Assignment 5: Data Visualization

Thomas Hancock

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., "Salk_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 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 (tidy and gathered) and the processed data file for the Niwot Ridge litter dataset.

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
getwd()
```

```
## [1] "C:/Users/thoma/Thomas/2018 Grad School/Duke MEM/ENV 872/Environmental_Data_Analytics_2020"
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(cowplot)
```

```
##
## ************************************************
```

```
## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ******************************************************
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
# Load data
NTL.PP.nut.tidy <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
NTL.PP.nut.gathered <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.cs
NIWO.Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(NTL.PP.nut.tidy$sampledate) # Check date format
```

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

```
#Change dates to read as dates
NTL.PP.nut.tidy$sampledate <-
  as.Date(NTL.PP.nut.tidy$sampledate, format = "%Y-%m-%d")
NTL.PP.nut.gathered$sampledate <-
  as.Date(NTL.PP.nut.gathered$sampledate, format = "%Y-%m-%d")
NIWO.Litter$collectDate <- as.Date(NIWO.Litter$collectDate, format = "%Y-%m-%d")

class(NTL.PP.nut.tidy$sampledate) # Verify date format
```

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

Define your theme

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

```
myTheme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top") # Define a theme based off of the classic theme

theme_set(myTheme) # Set defined theme to 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.

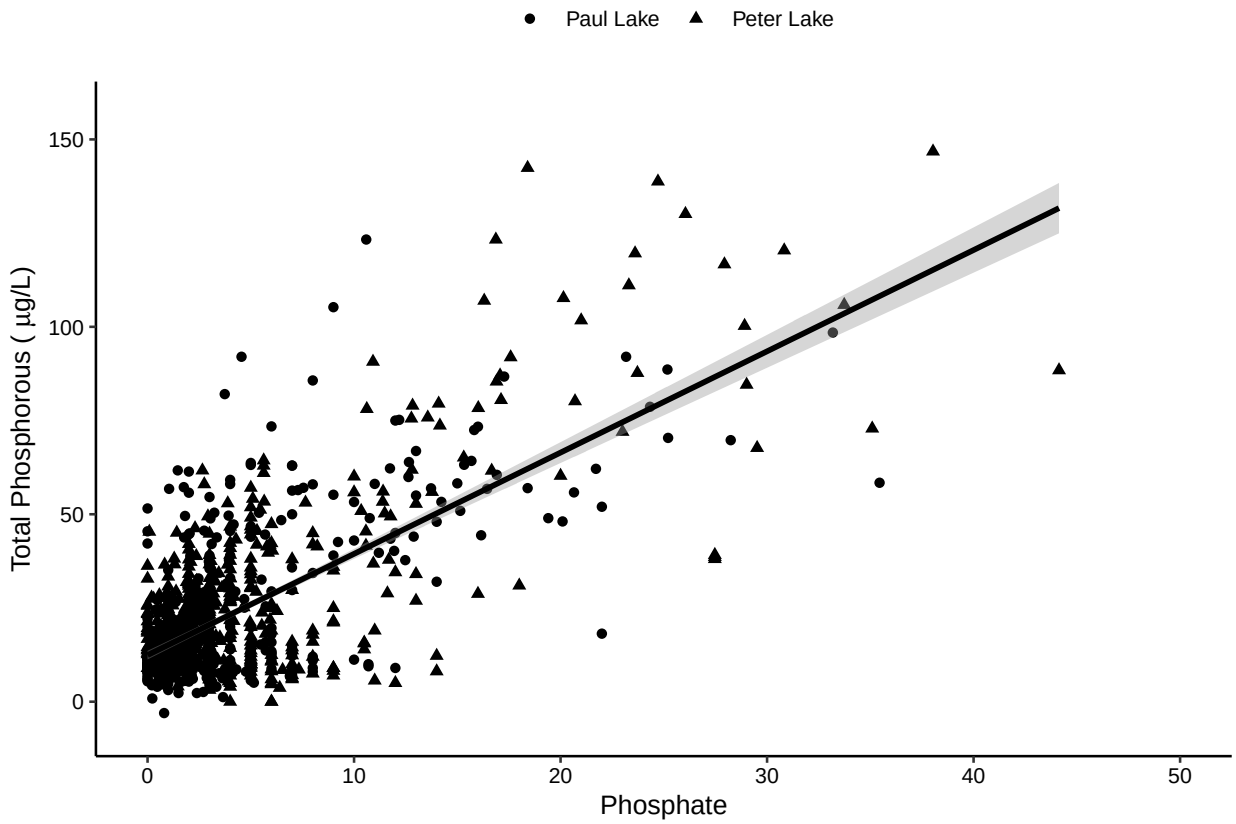
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values.

```
ppPlot1 <- ggplot(NTL.PP.nut.tidy, aes(x = po4, y = tp_ug)) + # Plot of tp vs po4
  geom_point(aes(shape = lakename)) + # Scatter plot with lakes as different shapes
  geom_smooth(method = lm, color = "black") + # Add linear best fit line
  xlim(0,50) + # Limit x-axis to hide outliers
  labs(x = "Phosphate", y = expression(paste("Total Phosphorous ( ", mu, "g/L)")),
       shape = "") # Change axis and legend labels

print(ppPlot1 )
```

```
## Warning: Removed 21947 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21947 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.

```
# Make Temperature boxplots
ppTempPlot <- ggplot(NTL.PP.nut.tidy) +
  geom_boxplot(aes(x = as.factor(month), y = temperature_C, color = lakename)) +
  labs(x = "Month", y = "Temperature (C)", color = "") +
  scale_color_manual(values = c("#0c2c84", "#ea6827ff")) # Set colors

#ppTempPlot # Used for debugging

#Make Phosphorous boxplots
ppTPPlot <- ggplot(NTL.PP.nut.tidy) +
  geom_boxplot(aes(x = as.factor(month), y = tp_ug, color = lakename)) +
  labs(x = "Month", y = expression(paste("Total Phosphorous ( ", mu, "g/L)"))) +
  scale_color_manual(values = c("#0c2c84", "#ea6827ff")) # Set colors

# Make Nitrogen boxplots
ppTNPlot <- ggplot(NTL.PP.nut.tidy) +
  geom_boxplot(aes(x = as.factor(month), y = tn_ug, color = lakename)) +
  labs(x = "Month", y = expression(paste("Total Nitrogen ( ", mu, "g/L)"))) +
  scale_color_manual(values = c("#0c2c84", "#ea6827ff")) # Set colors

# Extract legend from one of the plots to include in combined cowplot
```

```
ppLegend <- get_legend(
    ppTempPlot + theme(legend.box.margin = margin(0, 0, 0, 12)))
```

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

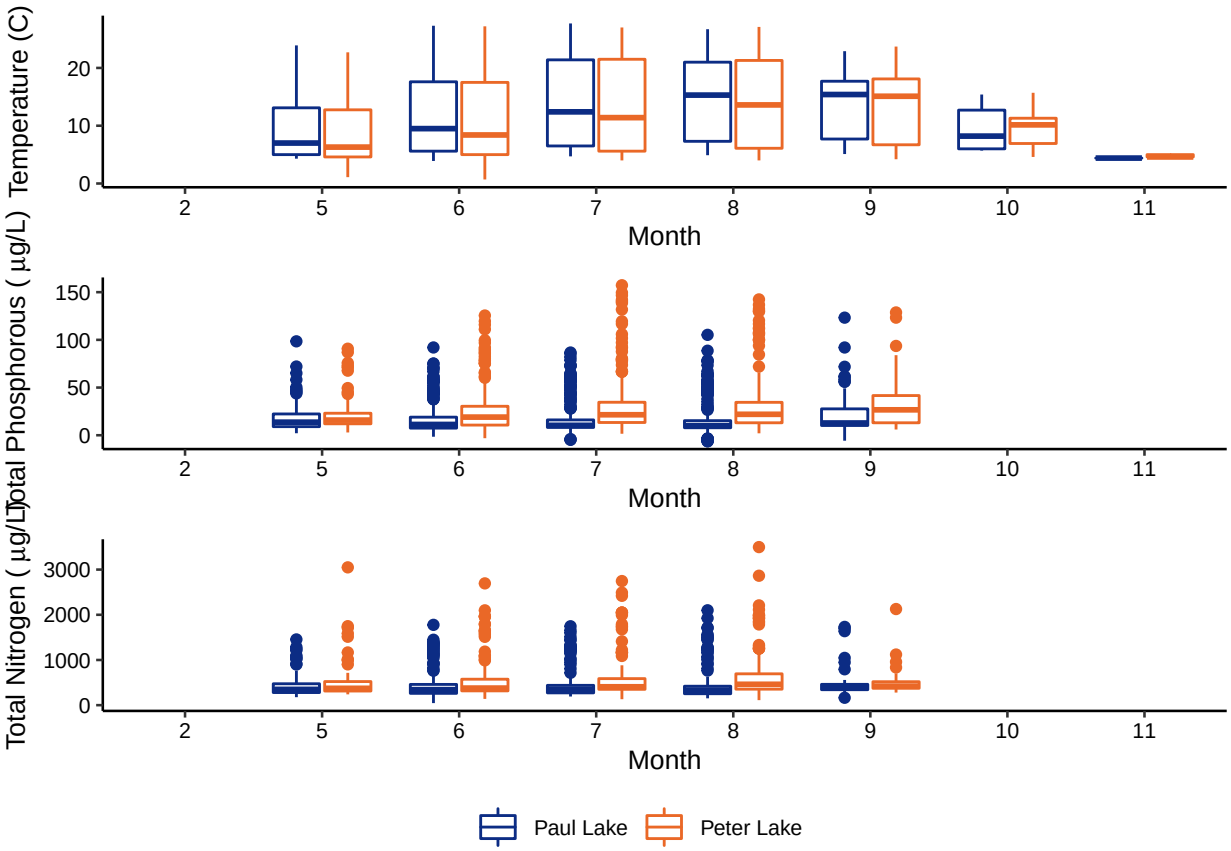
```
# Create a plot of the three sets of boxplots
ppPlotGrid <- plot_grid(ppTempPlot + theme(legend.position="none"), # Remove legends
                        ppTPPlot + theme(legend.position="none"),
                        ppTNPlot + theme(legend.position="none"),
                        align = 'v', ncol = 1) # Align axes, limit to one column
```

```
## 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).
```

```
# Create plot with boxplots and legend
ppPlot2 <- plot_grid(ppPlotGrid, ppLegend, ncol = 1, rel_heights = c(3,.3))
print(ppPlot2)
```



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

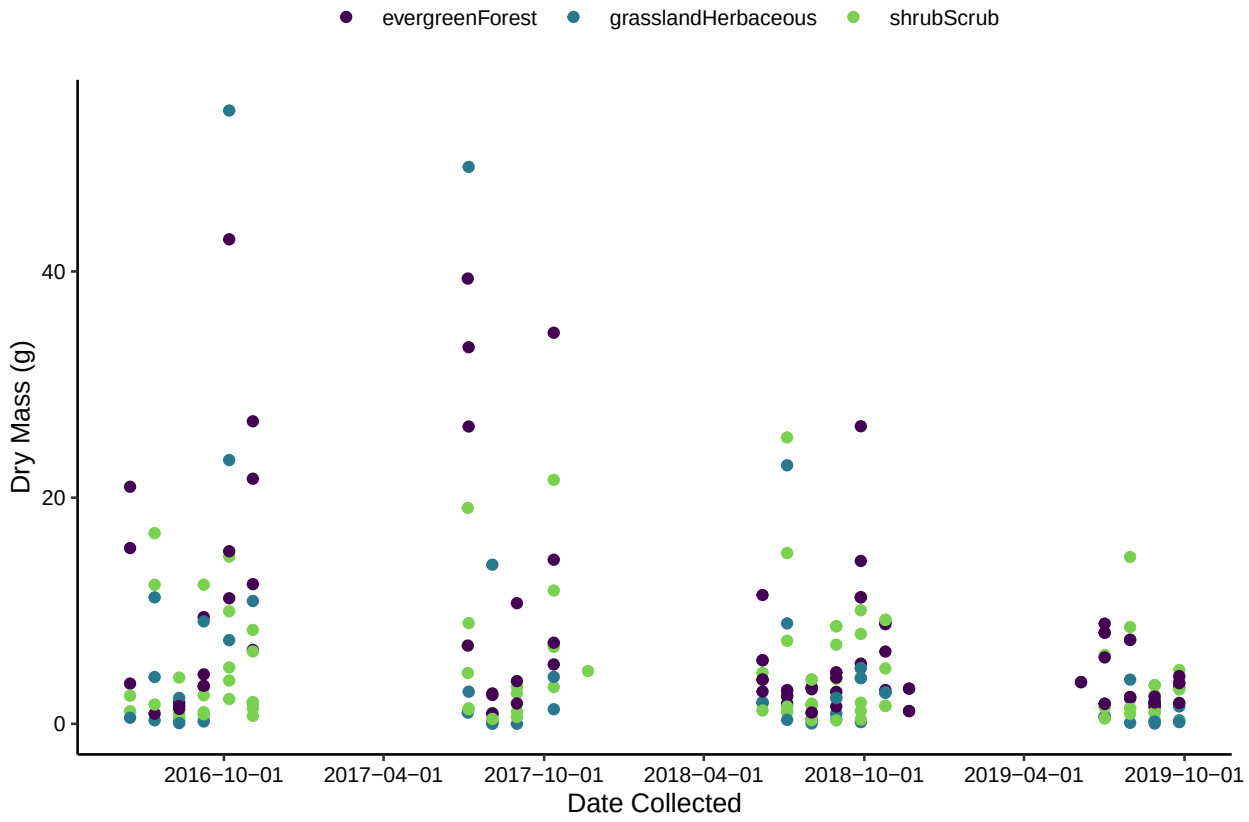
Answer: As would be expected, the median temperature for both lakes rises during the summer months and is lower towards both ends of the year (in the winter). There does not appear to be an appreciable difference between the temperatures of the two lakes. Total phosphorous is similar between the two lakes, but it seems like Peter Lake tends to have a slightly higher concentration, especially in later summer months. A similar pattern is seen regarding nitrogen concentration. Phosphorous and Nitrogen measurements were not taken for colder months for either 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 NCLD classes separated by color
nrPlot1 <- ggplot(subset(NIWO.Litter, functionalGroup == "Needles")) +
  geom_point(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  labs(x = "Date Collected", y = "Dry Mass (g)", color = "") +
  scale_color_viridis(discrete = TRUE, end = 0.8) + # Set color scheme
  scale_x_date(date_breaks = "6 months") # Change the breaks on the x-axis for readability

print(nrPlot1)
```



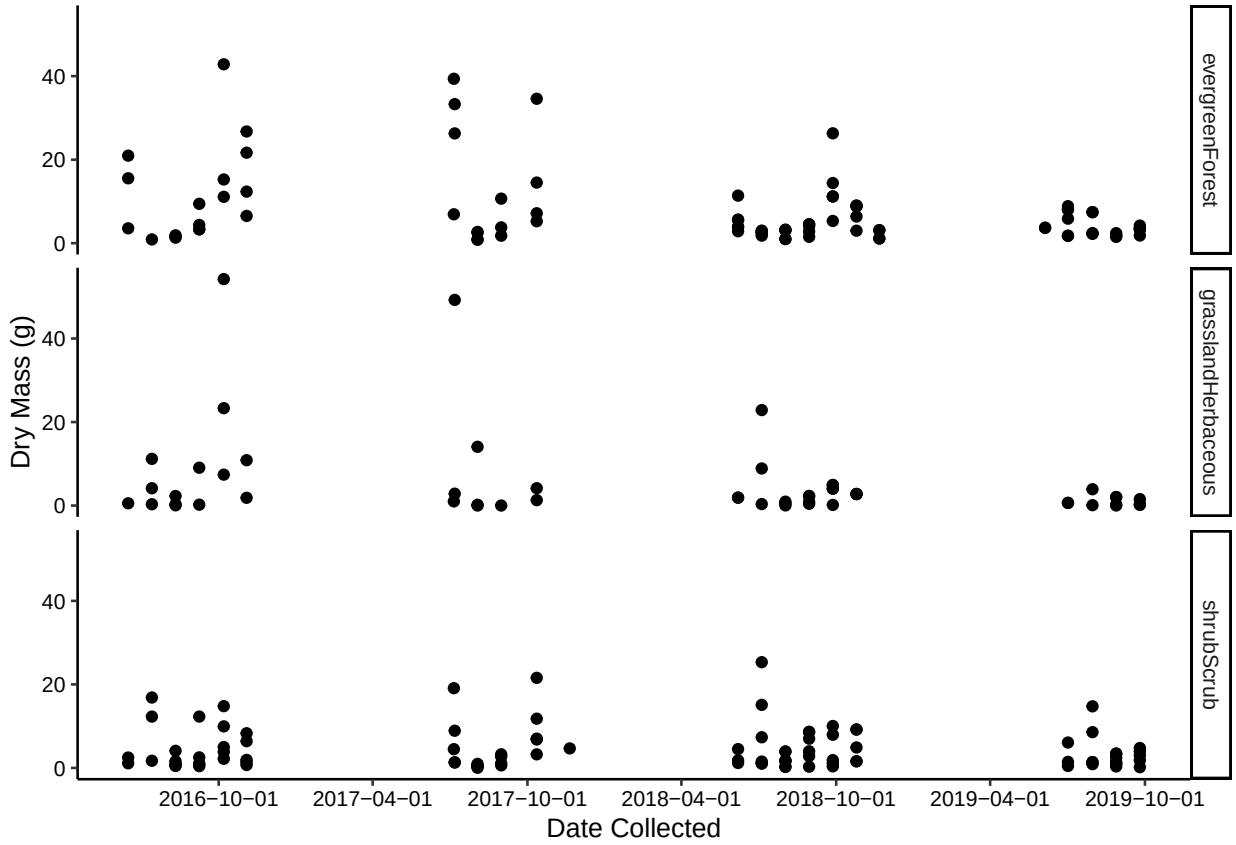

```
#nrPlot2 <- ggplot(subset(NIWO.Litter, functionalGroup == "Needles")) +
#  geom_boxplot(aes(x = as.factor(collectDate), y = dryMass, color = nlcdClass)) +
#  scale_color_viridis(discrete = TRUE, end = 0.8)

#print(nrPlot2)

# 7 NLCD Classes separated on different plots
nrPlot3 <- ggplot(subset(NIWO.Litter, functionalGroup == "Needles")) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  facet_grid(vars(nlcdClass)) + # Create facet grid based on NCLD Class
  scale_x_date(date_breaks = "6 months") + # Change the breaks on the x-axis for readability
```

```
  labs(x = "Date Collected", y = "Dry Mass (g)")

print(nrPlot3)
```



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

> Answer: Plot 6 seems more effective because you can see the points side-by-side and on top of each other on the same set of axes. This format allows you to easily see how they compare to each other. In plot 7, it is hard to really determine if the points in one NLCD class are relatively higher or lower than a different class unless there is extreme variation.