Assignment 5: Data Visualization

Quinn Bankson

Fall 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. 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 packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.2
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(here)
```

```
## here() starts at /Users/mac/Documents/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggplot2)
library(dplyr)
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
getwd()
```

```
## [1] "/Users/mac/Documents/EDE_Fall2023"
```

```
here()
```

```
## [1] "/Users/mac/Documents/EDE_Fall2023"
```

```
ntl_lter <- read.csv(here
("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
niwo_litter <- read.csv(here
("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2
ntl_lter$sampledate <- ymd(ntl_lter$sampledate)
niwo_litter$collectDate <- ymd(niwo_litter$collectDate)

class(ntl_lter$sampledate)
```

```
## [1] "Date"
```

```
class(niwo_litter$collectDate)
```

```
## [1] "Date"
```

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
custom_default <- theme_base() +
  theme(
    line = element_line(
      color='darksalmon',
      linewidth =1.5
    ),
    legend.background = element_rect(
      color='orange',
      fill = 'beige'
    ),
    legend.title = element_text(
      color='darkorchid4'
    ),
    plot.background = element_rect(
      color='grey23',
      fill ='tan'
    )

  )
theme_set(custom_default)
```

Create graphs

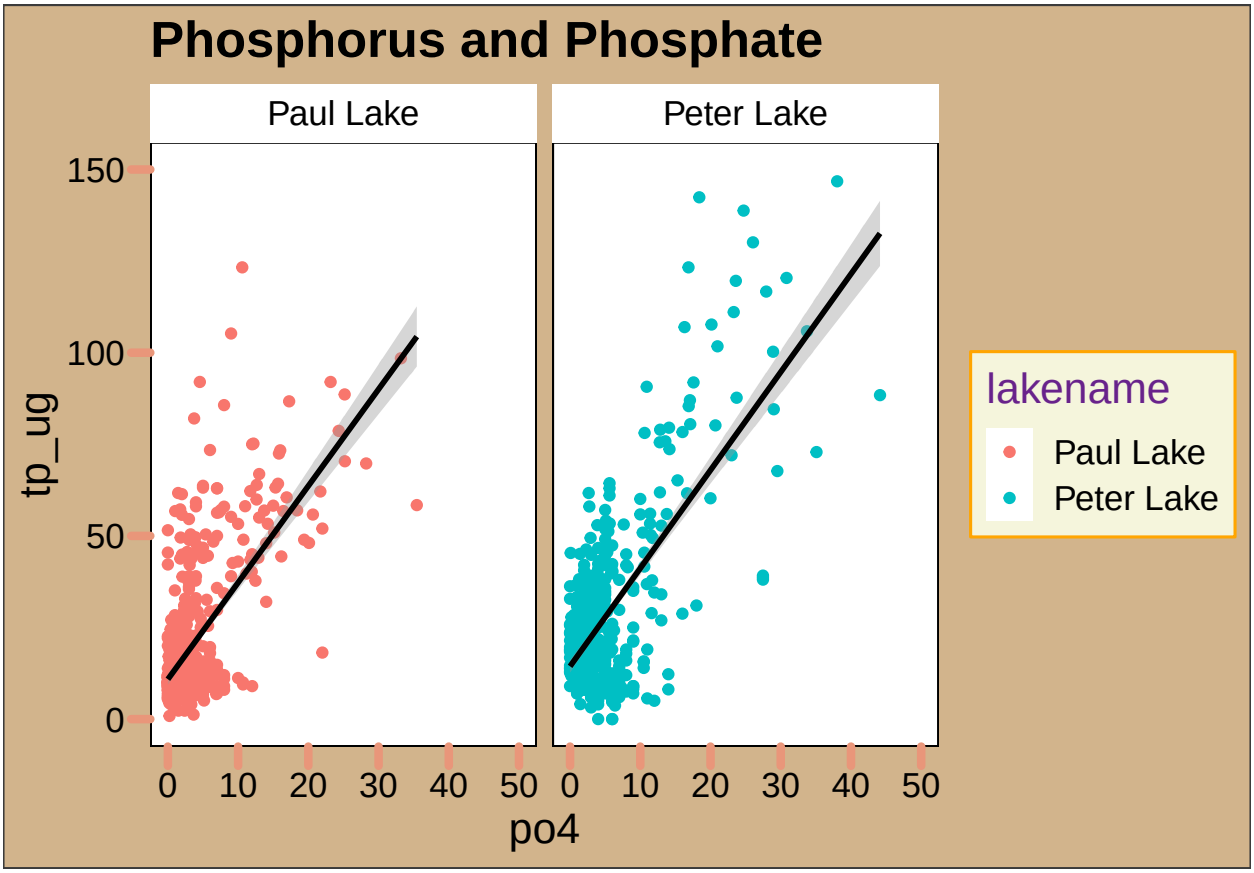
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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 phosphorus by phosophate indicates x=po4 and y=tp_ug
#"Separate aesthetics" unclear: facet used to show unique aesthetic for each plot and trend lines uniqu
phosporus_phosphate <- ntl_lter %>%
```

```
  ggplot(
    mapping = aes(
      x=po4,
      y=tp_ug,
      color=lakename)
    ) +
  geom_point()+
  facet_wrap("lakename")+
  geom_smooth(method="lm", color='black')+
  labs(title = "Phosphorus and Phosphate")+
  xlim(0, 50)+
  ylim(0, 150)

phosporus_phosphate
```

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

## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21948 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.

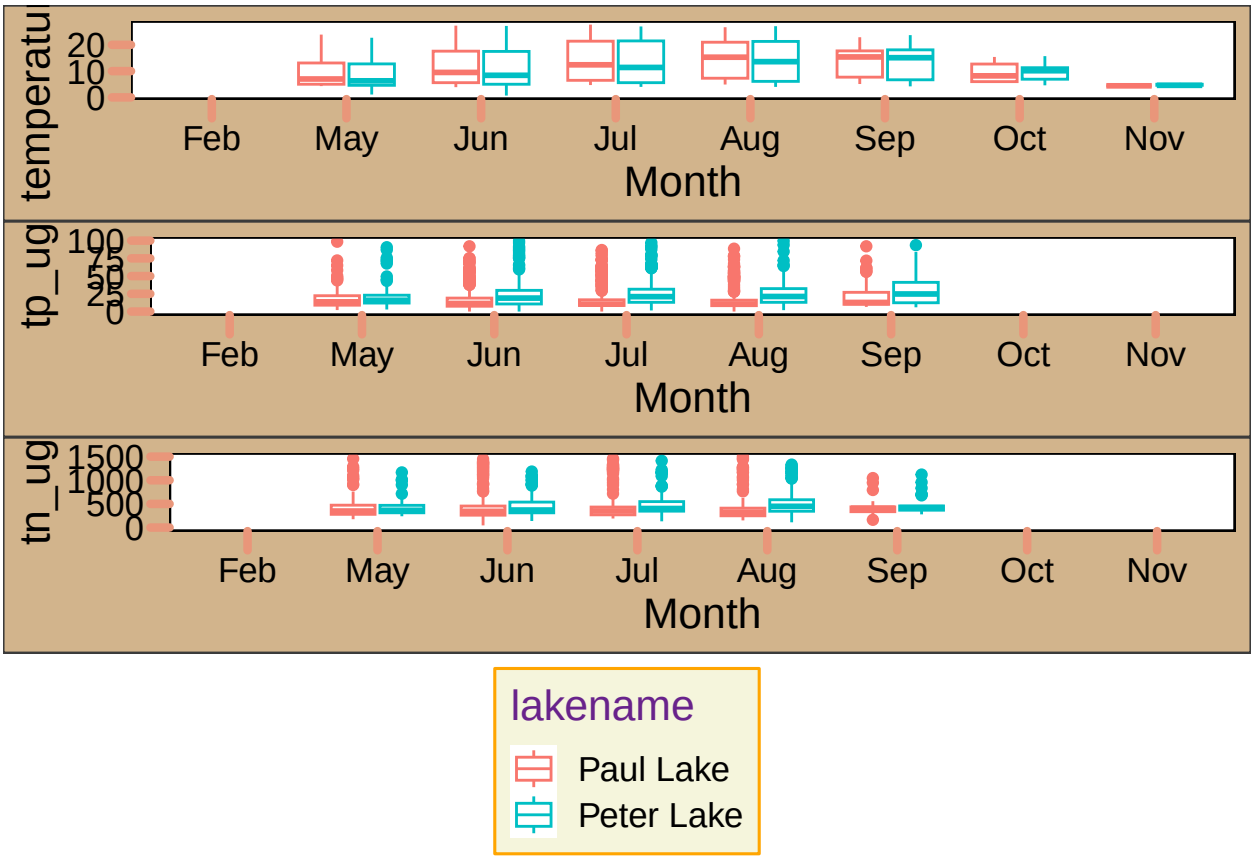
Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
temp_vis <- ntl_lter %>%
  ggplot(
    mapping = aes(
      x = factor(month, levels = 1:12,
  labels = month.abb),
      y = temperature_C,
      color = lakename
    )
  ) +
  geom_boxplot(show.legend = FALSE)+
  xlab("Month")


phosphate_vis <- ntl_lter %>%
  ggplot(
    mapping = aes(
      x=factor(month, levels = 1:12,
  labels = month.abb),
      y=tp_ug,
      color=lakename
    )
  ) +
  geom_boxplot(show.legend = FALSE)+
  ylim(0,100) +
  xlab("Month")


tn_vis <- ntl_lter %>%
  ggplot(
    mapping = aes(
      x=factor(month, levels = 1:12,
  labels = month.abb),
      y=tn_ug,
      color=lakename
    )
  ) +
  geom_boxplot(show.legend = FALSE)+
  ylim(0,1500 )+
  xlab("Month")

legend_plot <- ntl_lter %>%
  ggplot(
    mapping = aes(
      x = factor(month, levels = 1:12,
  labels = month.abb),
      y = temperature_C,
      color = lakename
    )
  ) +
  geom_boxplot()
```

```
triple_vis <- plot_grid(
  temp_vis,
  phosphate_vis,
  tn_vis,
  legend = get_legend(legend_plot),
  align = "h",
  ncol = 1
)
```

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

## Warning: Removed 20799 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Removed 21641 rows containing non-finite values (‘stat_boxplot()‘).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
triple_vis
```



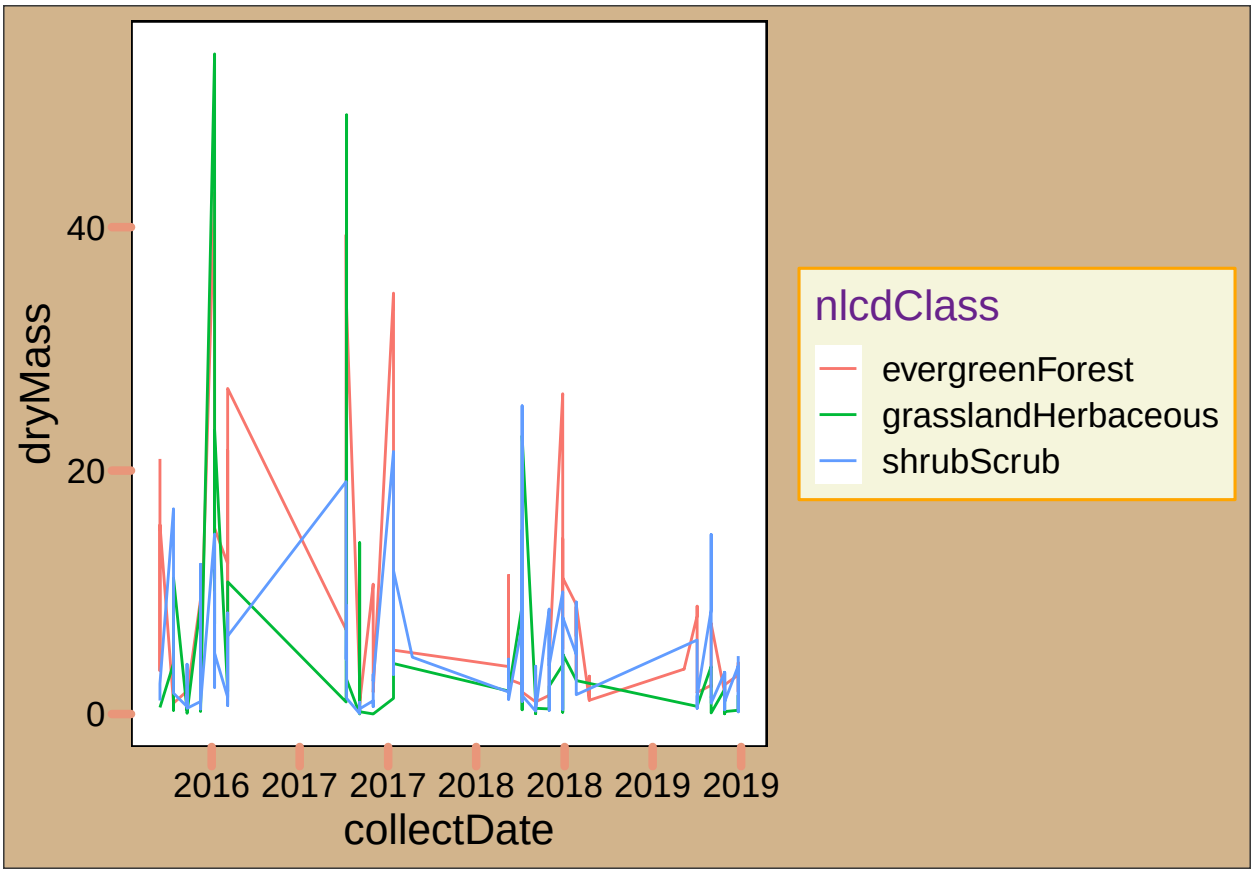
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Paul lake consistently has lower median levels of tp and tn than Peter lake. Peter lake consistently have lower median temperature levels than Paul lake. Medians for tn and tp both rise during warmer seasons for both lakes.

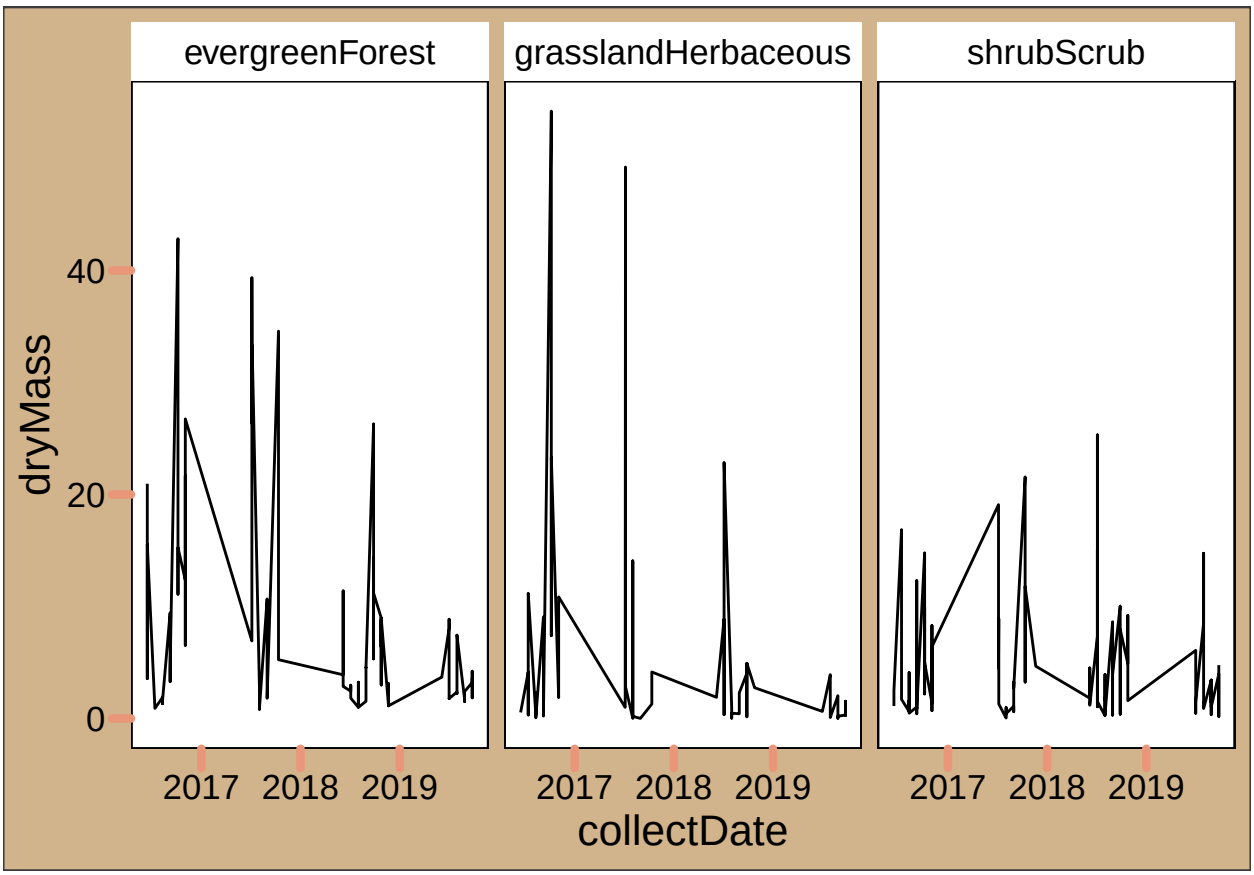
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

needle_vis <- niwo_litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x = collectDate,
      y = dryMass,
      color = nlcdClass
    )
  ) +
  geom_line()+
  scale_x_date(date_breaks = "6 months", date_labels="%Y")

needle_vis
```



```
#7
needle_wrap <- niwo_litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x = collectDate,
      y = dryMass,
    )
  ) +
  geom_line()+
  facet_wrap("nlcdClass")

needle_wrap
```



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

Answer: I think that 6 is more effective. I can more easily see how the dryMass is changing during the year in visualization no. 6. I have to look over the multiple sections of number seven and try to compare the heights and values of dry mass in different seasons, whereas six overlays the dryMass values and therefore overlays the changes in value more easily.