Assignment 5: Data Visualization

Shivani Kuckreja

#install.packages(“formatR”)

## 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] "/home/guest/R/Environmental\_Data\_Analytics\_2022"

library(tidyverse)

## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'  
## had status 1

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.6 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.4 ✓ stringr 1.4.0  
## ✓ readr 2.1.1 ✓ 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("./Data/Processed/NTL-LTER\_Lake\_Chemistry\_Nutrients\_PeterPaul\_Processed.csv")  
   
NW\_litter <-  
 read.csv("./Data/Processed/NEON\_NIWO\_Litter\_mass\_trap\_Processed.csv")  
  
#2   
class(PeterPaul\_chem\_nutrients$sampledate) #originally a character

## [1] "character"

# Change date to date object  
PeterPaul\_chem\_nutrients$sampledate <- ymd(PeterPaul\_chem\_nutrients$sampledate)  
  
class(NW\_litter$collectDate) #originally a character

## [1] "character"

# Change date to date object  
NW\_litter$collectDate <- ymd(NW\_litter$collectDate)

## Define your theme

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

#3  
  
mytheme <- theme\_classic(base\_size = 14) +  
 theme(axis.text = element\_text(color = "blue"),   
legend.position = "top") #alternative: legend.position + legend.justification  
  
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.

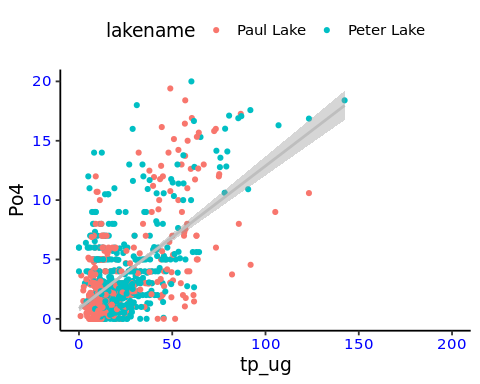
1. [NTL-LTER] Plot total phosphorus (tp\_ug) by phosphate (po4), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using xlim() and ylim()).

#4  
NTLLTER\_Plot <- ggplot(PeterPaul\_chem\_nutrients, aes(x = tp\_ug, y = po4,   
 color=lakename)) +  
 geom\_point() +  
 geom\_smooth(method = lm, color = "grey") +  
 ylab('Po4') +  
 xlim(0, 200) +  
 ylim(0, 20)   
print(NTLLTER\_Plot)

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

## Warning: Removed 21981 rows containing non-finite values (stat\_smooth).

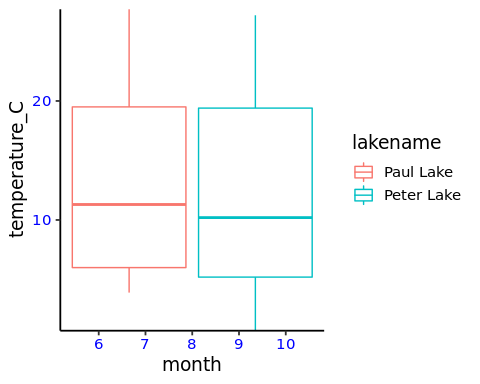
## Warning: Removed 21981 rows containing missing values (geom\_point).



1. [NTL-LTER] Make three separate boxplots of (a) temperature,
2. 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.

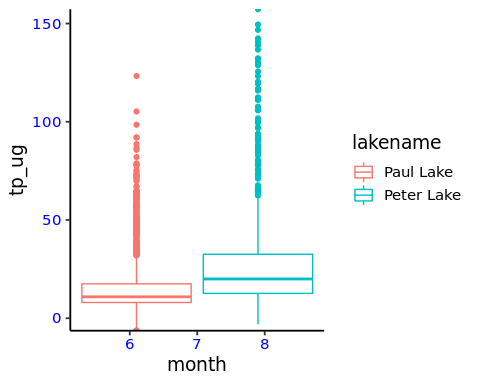
#5  
PeterPaul\_chem\_nutrients$month <- as.numeric(PeterPaul\_chem\_nutrients$month)  
  
NTLLTER\_BoxPlot\_Temp <- ggplot(PeterPaul\_chem\_nutrients,   
 aes(x = month, y = temperature\_C,   
 color=lakename)) +  
 geom\_boxplot() +  
 scale\_y\_continuous(expand = c(0, 0)) +  
 theme(legend.position = "right")  
print(NTLLTER\_BoxPlot\_Temp)

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



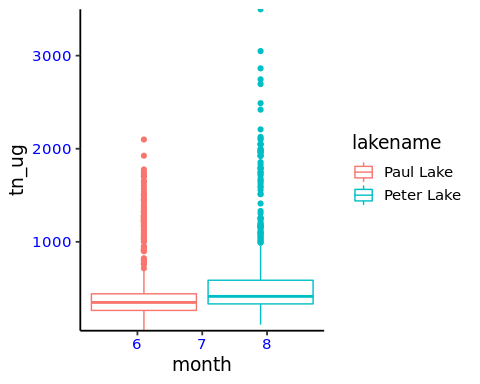
NTLLTER\_BoxPlot\_TP <- ggplot(PeterPaul\_chem\_nutrients, aes(x = month,   
 y = tp\_ug,   
 color=lakename)) +  
 geom\_boxplot() +  
 scale\_y\_continuous(expand = c(0, 0)) +  
 theme(legend.position = "right")  
print(NTLLTER\_BoxPlot\_TP)

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



NTLLTER\_BoxPlot\_TN <- ggplot(PeterPaul\_chem\_nutrients,   
 aes(x = month, y = tn\_ug, color=lakename)) +  
 geom\_boxplot() +  
 scale\_y\_continuous(expand = c(0, 0)) +  
 theme(legend.position = "right")  
print(NTLLTER\_BoxPlot\_TN)

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

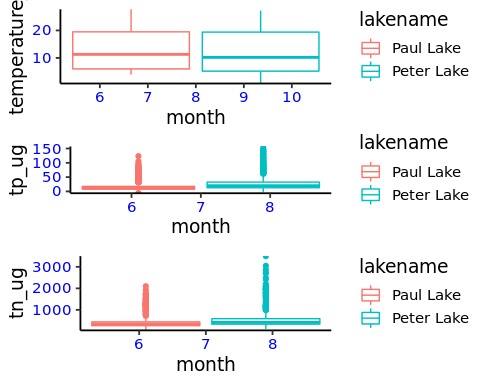


#install.packages("cowplot")   
library(cowplot)  
plot\_grid(NTLLTER\_BoxPlot\_Temp, NTLLTER\_BoxPlot\_TP,   
 NTLLTER\_BoxPlot\_TN, nrow = 3, align = 'h', rel\_heights = c(1.25, 1))

## 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).

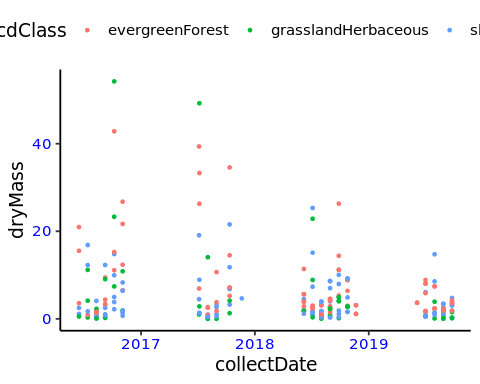


Question: What do you observe about the variables of interest over seasons and between lakes?

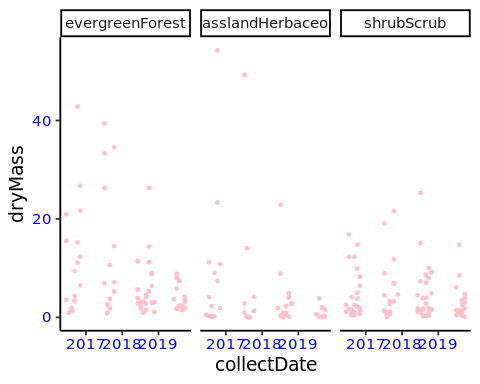
Answer: tp\_ug: There are a lot of outliers in this dataset, especially compared to the Temperature plots. There is low variability in the data for Paul Lake. tn\_ug: There are a lot of outliers in this dataset, and Peter Lake seems to have significantly more outliers (of higher value) than does Paul Lake. temperature: Temperature over seasons and between lakes appears to vary very little.

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  
NW\_litter\_filter <- NW\_litter %>%   
 filter(functionalGroup == "Needles")   
  
ggplot(filter(NW\_litter\_filter)) +  
 geom\_point(aes(x = collectDate, y = dryMass,color=nlcdClass), size=1)



#7  
NW\_litter\_filter <- NW\_litter %>%   
 filter(functionalGroup == "Needles")   
  
ggplot(filter(NW\_litter\_filter)) +  
 geom\_point(aes(x = collectDate, y = dryMass), color="pink", size=1) +  
facet\_wrap('nlcdClass')

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

Answer: Plot 6 is more effective because it allows easier comparison across NLCD classes. Plot 7 only allows easier comparison by year within each NLCD class.