Assignment 5: Data Visualization

Aislinn McLaughlin

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) 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] "/Users/Aislinn/Documents/GitHub/Environmental_Data_Analytics_2021/Assignments"
```

```
setwd("~/Documents/GitHub/Environmental_Data_Analytics_2021")
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```
library(cowplot)
#install.packages("viridisLite")
library(viridis)
```

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

```r
library(RColorBrewer)
library(colormap)
library(ggplot2)

chemnutrients_PeterPaul_processed <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterP

nutrients_PeterPaul_gathered <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Pro

Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")


#2

chemnutrients_PeterPaul_processed$sampledate <-
  as.Date(chemnutrients_PeterPaul_processed$sampledate, "%Y-%m-%d")

nutrients_PeterPaul_gathered$sampledate <-
  as.Date(nutrients_PeterPaul_gathered$sampledate, "%Y-%m-%d")

Litter$collectDate <-
  as.Date(Litter$collectDate, "%Y-%m-%d")
```

Define your theme

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

```r
mytheme <-
  theme_gray(base_size = 12) +
  theme(axis.title = element_text(color = "black"), legend.position = "bottom", legend.background = elem

theme_set(mytheme)
```

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.

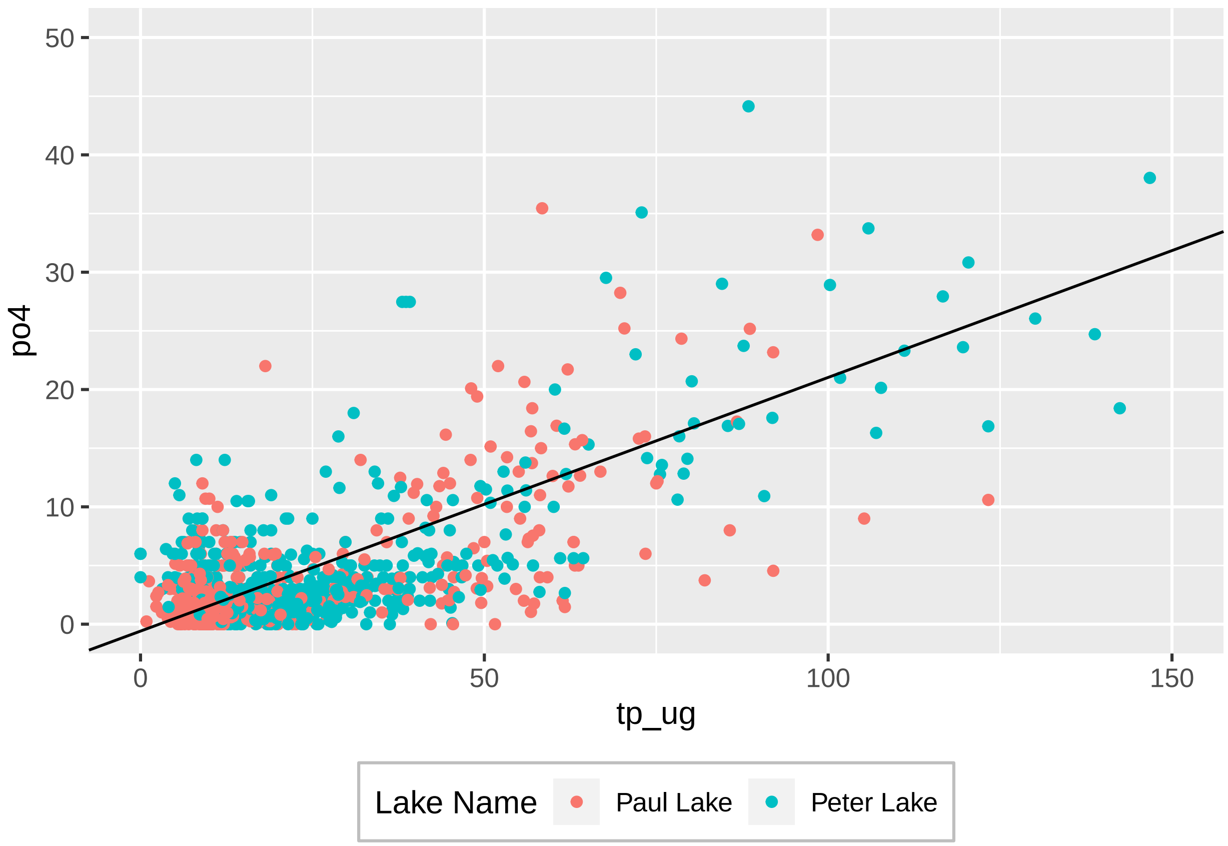
```r
#get slope and intercept for linear regression
lm(formula = chemnutrients_PeterPaul_processed$po4 ~ chemnutrients_PeterPaul_processed$tp_ug)
```

```
##
## Call:
## lm(formula = chemnutrients_PeterPaul_processed$po4 ~ chemnutrients_PeterPaul_processed$tp_ug)
##
## Coefficients:
##                            (Intercept)
##                                -0.5894
## chemnutrients_PeterPaul_processed$tp_ug
##                                 0.2162
```

```r
phos_plot <- ggplot(chemnutrients_PeterPaul_processed, aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() +
```

```
  geom_abline(slope = 0.2162, intercept = -0.5894) + #using geom_smooth(method = lm) gives 2 lines, but
  xlim(0, 150) +
  ylim(0, 50) +
  labs(color = "Lake Name")
print(phos_plot)
```

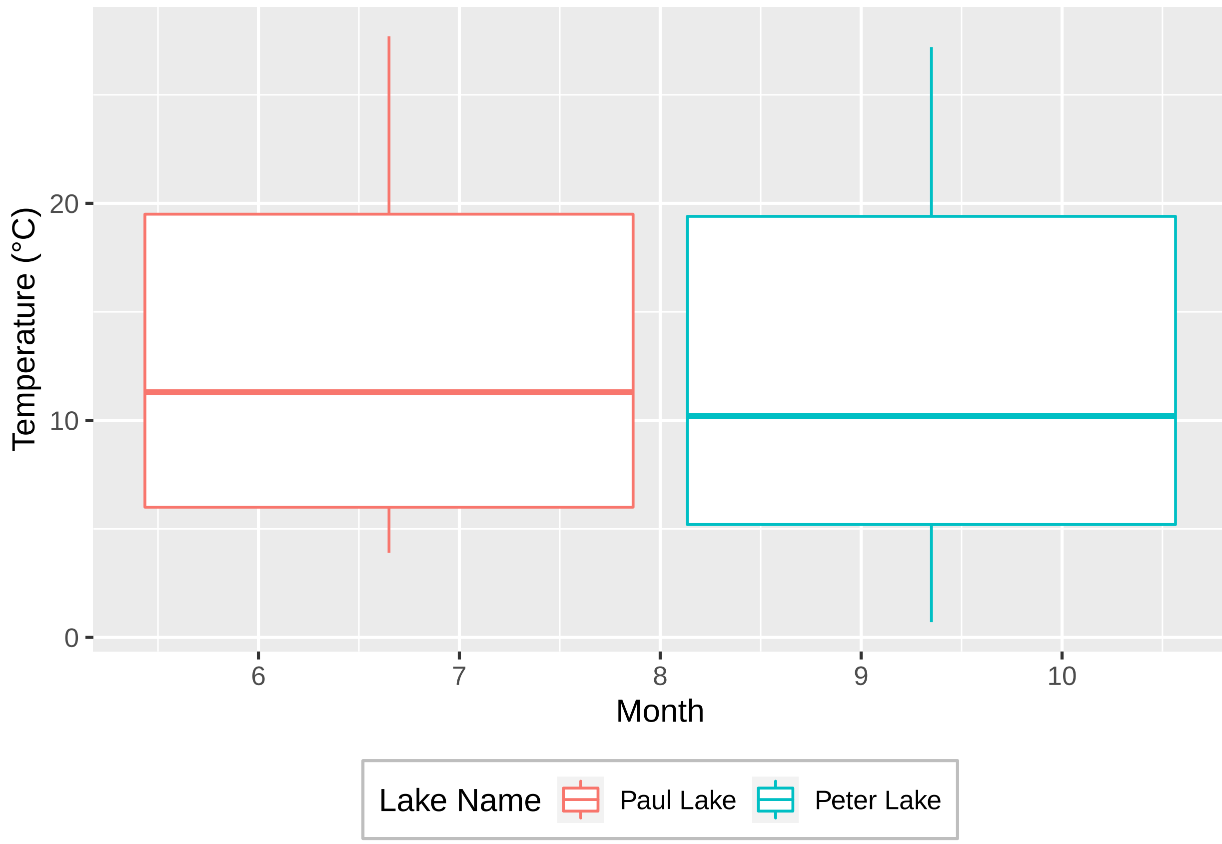
```
## Warning: Removed 21948 rows containing missing values (geom_point).
```



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.

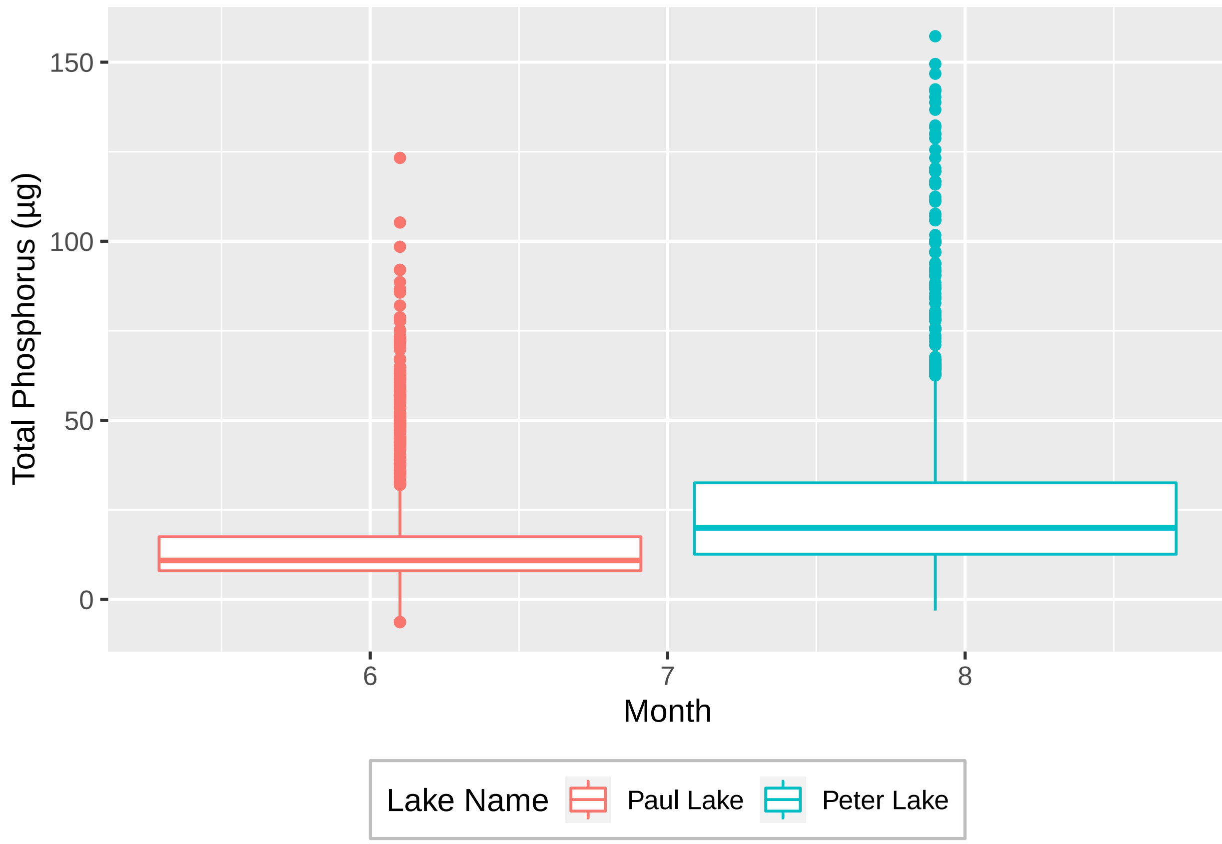
```
#5a

box_temp <- ggplot(chemnutrients_PeterPaul_processed, aes(x = month, y = temperature_C, color = lakenam
  geom_boxplot() +
  labs(color = "Lake Name") +
  xlab("Month") +
  ylab("Temperature (°C)")
print(box_temp)
```

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



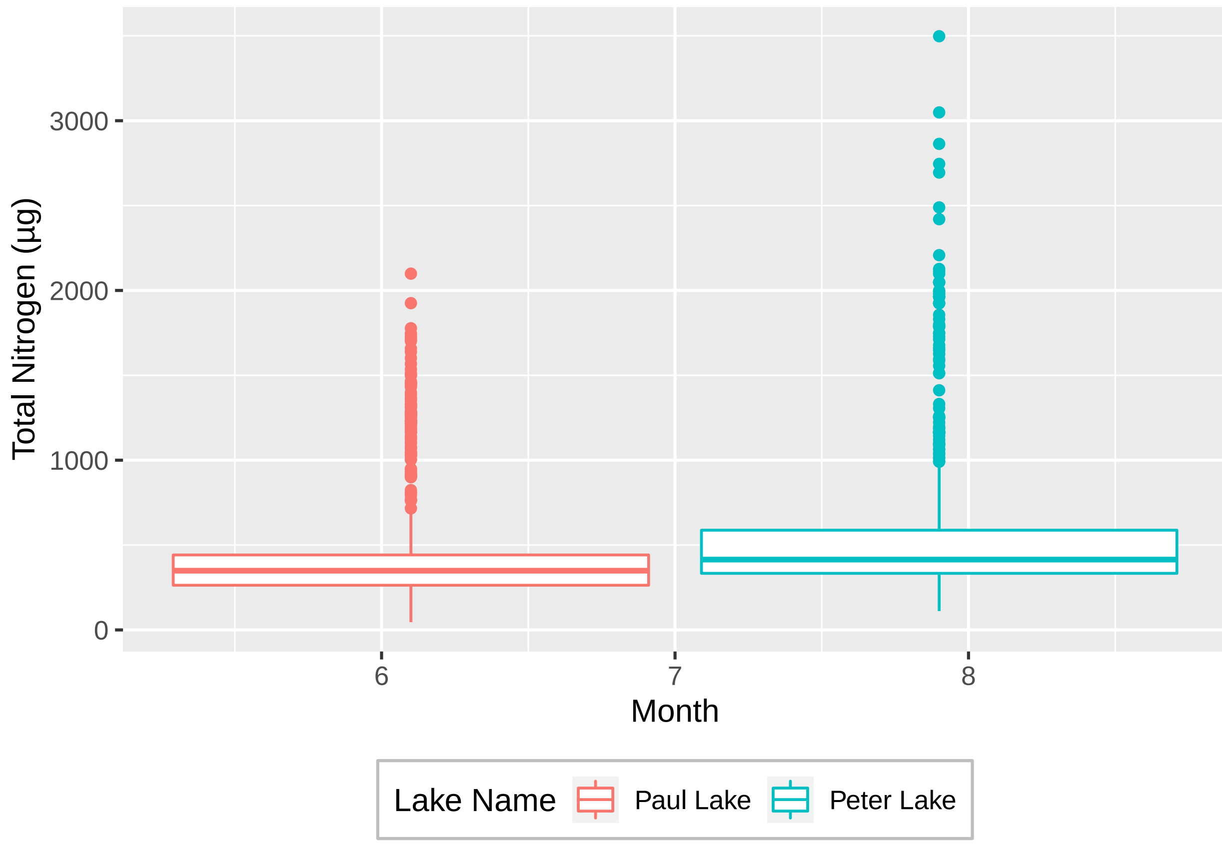

```
#5b

box_tp <- ggplot(chemnutrients_PeterPaul_processed, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(color = "Lake Name") +
  xlab("Month") +
  ylab("Total Phosphorus (µg)")
print(box_tp)
```

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




```
#5c

box_tn <- ggplot(chemnutrients_PeterPaul_processed, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(color = "Lake Name") +
  xlab("Month") +
  ylab("Total Nitrogen (µg)")
print(box_tn)
```

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



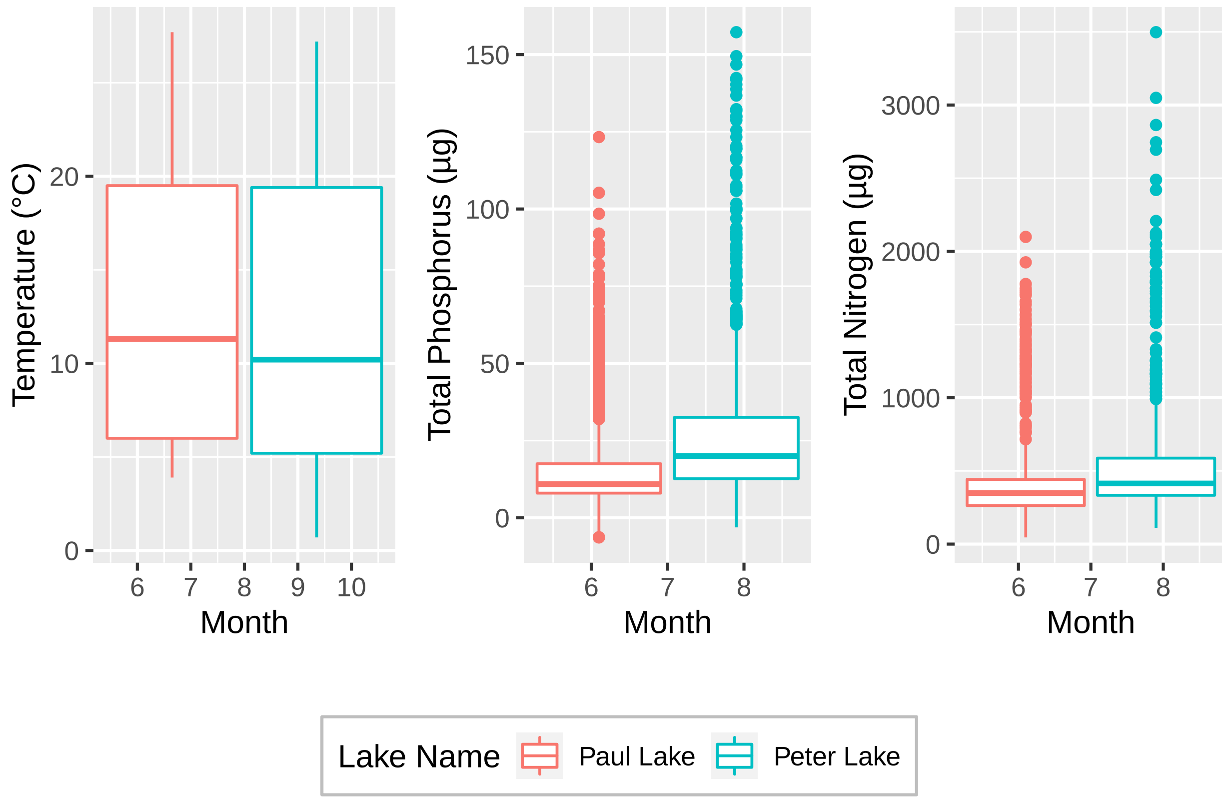

```r
#5d

box_combo_nl <- plot_grid(
  box_temp + theme(legend.position="none"),
  box_tp + theme(legend.position="none"),
  box_tn + theme(legend.position="none"),
  nrow = 1,
  axis = "l",
  rel_heights = c(1.25, 1))
```

```
## 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).
```

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

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

```r
box_combo <- plot_grid(box_combo_nl, legend, ncol = 1, rel_heights = c(1, .3))

print(box_combo)
```



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

> Answer:

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.

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

> Answer: