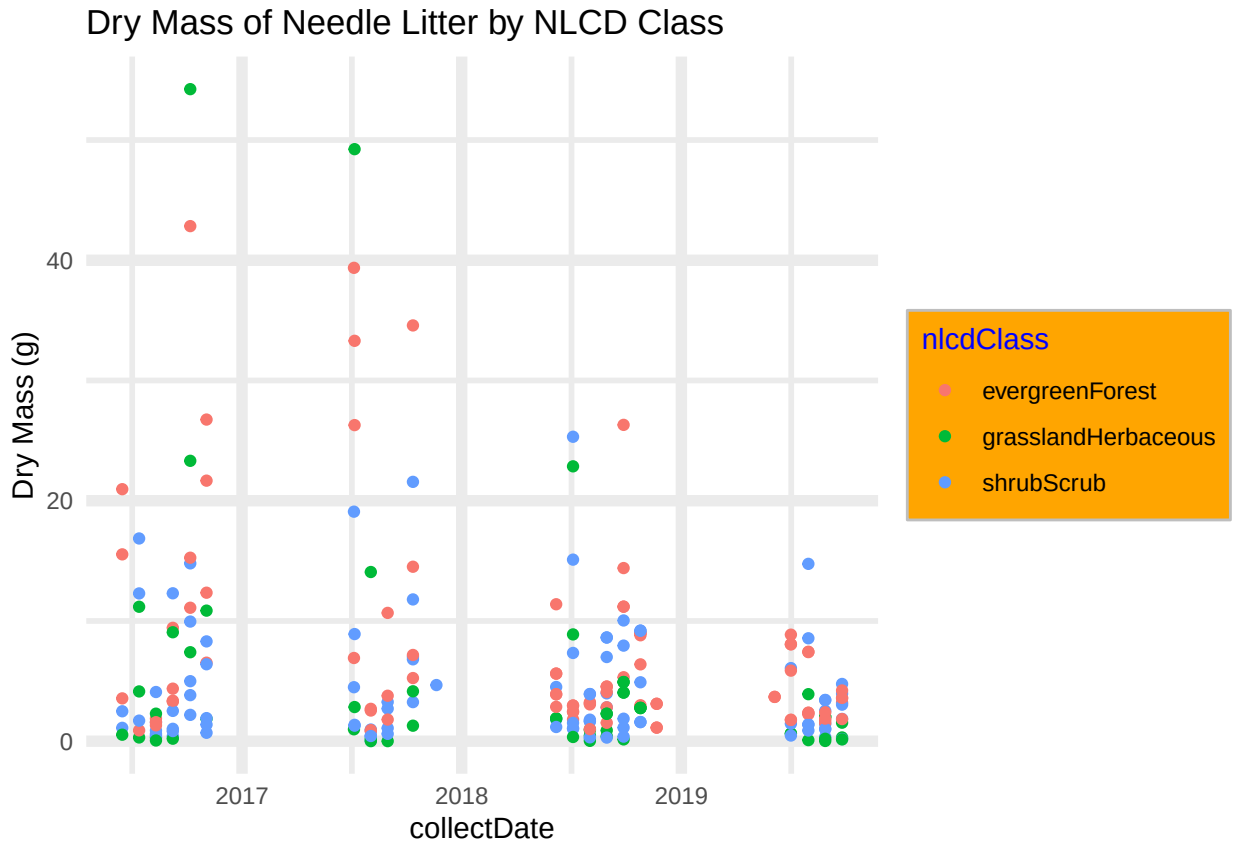
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: I observe for temperature that there is alot of variability over seasons, however both lakes are similar in temperatures. For tp and tn there is much less variability over seasons with one lake being less than other.

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)

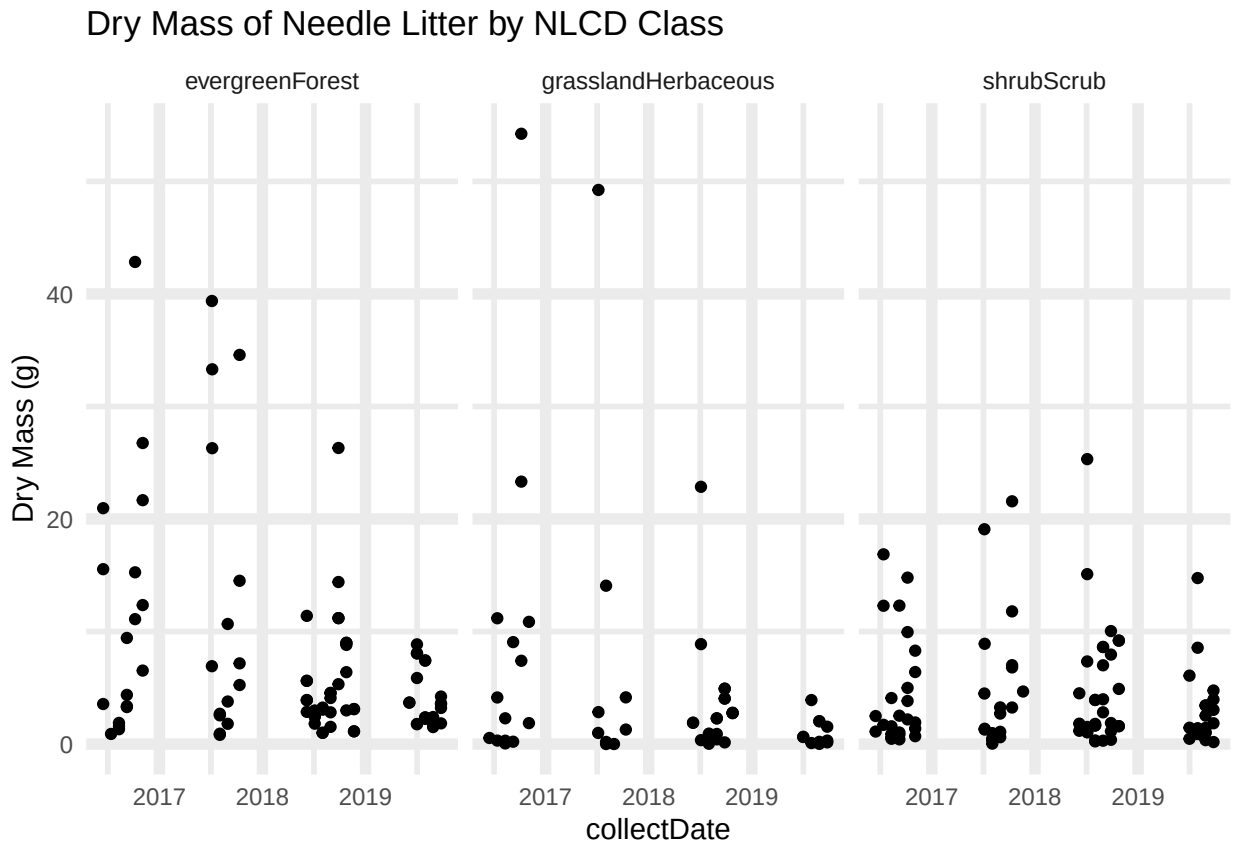
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
# Plot dry mass of needle litter by date and NLCD class with color aesthetic
Niwot_Ridge_Needles <- ggplot(NIWO_Litter %>% filter(functionalGroup == "Needles"), aes(x =collectDate,
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by NLCD Class",
       x = "collectDate",
       y = "Dry Mass (g)")


Niwot_Ridge_Needles
```




```
#7
# Plot dry mass of needle litter by date with NLCD classes separated into facets
Niwot_Ridge_facet <- ggplot(NIWO_Litter %>% filter(functionalGroup == "Needles"), aes(x = collectDate,
  geom_point() +
  facet_wrap(~ nlcdClass, ncol = 3) +
  labs(title = "Dry Mass of Needle Litter by NLCD Class",
```

```
       x = "collectDate",
       y = "Dry Mass (g)")

Niwot_Ridge_facet
```



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

> Answer: The facet plots in #7 are more effective visually because for each NLCD class you can see the relationship between the years passed (2017-2019) and the dry mass of Needle litter. Therefore, you can analyze any patterns or trends better. The plots in #6 are more confusing to visually look at because the data point overlap, versus being spread out into three facets, and therefore you can't analyze it appropriately.