Assignment 5: Data Visualization

Garrett Pearce

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Monday, February 14 at 7:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and cowplot packages. Upload 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) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed.csv`] version).
2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
getwd()
```

```
## [1] "C:/Users/dgp20/Documents/ENV 872/Environmental_Data_Analytics_2022/Assignments"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
```

```
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)

peterpaul <- read.csv('../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv', str
niwotridge <- read.csv('../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv', stringsAsFactors =

#2
class(niwotridge$collectDate)
```

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

```
#Checking to see if the Niwot Ridge data column "collectDate" is read as a date- looks like it's not.

niwotridge$collectDate <- as.Date(niwotridge$collectDate, format = "%Y-%m-%d")

peterpaul$sampledate <- as.Date(peterpaul$sampledate, format = "%Y-%m-%d")

#Changed both date columns in the datasets to date format.

class(niwotridge$collectDate)
```

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

```
class(peterpaul$sampledate)
```

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

```
#Both look good!
```

Define your theme

3. Build a theme and set it as your default theme.

```
#3
newTheme <- theme(legend.position = "right", axis.text = element_text(color = "blue", size = 10),
                  legend.title = element_text(size = 15, color = "red"))

#Setting a new (potentially ugly) theme and setting as default

theme_set(newTheme)
```

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 `ylim()`).

```
#4
phos_Plot <- ggplot(peterpaul, aes(x = tp_ug, y = po4, color = lakename, shape = lakename)) +
  geom_point(alpha=0.7, size=2.5) +
```

```
  geom_smooth(color = 'black') + xlim(0,150) + ylim(0,50) +
  labs(title = 'Total Phosphorous by Phosphate') +
  xlab('Total Phosphorous') + ylab('Phosphate')

#  Note- I am not sure if geom_smooth() gives the best line of best fit here- ideally I would have
# liked straight lines, and to have one line of best fit for the graph as a whole,
# but these do give a solid overview of the trends for both lakes.
```

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.

```
#5
peterpaul$month <- as.factor(peterpaul$month)

#Temperature
temp_Plot <- ggplot(peterpaul, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  theme(legend.position = 'none') +
  xlab('Month') +
  ylab('Temperature (C)')

#Phosphorous
TP_Plot <- ggplot(peterpaul, aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  theme(legend.position = '')+
  xlab('Month') +
  ylab('Phosphorous')

#Nitrogen
TN_Plot <- ggplot(peterpaul, aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  theme(legend.position = 'none') +
  xlab('Month') +
  ylab('Nitrogen')

#Creating objects to align plots and show legend
all_Plots <- plot_grid(temp_Plot, TP_Plot, TN_Plot, nrow = 1, align = "vh")
```

```
## 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).
```

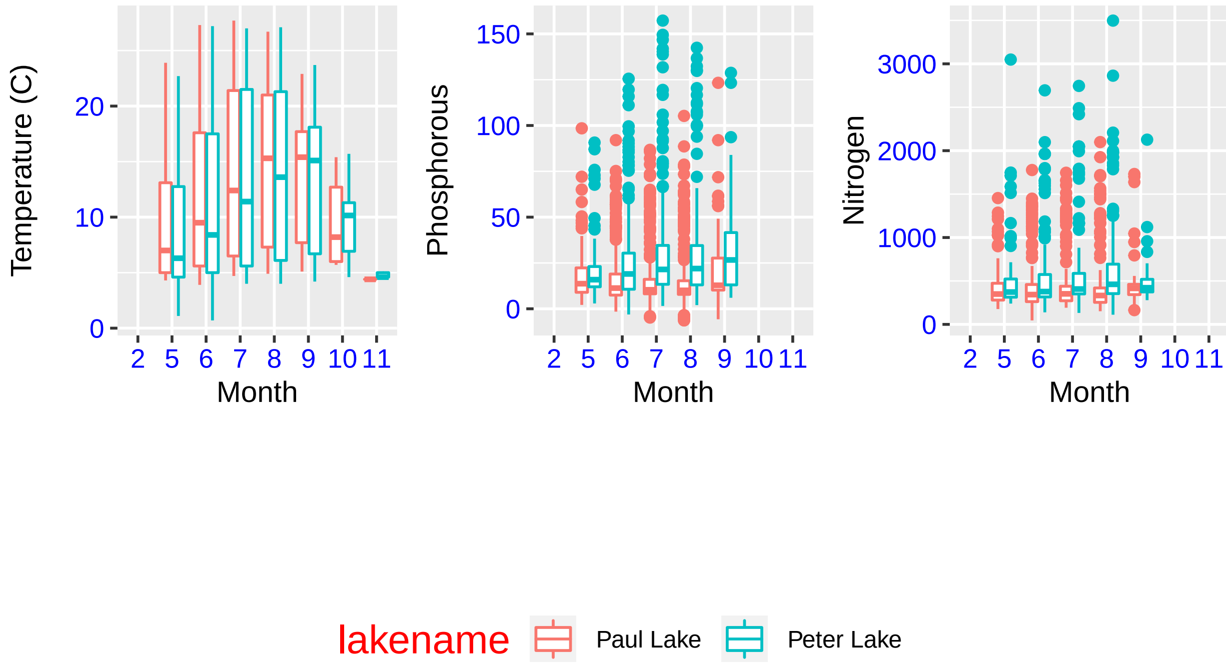
```
legend_final <- get_legend(temp_Plot +
    guides(color = guide_legend(nrow = 1)) +
    theme(legend.position = "bottom"))
```

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

```
#Combining aligned plots and legend into final product

final_Plot <- plot_grid(all_Plots, legend_final, nrow = 2)

print(final_Plot)
```



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

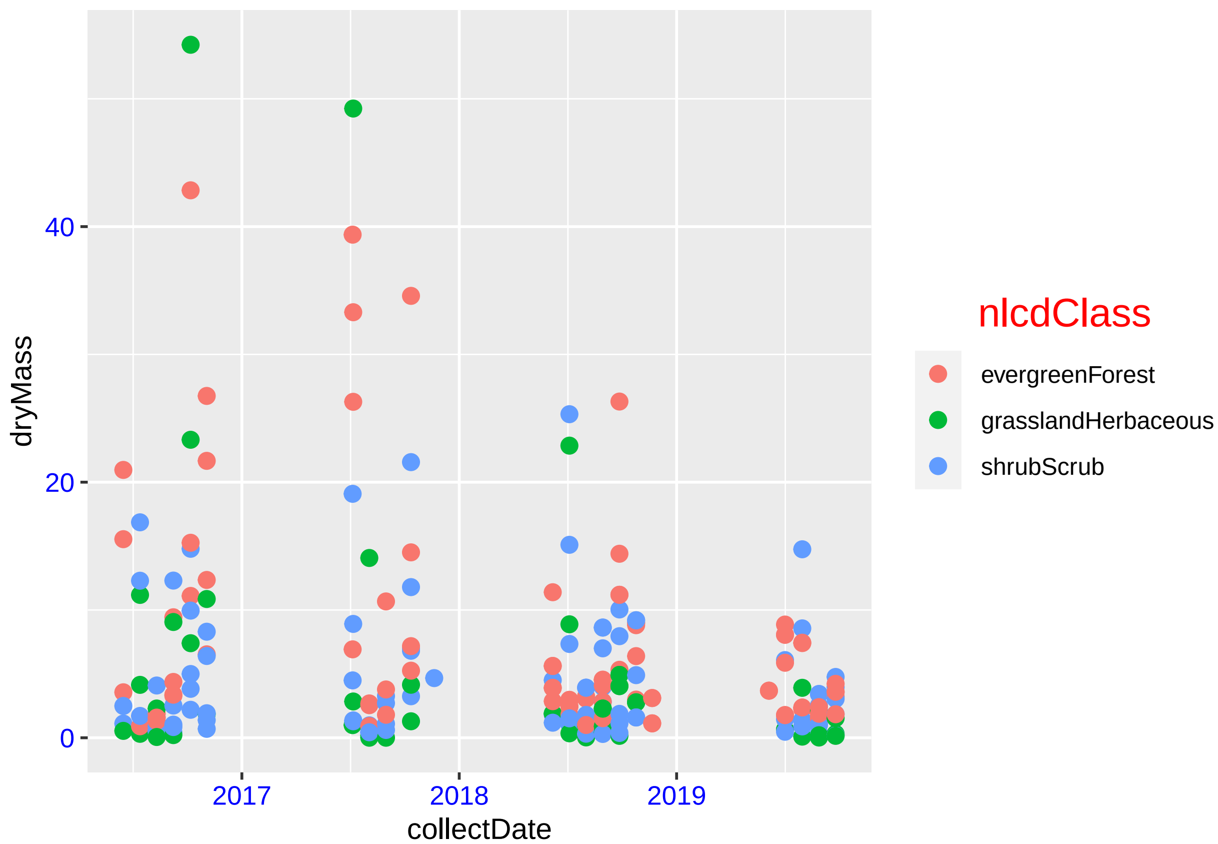
> Answer: Temperature, unsurprisingly, seems to fluctuate seasonally, but remains similar between lakes. Phosphorous and nitrogen remains relatively constant through the seasons with slightly higher levels in mid-to-late months of the year. Notably, though, there are no measurements shown in winter. Phosphorous and nitrogen levels also appear to both be lower on average at Paul Lake than at Peter Lake.

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
niwotFiltered <- filter(niwotridge, functionalGroup == 'Needles')

ggplot(niwotFiltered, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point(size = 2.5)
```




```
#7
ggplot(niwotFiltered, aes(x = collectDate, y = dryMass)) +
  geom_point(size = 2.5) +
  facet_wrap(~ nlcdClass)
```

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

Answer: I like the multiple plots from #7 more, because it allows for easier comparison across different site classes. The combined plot in #6 looks a little more cluttered, and the separate plots show overall trends better.