Assignment 5: Data Visualization

Aurora McCollum

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] "/Users/rorymccollum/Desktop/Rdata/Environmental_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(cowplot)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

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

```
PeterPaul.chem.nutrients <- read.csv(
  "/Users/rorymccollum/Desktop/Rdata/Environmental_Data_Analytics_2022/Data/Processed/NTL-LTER_Lake_Chem
  stringsAsFactors = TRUE)

Niwot.litter <- read.csv(
  "/Users/rorymccollum/Desktop/Rdata/Environmental_Data_Analytics_2022/Data/Processed/NEON_NIWO_Litter_m
  stringsAsFactors = TRUE)

#2

class(PeterPaul.chem.nutrients$sampledate)
```

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

```
PeterPaul.chem.nutrients$sampledate<-ymd(PeterPaul.chem.nutrients$sampledate)
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(Niwot.litter$collectDate)
```

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

```
Niwot.litter$collectDate<-ymd(Niwot.litter$collectDate)
class(Niwot.litter$collectDate)
```

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

Define your theme

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

```
#3

rorystheme<-theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        panel.background = element_rect(fill = "blanchedalmond"),
        legend.position = "right")

theme_set(rorystheme)
```

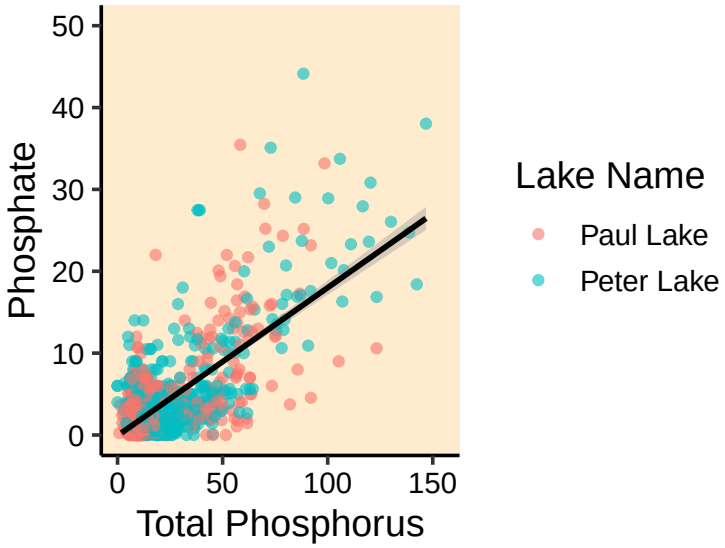
Create graphs

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()`).

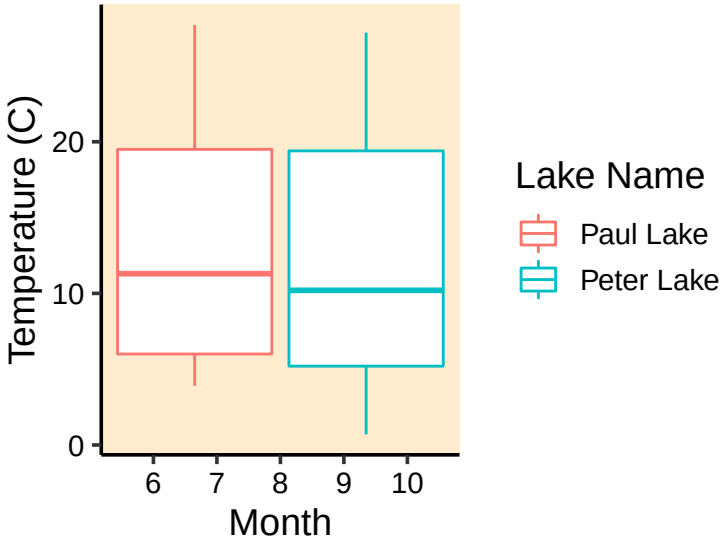
```
#4

phorus_by_phate<-ggplot(PeterPaul.chem.nutrients, aes(x=tp_ug, y=po4, color=lakename))+
   xlim(0,155)+ ylim(0, 50)+
  xlab("Total Phosphorus") + ylab("Phosphate")+
  labs(color="Lake Name")+
  geom_point(alpha=0.6)+
  geom_smooth(method = lm, color="black")

print(phorus_by_phate)
```

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

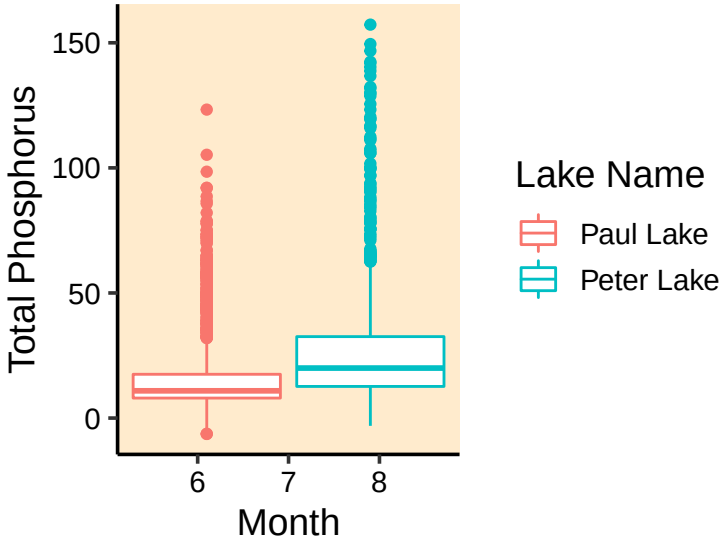


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

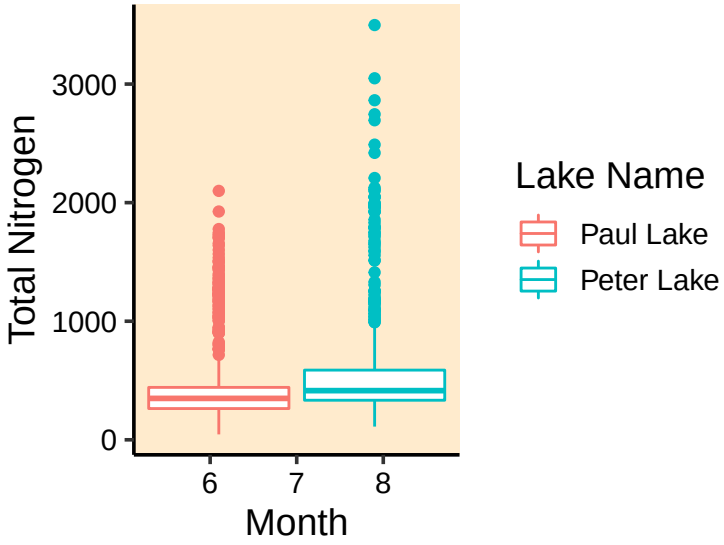
```
#5

#Temperature
temp<-ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=month, y=temperature_C, color=lakename))+
  ylab("Temperature (C)") + xlab("Month")+
  labs(color="Lake Name")

print(temp)
```



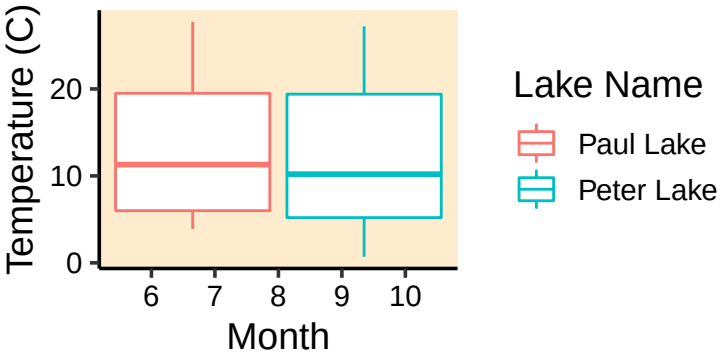

```
#TP
TP<-ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=month, y=tp_ug, color=lakename))+
  ylab("Total Phosphorus") + xlab("Month")+
  labs(color="Lake Name")

print(TP)
```

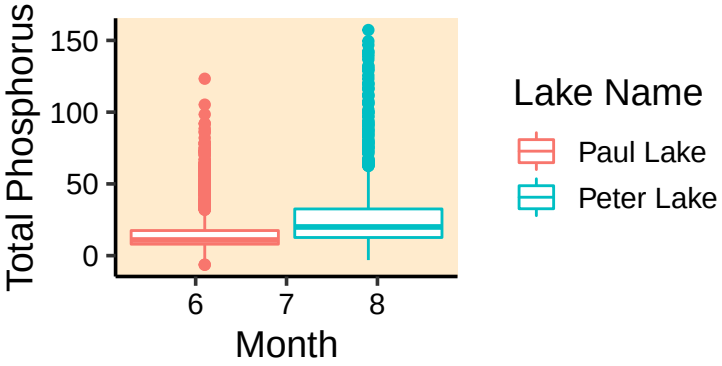


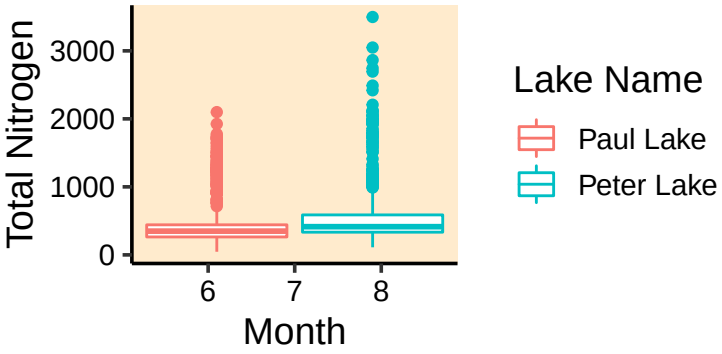

```
#TN
TN<-ggplot(PeterPaul.chem.nutrients)+
  geom_boxplot(aes(x=month, y=tn_ug, color=lakename))+
  ylab("Total Nitrogen") + xlab("Month")+
  labs(color="Lake Name")

print(TN)
```




```
#all together

plot_grid(temp, TP, TN, nrow = 3, align = 'h', rel_heights = c(1, 1, 1))+
  labs(color="Lake Name")
```







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.

```
#6

#drymass_needles<-ggplot(filter(Niwot.litter, functionalGroup==Needles))+
  #geom_point(aes(x=collectDate, y=dryMass, color=nlcdClass))+
  #xlab("Date")+ ylab("Dry Mass")+
  #labs(color="NLCD Class")

#print(drymass_needles)
```

```
#7
```

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

Answer: