# Assignment 5: Data Visualization

Sierra Kindley

## 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() #verify working directory
```

```
## [1] "C:/Users/sierr/Documents/Duke University/ENVIRON-872L/Environmental_Data_Analytics_2020"
```

```
library(tidyverse) #load tidyverse packages
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyv
```

```
## 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_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot) #load cowplot package
```

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

```
## 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())
```

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

```
PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
#upload NTL-LTER processed data files
NiwotRidge.litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
#upload Niwot Ridge litter processed data file

#2
class(PeterPaul.chem.nutrients$sampledate) #check class of 'sampledate' column
```

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

```
PeterPaul.chem.nutrients$sampledate <-
  as.Date(PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
  #format 'sampledate' column as date

class(PeterPaul.chem.nutrients.gathered$sampledate) #check class of 'sampledate' column
```

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

```
PeterPaul.chem.nutrients.gathered$sampledate <-
  as.Date(PeterPaul.chem.nutrients.gathered$sampledate, format = "%Y-%m-%d")
  #format 'sampledate' column as date

class(NiwotRidge.litter$collectDate) #check class of 'collectDate' column
```

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

```
NiwotRidge.litter$collectDate <-
  as.Date(NiwotRidge.litter$collectDate, format = "%Y-%m-%d")
  #format 'collectDate' column as date
```

## Define your theme

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

```
sierratheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
#build new theme 'sierratheme' and define what its parameters
theme_set(sierratheme) #set sierratheme as default theme
```

## Create graphs

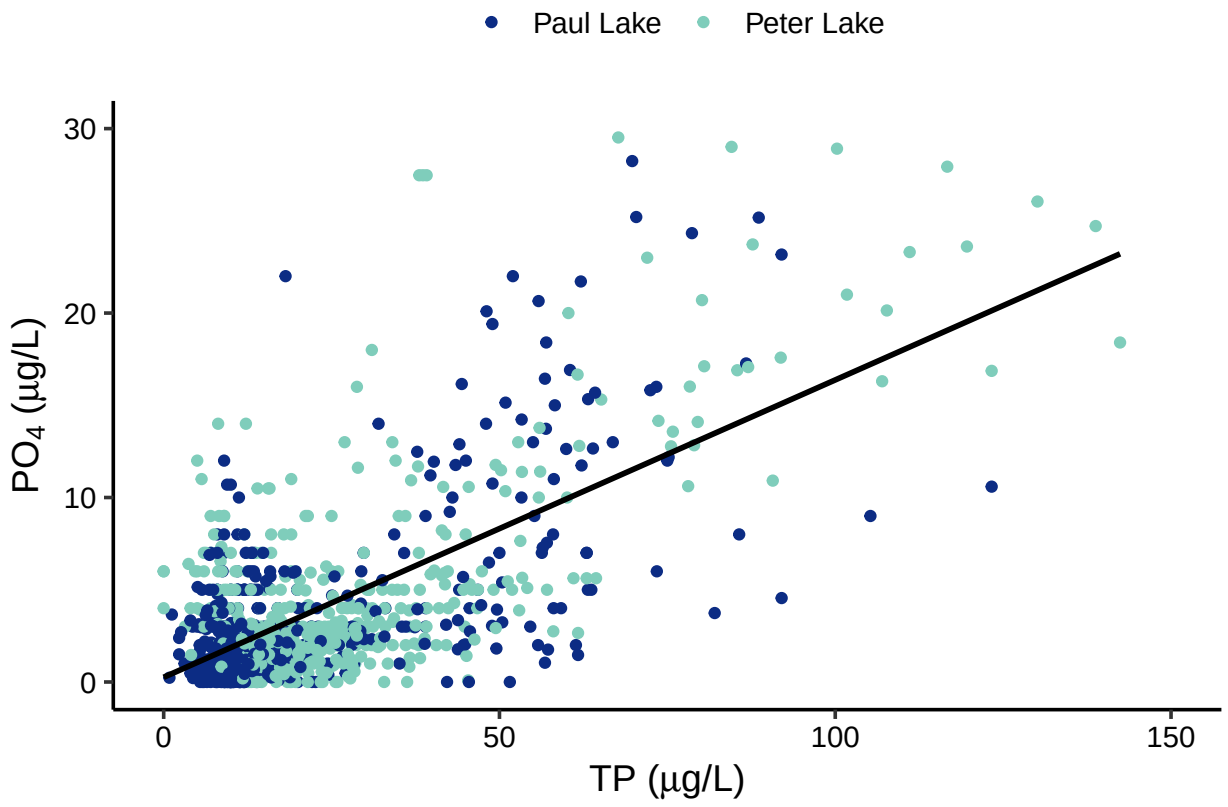
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus 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.

```
PeterPaulPlot1 <- ggplot(PeterPaul.chem.nutrients,
                         aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() +
  ylim(0,30) + #define limits of y-axis
  xlim(0,150) + #define limits of x-axis
  ylab(expression(paste("PO"[4]* " (", mu, "g/L)"))) + #change y-label
  xlab(expression(paste("TP (", mu, "g/L)"))) + #change x-label
  geom_smooth(method = lm, se = FALSE, color = "black") + #add black line of best fit
  scale_color_manual(values = c("#0c2c84", "#7fcdbb")) + #assign colors to Peter & Paul Lakes
  theme(legend.title = element_blank()) #remove legend title
print(PeterPaulPlot1) #print plot
```

```
## Warning: Removed 21955 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21955 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.

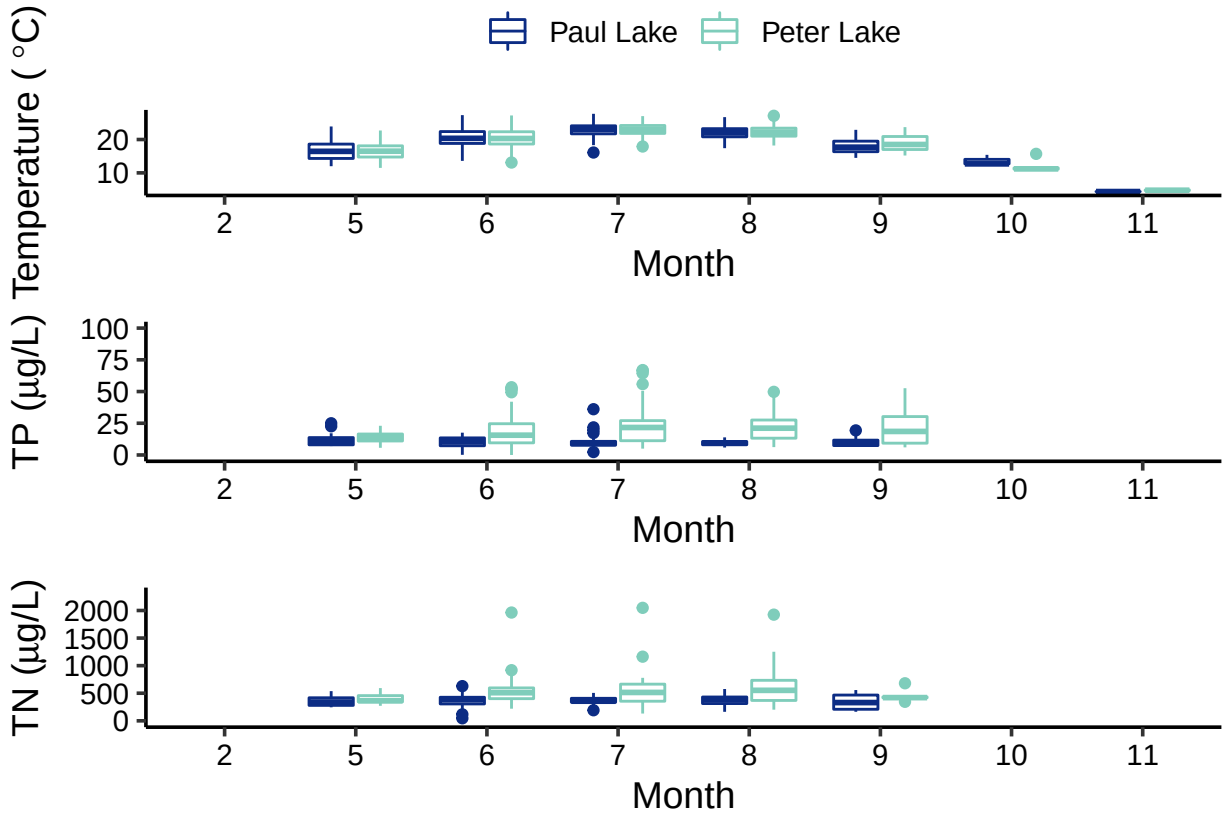
```
PeterPaulPlot2 <- ggplot(subset(PeterPaul.chem.nutrients, depth < 0.25)) +
  geom_boxplot(aes(x = as.factor(month), y = temperature_C, color = lakename)) +
  xlab(expression(paste("Month"))) + #change x-label
  ylab(expression(paste("Temperature ("*~degree*C*")")))+ #change y-label
  scale_color_manual(values = c("#0c2c84", "#7fcdbb")) + #assign colors to Peter & Paul Lakes
  theme(legend.title = element_blank()) #remove legend title
```

```
PeterPaulPlot3 <- ggplot(subset(PeterPaul.chem.nutrients, depth < 0.25)) +
  geom_boxplot(aes(x = as.factor(month), y = tp_ug, color = lakename)) +
  xlab(expression(paste("Month"))) + #change x-label
  ylab(expression(paste("TP (", mu, "g/L)"))) + #change y-label
  ylim(0,100) + #define limits of y-axis
  scale_color_manual(values = c("#0c2c84", "#7fcdbb")) + #assign colors to Peter & Paul Lakes
  theme(legend.position = "none") #remove legend from plot

PeterPaulPlot4 <- ggplot(subset(PeterPaul.chem.nutrients, depth < 0.25)) +
  geom_boxplot(aes(x = as.factor(month), y = tn_ug, color = lakename)) +
  xlab(expression(paste("Month"))) + #change x-label
  ylab(expression(paste("TN (", mu, "g/L)"))) + #change y-label
  ylim(0,2300) + #define limits of y-axis
  scale_color_manual(values = c("#0c2c84", "#7fcdbb")) + #assign colors to Peter & Paul Lakes
  theme(legend.position = "none") #remove legend from plot

plot_grid(PeterPaulPlot2, PeterPaulPlot3, PeterPaulPlot4, nrow = 3,
          align = 'v', rel_heights = c(1.25, 1, 1))
```

```
## Warning: Removed 17 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 806 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 872 rows containing non-finite values (stat_boxplot).
```




```
#combine three previous graphs into one cowplot with one legend and aligned axes
```

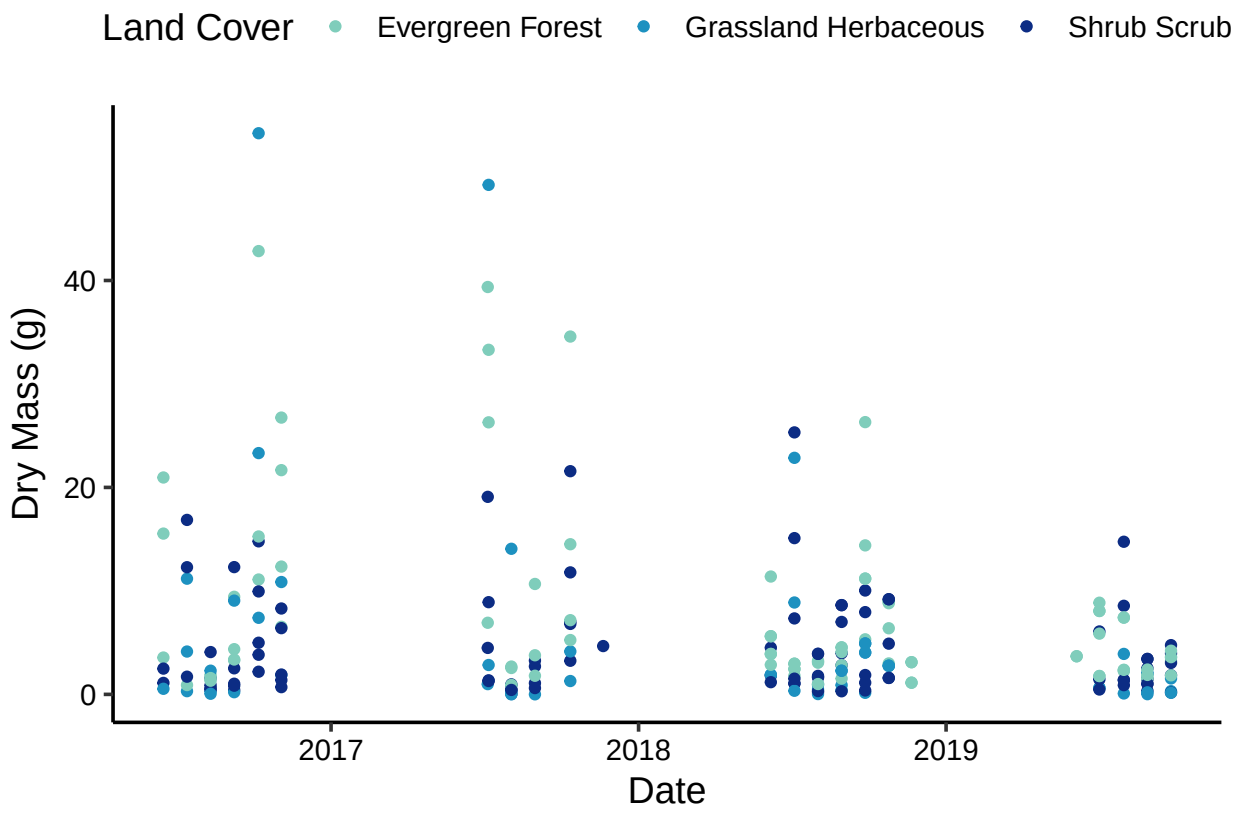
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: The temperature in the lakes appears slightly higher in the summer months and, in general, the temperature does not vary too much between lakes. Total phosphorous does not appear to vary much over seasons, but the TP values in Peter Lake are consistently higher than those in Paul Lake. Similarly, total nitrogen values do not vary much over seasons, but the TP values in Peter Lake are consistently higher than those in Paul Lake.

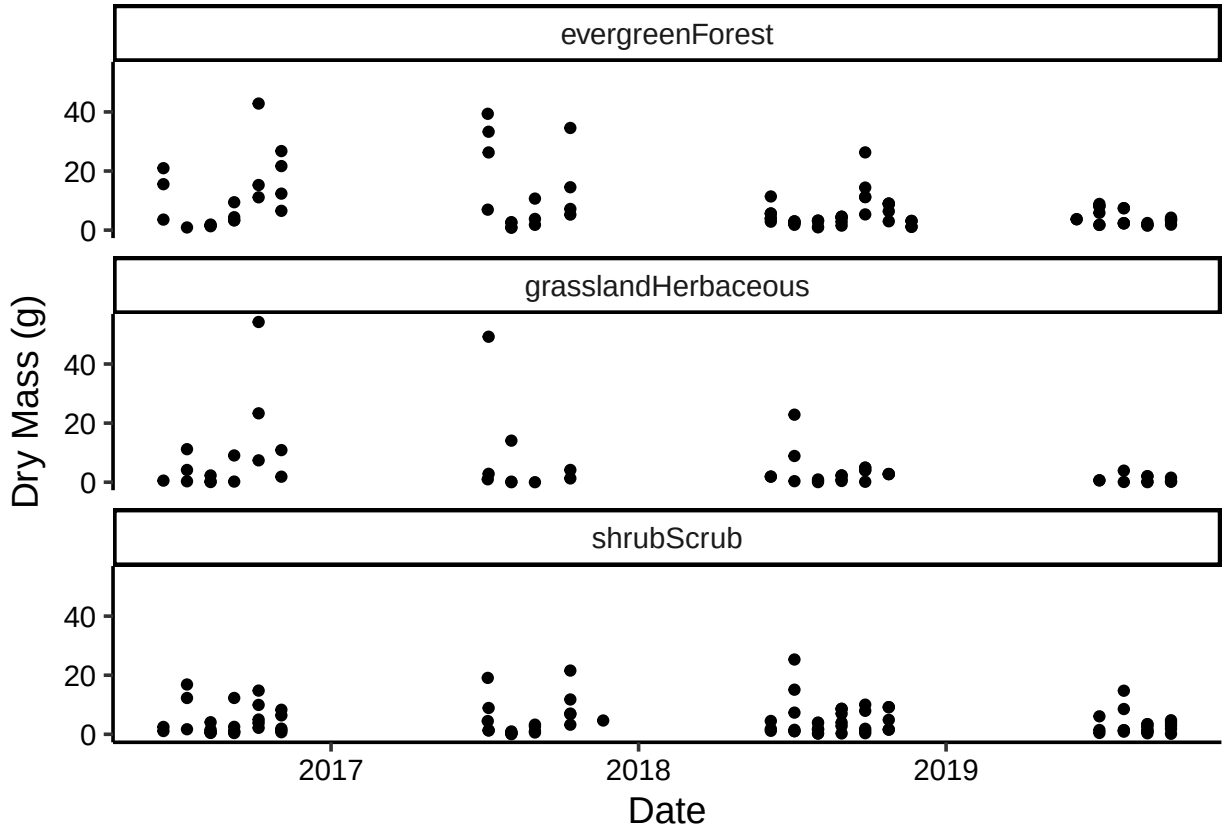
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NiwotRidgePlot1 <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles"),
                aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  xlab(expression(paste("Date"))) + #change x-label
  ylab(expression(paste("Dry Mass (g)"))) + #change y-label
  scale_color_manual(values = c("#7fcdbb", "#1d91c0", "#0c2c84"),
                     labels = c(expression("Evergreen Forest"),
                                expression("Grassland Herbaceous"),
                                expression("Shrub Scrub"))) +
  #assign colors & labels to data by NLCD class
  labs(color = "Land Cover")
print(NiwotRidgePlot1) #print plot
```



```r
#7
NiwotRidgePlot1.faceted <-
  ggplot(subset(NiwotRidge.litter, functionalGroup == "Needles")) +
  geom_point(aes(x = collectDate, y = dryMass)) +
  facet_wrap(vars(nlcdClass), nrow = 3) + #create facets by NLCD class
  xlab(expression(paste("Date"))) + #change x-label
  ylab(expression(paste("Dry Mass (g)"))) + #change y-label
  theme(legend.position = "none") #remove legend from plot
print(NiwotRidgePlot1.faceted) #print plot
```




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

Answer: I think the faceted plot (plot in question 7) is more effective, as it is more visually appealing and understandable. I think separating the data into three facets based on NLCD class helps greatly with the overall interpretation of the data, as it separates it out and allows for clearer interpretation and easier comparison between NLCD classes. The plot in question 6 is okay, but because all the data is lumped together on a single plot, I find it much more difficult to visualize and interpret the data, as well as distinguish and make comparisons between the three NLCD classes.