Assignment 5: Data Visualization

Samriddha Ghosh

Spring 2023

## OVERVIEW

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

## Directions

1. Rename this file <FirstLast>\_A05\_DataVisualization.Rmd (replacing <FirstLast> with your first and last name).
2. Change “Student Name” on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

## Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. 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 Loading packages  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0  
## ✔ readr 2.1.3 ✔ forcats 1.0.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(lubridate)

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

library(here)

## here() starts at D:/DUKE/EDA-Spring2023

library(cowplot)

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

getwd() #checking directory

## [1] "D:/DUKE/EDA-Spring2023"

Litter<-read.csv("./Data/Processed/NEON\_NIWO\_Litter\_mass\_trap\_Processed.csv", stringsAsFactors = TRUE)  
Nutrient<-read.csv("./Data/Processed/NTL-LTER\_Lake\_Chemistry\_Nutrients\_PeterPaul\_Processed.csv", stringsAsFactors = TRUE)  
  
#2 Changing the format to date  
Litter$collectDate<-ymd(Litter$collectDate)  
Nutrient$sampledate<-ymd(Nutrient$sampledate)  
Nutrient$month<-month(Nutrient$month, label=TRUE)

## Define your theme

1. Build a theme and set it as your default theme. Customize the look of at least two of the following:

* Plot background
* Plot title
* Axis labels
* Axis ticks/gridlines
* Legend

#3 Defining the plot by:  
#a)changing the plot background  
#b)Setting the position and alignment of the plot legend  
#c)Changing the color of the axis texts to black  
mytheme<-theme\_bw(base\_size = 12) +  
 theme(axis.text = element\_text(color="black"),  
 legend.position = "top",   
 legend.justification = "centre")  
  
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.

1. [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/or ylim()).

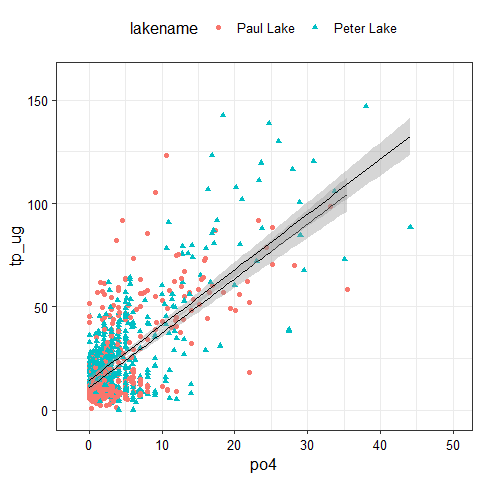
#4   
  
ggplot(Nutrient, aes(x= po4, y= tp\_ug, color=lakename, shape=lakename))+  
 geom\_point()+  
 xlim(-2,50)+  
 ylim(-2,160)+  
 geom\_smooth(method=lm, color="black", size=0.5)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 21948 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 21948 rows containing missing values (`geom\_point()`).

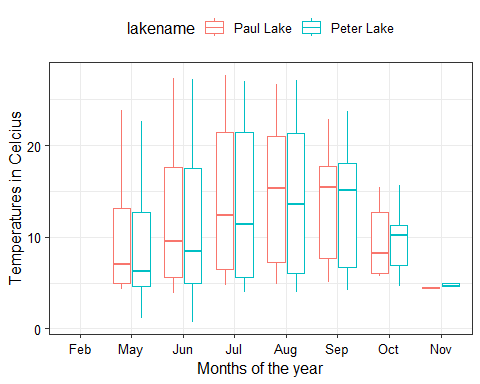


1. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: R has a build in variable called month.abb that returns a list of months;see <https://r-lang.com/month-abb-in-r-with-example>

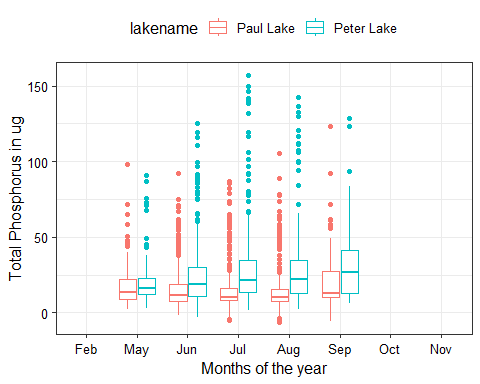
#5   
  
temperature<-ggplot(Nutrient, aes(x=month, y=temperature\_C))+  
 geom\_boxplot(aes(color = lakename))+  
 xlab("Months of the year")+  
 ylab("Temperatures in Celcius")  
print(temperature)

## Warning: Removed 3566 rows containing non-finite values (`stat\_boxplot()`).



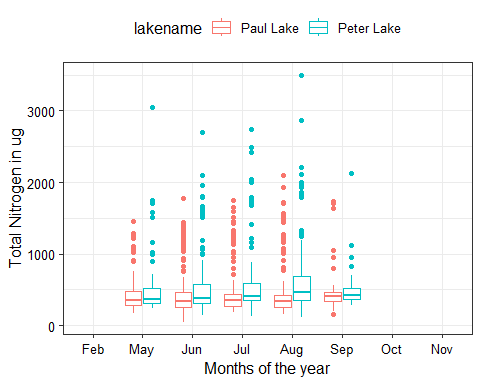
TP<-ggplot(Nutrient, aes(x=month, y=tp\_ug, color=lakename))+  
 geom\_boxplot()+  
 xlab("Months of the year")+  
 ylab("Total Phosphorus in ug")  
  
print(TP)

## Warning: Removed 20729 rows containing non-finite values (`stat\_boxplot()`).



TN<-ggplot(Nutrient, aes(x=month, y=tn\_ug, color=lakename))+  
 geom\_boxplot()+  
 xlab("Months of the year")+  
 ylab("Total Nitrogen in ug")  
print(TN)

## Warning: Removed 21583 rows containing non-finite values (`stat\_boxplot()`).



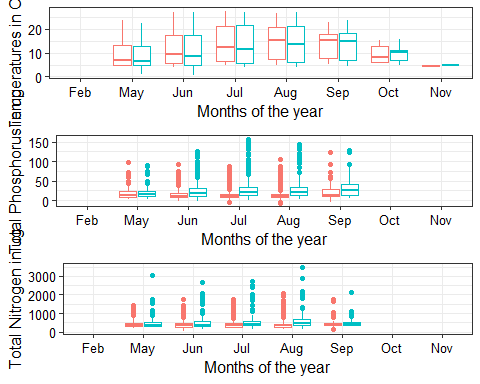
Legend\_less<-plot\_grid(temperature + theme(legend.position = "none"),  
 TP + theme(legend.position = "none"),  
 TN + theme(legend.position = "none"),  
 nrow=3, align='h')

## 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()`).

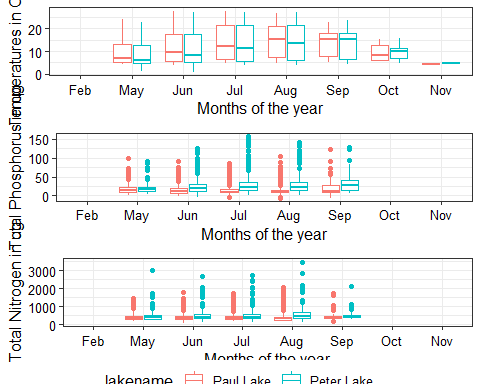
print(Legend\_less)



Legend<-get\_legend(  
 temperature)

## Warning: Removed 3566 rows containing non-finite values (`stat\_boxplot()`).

print(plot\_grid(Legend\_less, Legend, nrow=2, rel\_heights = c(5,0.1)))

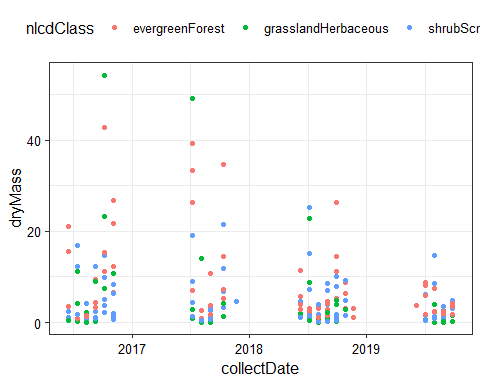


Question: What do you observe about the variables of interest over seasons and between lakes?

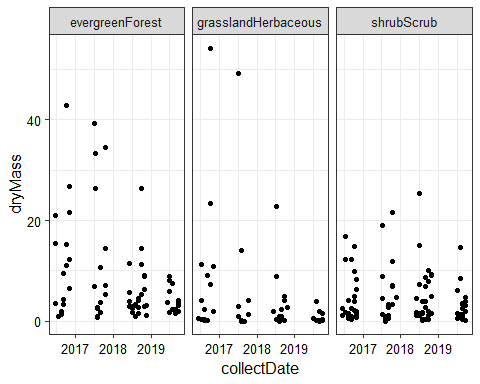
Answer: In case of plot 1 for temperature against the months of the year: The mean temperature of Paul lake is higher than Peter lake up until September after which the mean temperature of Peter lake increases upto November. In case of plot 2 for total phosphorus in ug vs months of the year: The mean of the total phosphorus in ug for Peter lake keeps increasing with each passing month. In case of plot 2 for total nitrogen in ug vs months of the year: The mean of the total nitrogen in ug for Peter lake keeps increasing with each passing month.

1. [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)
2. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

#6  
Litter %>%  
 filter(functionalGroup == "Needles") %>%  
 ggplot(aes(y = dryMass, x = collectDate, color = nlcdClass)) +  
 geom\_point()



#7  
  
Litter %>%  
 filter(functionalGroup == "Needles") %>%  
 ggplot(aes(y = dryMass, x = collectDate)) +  
 geom\_point()+  
 facet\_wrap(vars(nlcdClass),nrow=1)

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

Answer: The graph plotted in Q.7 is more effective. This is because now for each nlcdClass, we are automatically getting a separate plot for drymass against a date. This graph makes the data visualization and comparison easier and better.