Assignment 5: Data Visualization

Asreeta Ushasri

Fall 2024

## 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. Read in 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 in the Processed\_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the NEON\_NIWO\_Litter\_mass\_trap\_Processed.csv version, again from the Processed\_KEY folder).
2. Make sure R is reading dates as date format; if not change the format to date.

#1 Install packages and read in data  
  
#install.packages(tidyverse)  
#install.packages(cowplot)  
#install.packages(lubridate)  
#install.packages(here)  
#install.packages(ggridges)  
  
library(ggridges)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(here)

## here() starts at /home/guest/AU\_EDE\_Fall2024

getwd()

## [1] "/home/guest/AU\_EDE\_Fall2024"

here()

## [1] "/home/guest/AU\_EDE\_Fall2024"

PeterPaul <- read.csv(  
 file =  
 here("Data/Processed/NTL\_LTER\_Lake\_Chemistry\_Nutrients\_PeterPaul\_Processed.csv"),  
 stringsAsFactors = TRUE)  
  
Litter <- read.csv(  
 file =  
 here("Data/Processed/NEON\_NIWO\_Litter\_mass\_trap\_Processed.csv"),  
 stringsAsFactors = TRUE)  
  
#view(PeterPaul)  
#view(Litter)  
  
#2 Check the date format  
class(PeterPaul$sampledate)

## [1] "factor"

class(Litter$sampledate)

## [1] "NULL"

# Format sampledate as date  
PeterPaul$sampledate <-  
 as.Date(PeterPaul$sampledate,  
 format = '%Y-%m-%d')  
  
Litter$collectDate <-  
 as.Date(Litter$collectDate,  
 format = '%Y-%m-%d')  
  
class(PeterPaul$sampledate)

## [1] "Date"

class(Litter$collectDate)

## [1] "Date"

## 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 Creating a plot theme  
  
#install.packages(viridisLite)  
  
library(viridis)

## Loading required package: viridisLite

library(RColorBrewer)  
library(colormap)  
library(ggthemes)  
  
  
#Create a custom theme, using elements from lab document  
  
my\_theme <- theme\_tufte() +  
theme(  
 line = element\_line(),  
 rect = element\_rect(),  
 text = element\_text(),  
   
#Text Element  
 plot.title = element\_text("black"),  
 axis.title.x = element\_text("black"),  
 axis.title.y = element\_text("black"),  
 axis.text = element\_text("black"),  
  
#Line Element  
 axis.ticks = element\_line("lightgray"),  
 panel.grid.major = element\_line("lightgray"),  
 panel.grid.minor = element\_line("lightgray"),  
  
#Rectangle Element  
 plot.background = element\_rect("lightblue"),  
 panel.background = element\_rect("white"),  
 legend.key = element\_rect("white"),  
  
#Legend Position  
 legend.position = 'right',  
complete = TRUE)  
  
  
#Test My Default theme  
#PeterPaul %>%   
 #ggplot() +   
 #geom\_bar(aes(x=month)) + my\_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.

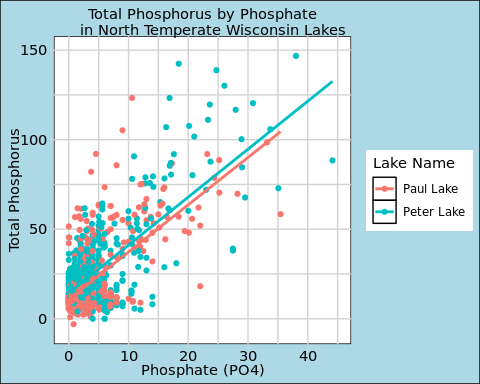
1. [NTL-LTER] Plot total phosphorus (tp\_ug) by phosphate (po4), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the lm method. Adjust your axes to hide extreme values (hint: change the limits using xlim() and/or ylim()).

#4 Plot total phosphorus by phosphate in Peter and Paul Lakes.  
  
#First, set up basic line plot  
PeterPaulPlot <- PeterPaul %>%  
 ggplot(mapping =  
 aes(x=po4,  
 y=tp\_ug,  
 color=lakename)) + geom\_point()  
  
#show(PeterPaulPlot)  
  
#Change scale of x and y axis  
PeterPaulPlotLimits <- PeterPaulPlot + xlim(NA, 45) + ylim(NA, 150)  
  
#show(PeterPaulPlotLimits)  
  
#Add axis titles  
PeterPaulPlotTitles <- PeterPaulPlotLimits +  
 labs(  
 title = "Total Phosphorus by Phosphate  
 in North Temperate Wisconsin Lakes",  
 x = "Phosphate (PO4)",  
 y = "Total Phosphorus",  
 color = "Lake Name") +  
 theme(axis.title.y =  
 element\_text(angle = 90,  
 vjust = 0.5, hjust=1))   
  
#show(PeterPaulPlotTitles)  
  
  
#Add line of best fit and custom theme  
  
Peter\_Paul\_Phos\_Graph <- PeterPaulPlotTitles +  
 geom\_smooth(method = lm,  
 se=FALSE) + my\_theme +  
 theme(axis.title.y = element\_text(angle = 90))  
  
show(Peter\_Paul\_Phos\_Graph)

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

## Warning: Removed 21947 rows containing non-finite outside the scale range  
## (`stat\_smooth()`).

## Warning: Removed 21947 rows containing missing values or values outside the scale range  
## (`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.

Tips: \* Recall the discussion on factors in the lab section as it may be helpful here. \* Setting an axis title in your theme to element\_blank() removes the axis title (useful when multiple, aligned plots use the same axis values) \* Setting a legend’s position to “none” will remove the legend from a plot. \* Individual plots can have different sizes when combined using cowplot.

#5 Create Boxplots  
  
#First, format months as factors  
  
PeterPaul$month <- factor(  
 PeterPaul$month)  
  
#Check levels of months  
  
class(PeterPaul$month)

## [1] "factor"

unique(PeterPaul$month)

## [1] 5 6 7 8 9 10 11 2   
## Levels: 2 5 6 7 8 9 10 11

#Add missing months  
  
levels(PeterPaul$month) <-  
 c(levels(PeterPaul$month),  
 "1", "3", "4", "12")  
  
levels(PeterPaul$month)

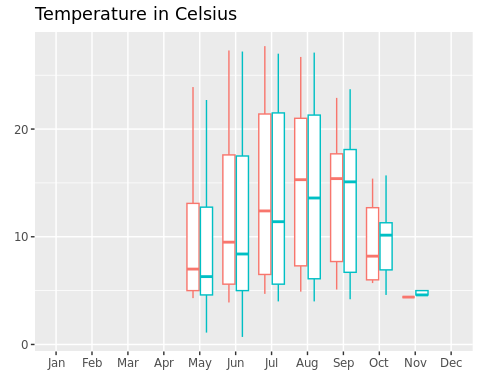
## [1] "2" "5" "6" "7" "8" "9" "10" "11" "1" "3" "4" "12"

#Reorder months in sequence  
  
PeterPaul$month <- factor(  
 PeterPaul$month,  
 levels = c("1", "2", "3", "4",  
 "5", "6", "7", "8",  
 "9", "10", "11", "12"))  
  
levels(PeterPaul$month)

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

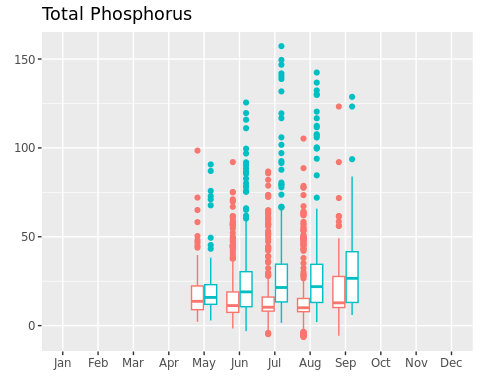
#Add month abbreviation names  
  
PeterPaul$month <- factor(  
 PeterPaul$month,  
 levels = c("1", "2", "3",  
 "4", "5", "6",  
 "7", "8", "9",  
 "10", "11", "12"),  
 labels = c("Jan", "Feb", "Mar", "Apr",  
 "May", "Jun", "Jul", "Aug",  
 "Sep", "Oct", "Nov", "Dec"))  
  
#Create temperature boxplot  
  
Peter\_Paul\_Temp <- ggplot(data = PeterPaul,  
 aes(  
 x=month,  
 y=temperature\_C,  
 color=lakename)) +  
 geom\_boxplot() +  
 scale\_x\_discrete(drop=FALSE) +  
 labs(  
 title = "Temperature in Celsius",  
 x = element\_blank(),  
 y = element\_blank()) +  
 theme(legend.position = "none")  
  
show(Peter\_Paul\_Temp)

## Warning: Removed 3566 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



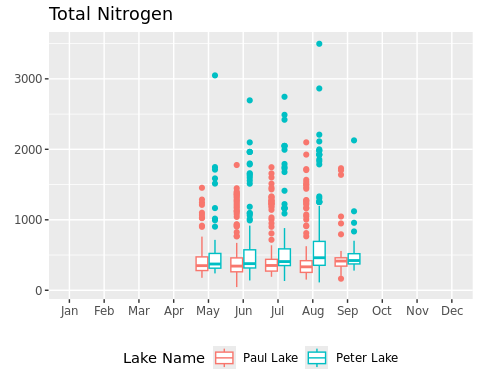
#Next, create phosphorus boxplot  
  
Peter\_Paul\_TP <- ggplot(data = PeterPaul,  
 aes(  
 x=month,  
 y=tp\_ug,  
 color=lakename)) +  
 geom\_boxplot() +  
 scale\_x\_discrete(drop=FALSE) +  
 labs(  
 title = "Total Phosphorus",  
 x = element\_blank(),  
 y = element\_blank()) +  
 theme(legend.position = "none")  
  
show(Peter\_Paul\_TP)

## Warning: Removed 20729 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



#Lastly, create nitrogen boxplot  
  
Peter\_Paul\_TN <- ggplot(data = PeterPaul,  
 aes(  
 x=month,  
 y=tn\_ug,  
 color=lakename)) +  
 geom\_boxplot() +  
 labs(  
 title = "Total Nitrogen",  
 x = element\_blank(),  
 y = element\_blank(),  
 color = "Lake Name") +   
 scale\_x\_discrete(drop=FALSE) +  
 theme(legend.position = "bottom")  
  
show(Peter\_Paul\_TN)

## Warning: Removed 21583 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



#Combine multiple plots  
  
library(cowplot)

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':  
##   
## theme\_map

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

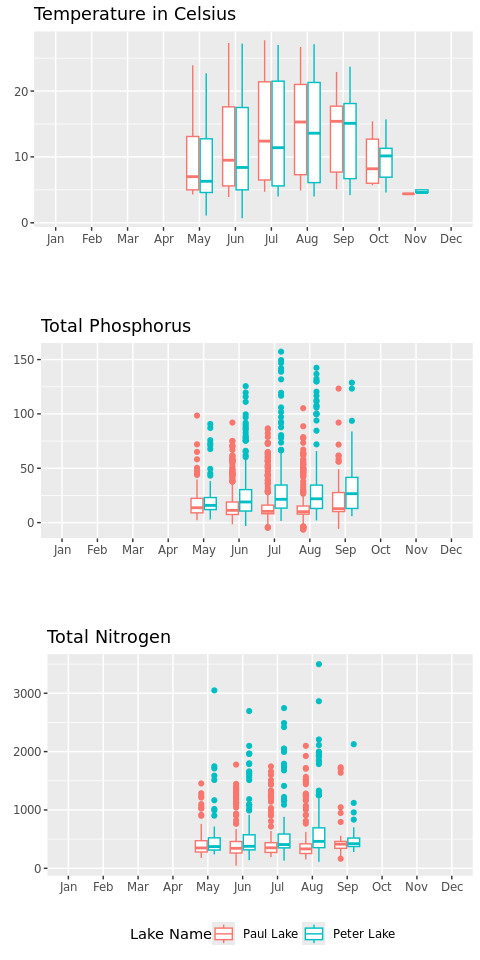
Boxplots <-  
 plot\_grid(Peter\_Paul\_Temp,  
 Peter\_Paul\_TP,  
 Peter\_Paul\_TN,  
 nrow = 3,  
 align = 'h',  
 axis = 'tb',  
 rel\_heights = c(3, 3, 3.25))

## Warning: Removed 3566 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 20729 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 21583 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

show(Boxplots)



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

Answer: The temperature for both Peter and Paul Lakes increases over the summer months, with a higher median in July and August. The phosphorus levels in Peter Lake increases over July and August as well, whereas the median phosphorus level remains at a similar level across seasons in Paul Lake. Aside from a slight increase in total nitrogen in Peter Lake during August, total nitrogen appears relatively the same across seasons in these lakes.

Between the two lakes, Peter Lake has much higher median phosphorus than Paul Lake across all months. Peter Lake also has a slighter higher median nitrogen across most of the months. There is only a slight difference in nitrogen levels between the lakes, but there is a wide difference in phosphorus levels between the lakes (with Peter Lake having more phosphorus than Paul Lake).

Lastly, the phosophorus and nitrogen graphs appear to contain a variety of outliers at the top of the boxplots. This indicates that the distribution of phosphorus and nitrogen for both Peter Lake and Paul Lake is skewed right.

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.

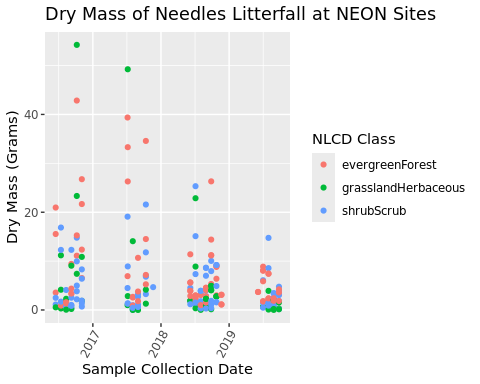
#6 Plotting Needles Group Dry Mass  
  
#First, create new dataset filtered by Needles  
Litter\_Needles <- Litter %>%   
 filter(functionalGroup == "Needles")  
  
#view(Litter\_Needles)  
  
#Check date format  
class(Litter\_Needles$collectDate)

## [1] "Date"

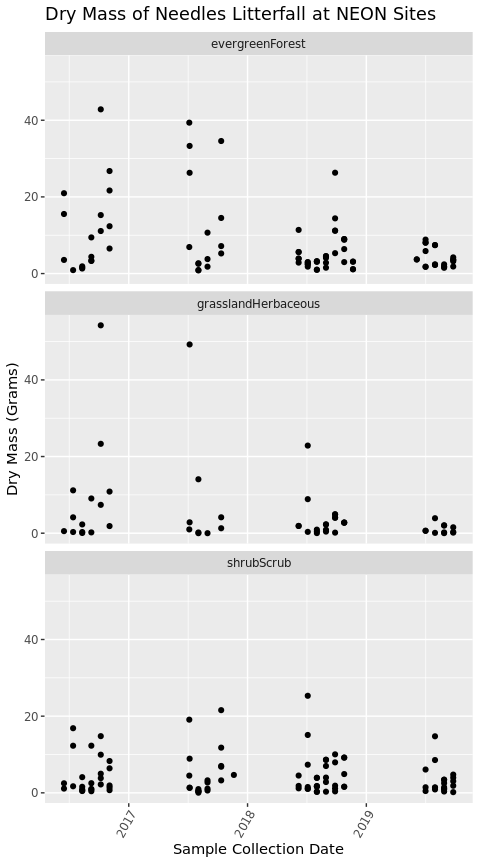
sum(is.na(Litter\_Needles))

## [1] 0

#Now, create plot from Litter\_Needles  
ggplot(data = Litter\_Needles,  
 aes(x = collectDate, y = dryMass,  
 color = nlcdClass)) +  
 geom\_point() +  
 labs(  
 title = "Dry Mass of Needles Litterfall at NEON Sites",  
 x = "Sample Collection Date",  
 y = "Dry Mass (Grams)",  
 color = "NLCD Class") +  
 theme(axis.text.x = element\_text(angle = 60, vjust = 1, hjust=1))



#7 Changing NLCD Class from color to facet  
  
#Use the Litter\_Needles dataset  
ggplot(data = Litter\_Needles,  
 aes(x = collectDate, y = dryMass)) +  
 geom\_point() + facet\_wrap("nlcdClass", nrow = 3) +  
 labs(  
 title = "Dry Mass of Needles Litterfall at NEON Sites",  
 x = "Sample Collection Date",  
 y = "Dry Mass (Grams)") +  
 theme(axis.text.x =  
 element\_text(angle = 60,  
 vjust = 1, hjust=1))

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

Answer: Plot 6 is effective in comparing between the different NLCD classes, because Plot 6 displays the different NLCD classes on the same graph. This allows researchers to compare and contrast the dry mass values of evergreen forest samples, grassland herbaceous samples, and shrub scrub samples easily with color differentiation. In Plot 6, we can see that the evergreen forest and grassland herbaceous samples had higher dry mass values from October 2016 to October 2017. The Shrub Scrub dry mass remained lower than the other two NLCD groups throughout this time frame. Plot 7 does not include any color differentiation, which makes it difficult to determine which scatterplot points represent the three different NLCD classes.

However, Plot 7 is effective in comparing between years. By separating out the different NLCD classes into three rows, Plot 7 clearly shows differences in the litterfall mass over time. 2017 had a higher dry mass in evergreen forest and grassland samples, where as 2019 had a lower dry mass in both of these NLCD classes.