Assignment 5: Data Visualization

Mara (Margaret) Michel

Fall 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. 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 Load in packages
library(cowplot);library(grid);
library(tidyverse);library(lubridate);library(here);library(ggthemes)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
```

```
## x lubridate::stamp() masks cowplot::stamp()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## here() starts at C:/Users/marga/OneDrive/Documents/EDE_Fall2023
##
##
## Attaching package: 'ggthemes'
##
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
#Verify home directory
print(R.home())
```

```
## [1] "C:/PROGRA~1/R/R-43~1.1"
```

```
#Read in data files
PeterPaul.chem.nutrients <- read.csv(
file=here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

litter <- read.csv(
  file=here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2 Update Date Format
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
class(PeterPaul.chem.nutrients$sampledate)
```

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

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

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

Define your theme

3. 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 Customizing theme
my_theme <- theme_base()  +
  theme(plot.background = element_rect(fill='azure'),
```

```
      plot.title= element_text(
                   size = 20,
                   face = 'bold',
                   hjust = .5),
      legend.title = element_text(face = 'bold'),
      legend.background = element_rect(fill = 'azure2'),legend.key=element_rect(fill='azure2')
  )

#Set new theme as default
theme_set(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.

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/or `ylim()`).

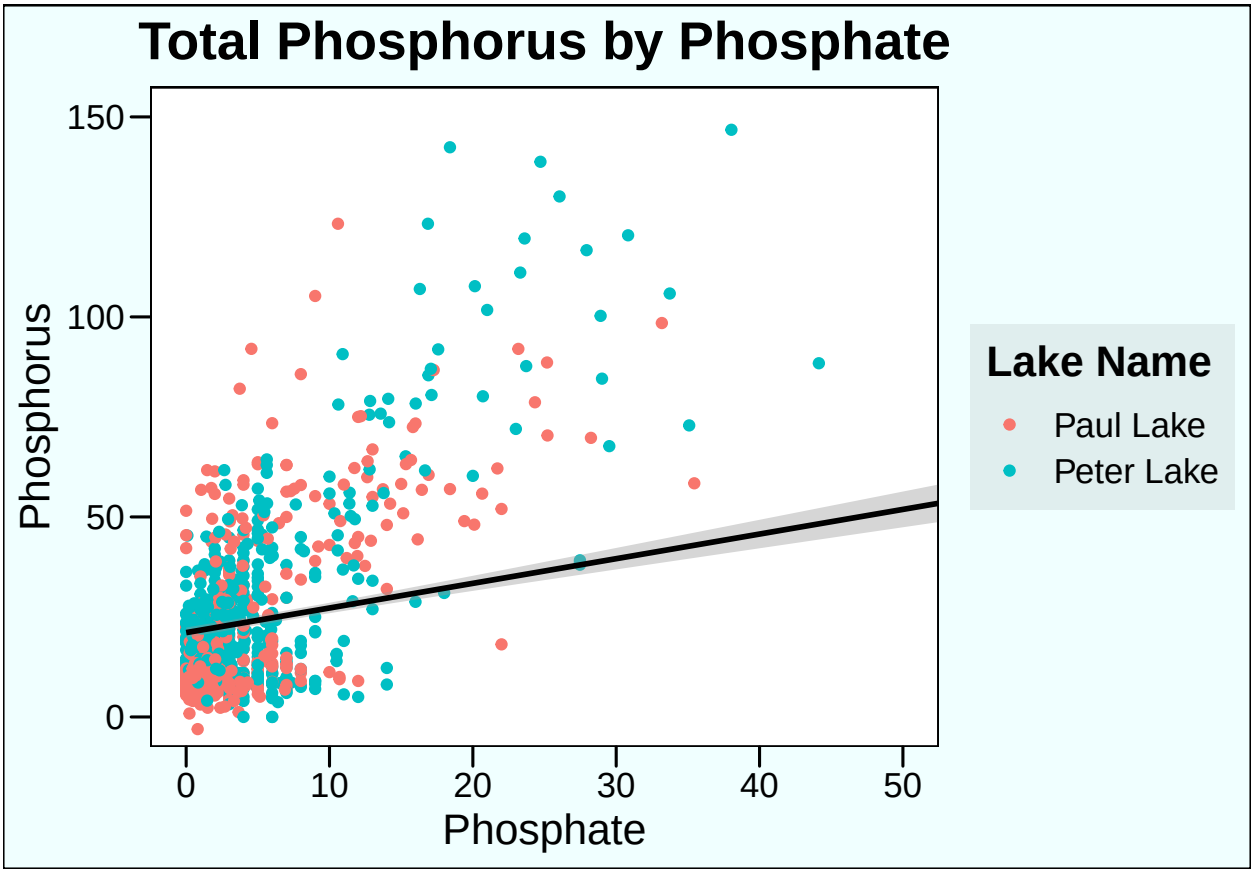
```
#4 Create plot titled "phos_plot"
phos_plot<-
  ggplot(PeterPaul.chem.nutrients, aes(x=po4,
      y=tp_ug,
      color=lakename))+
  geom_point()+
  geom_smooth(method="lm",color='black') +
  coord_cartesian(xlim=c(0,50),ylim=c(0,150))+
  ylab("Phosphorus")+
  xlab("Phosphate")+
  labs(color="Lake Name")+
  ggtitle("Total Phosphorus by Phosphate")

phos_plot
```

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

## Warning: Removed 21946 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21946 rows containing missing values (`geom_point()`).
```



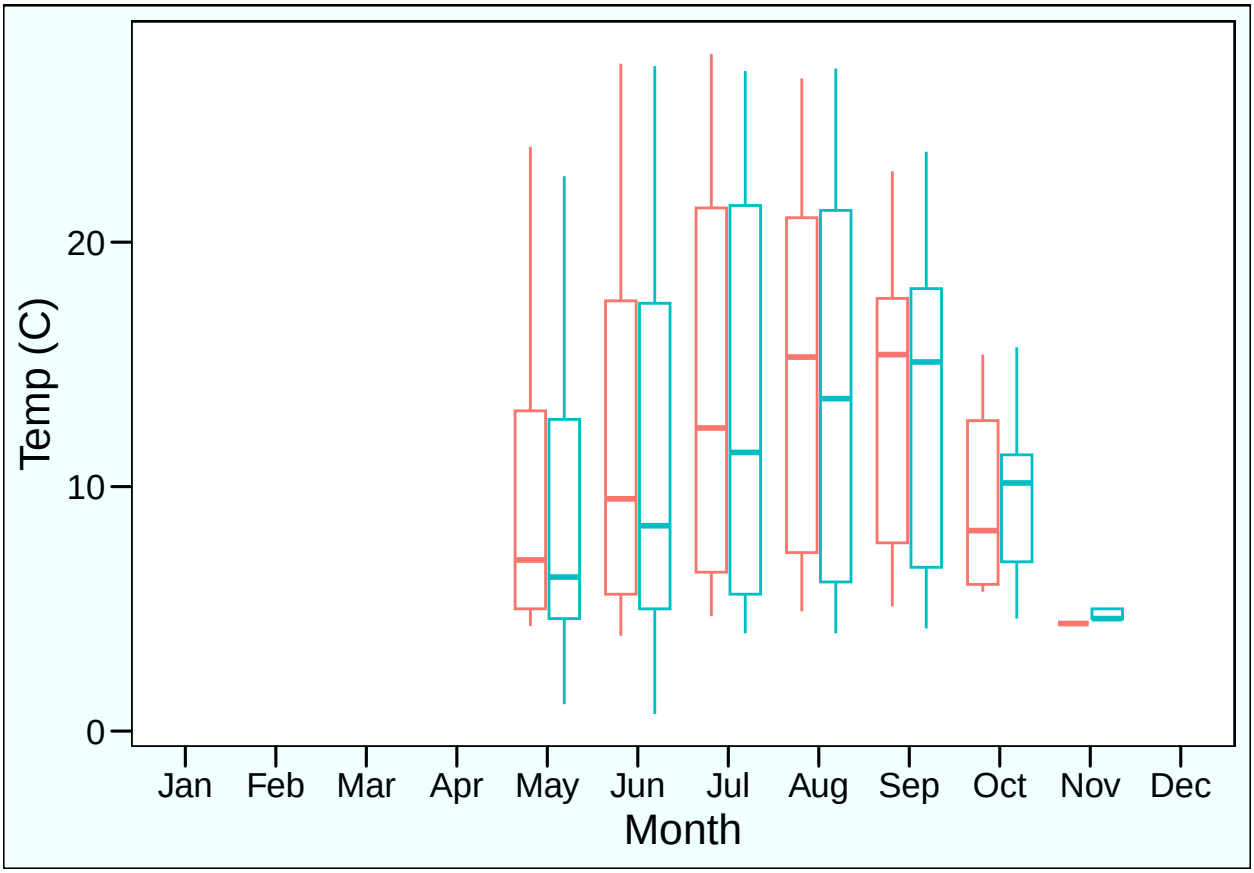
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
#Boxplot A: Temperature
temperature_plot<-
  ggplot(PeterPaul.chem.nutrients,
         aes(x=factor(month,levels=1:12,labels=month.abb),
             y=temperature_C,
             color=lakename))+
  geom_boxplot()+
  scale_x_discrete(name="Month",
                   drop=FALSE)+
  ylab("Temp (C)")+
  theme(legend.position = "none")

temperature_plot
```

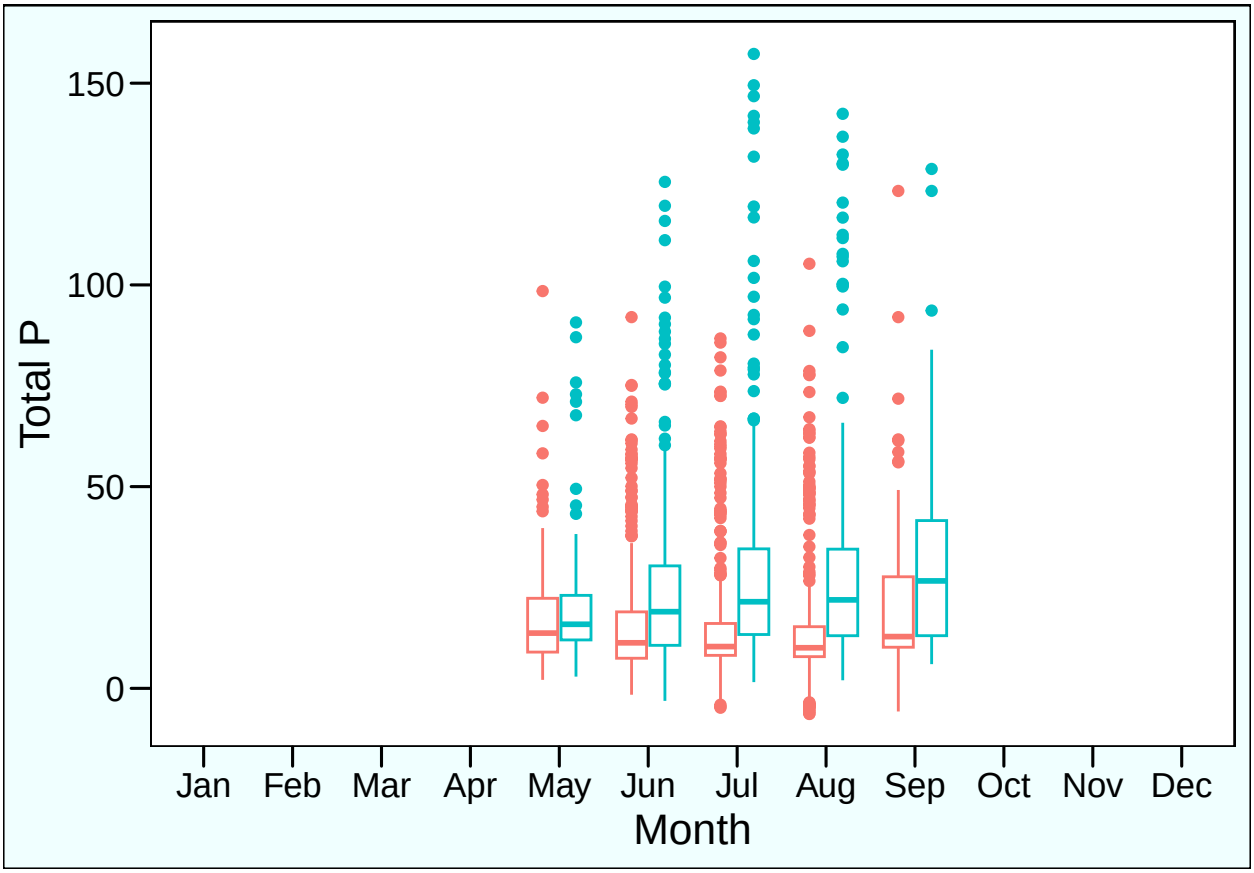
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#Boxplot B: TP
tp_plot<-
  ggplot(PeterPaul.chem.nutrients,
         aes(x=factor(month,levels=1:12,labels=month.abb),
             y=tp_ug,
             color=lakename))+
 geom_boxplot()+
 scale_x_discrete(name="Month",
                  drop=FALSE)+
  ylab("Total P")+
  theme(legend.position = "none")

tp_plot
```

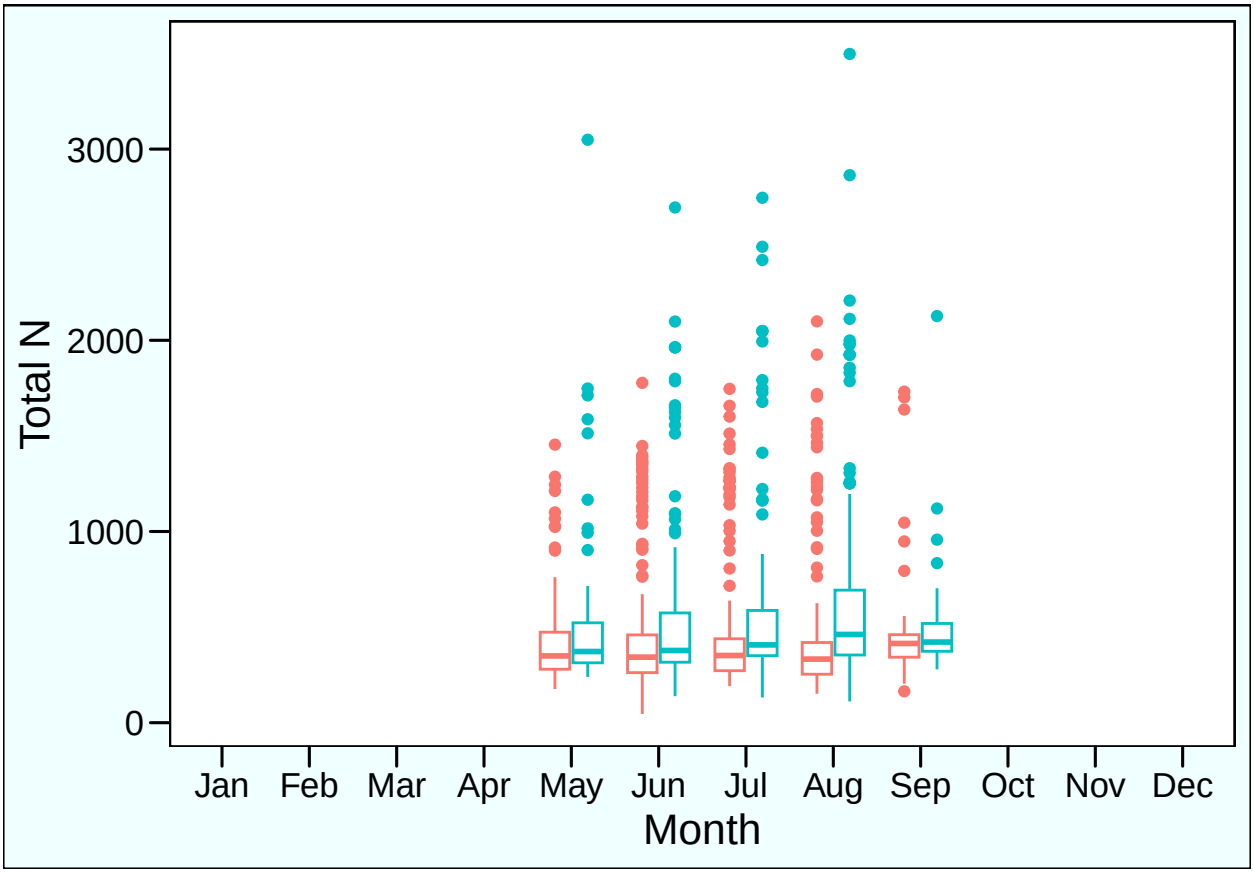
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#Boxplot C: TN w/o legend
tn_plot<-
  ggplot(PeterPaul.chem.nutrients,
         aes(x=factor(month,levels=1:12,labels=month.abb),
             y=tn_ug,
             color=lakename))+
 geom_boxplot()+
 scale_x_discrete(name="Month",
                  drop=FALSE)+
  theme(legend.position="none")+
  ylab("Total N")

tn_plot
```

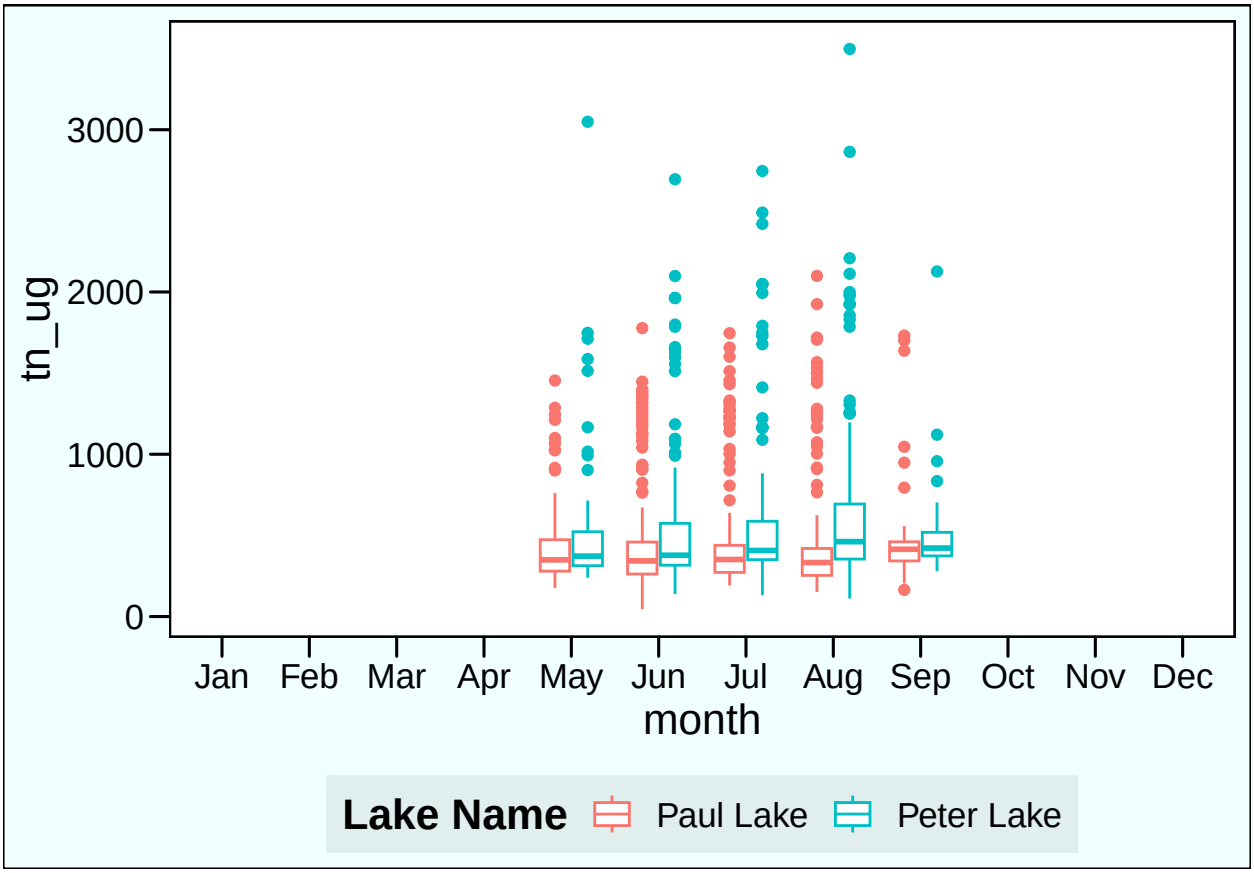
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#Boxplot C: TN WITH legend
tn_plot_legend<-
  ggplot(PeterPaul.chem.nutrients,
         aes(x=factor(month,levels=1:12,labels=month.abb),
             y=tn_ug,
             color=lakename))+
 geom_boxplot()+
 scale_x_discrete(name="month",
                  drop=FALSE)+
  labs(color="Lake Name")+
    theme(legend.position = "bottom")

tn_plot_legend
```

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




```
#Combined boxplots
#Create separate legend object
legend <- get_legend(tn_plot_legend)
```

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

```
#Combine plots A,B, and C above
combined_plot<-
plot_grid(temperature_plot,tp_plot,tn_plot,ncol=1, align = "v")
```

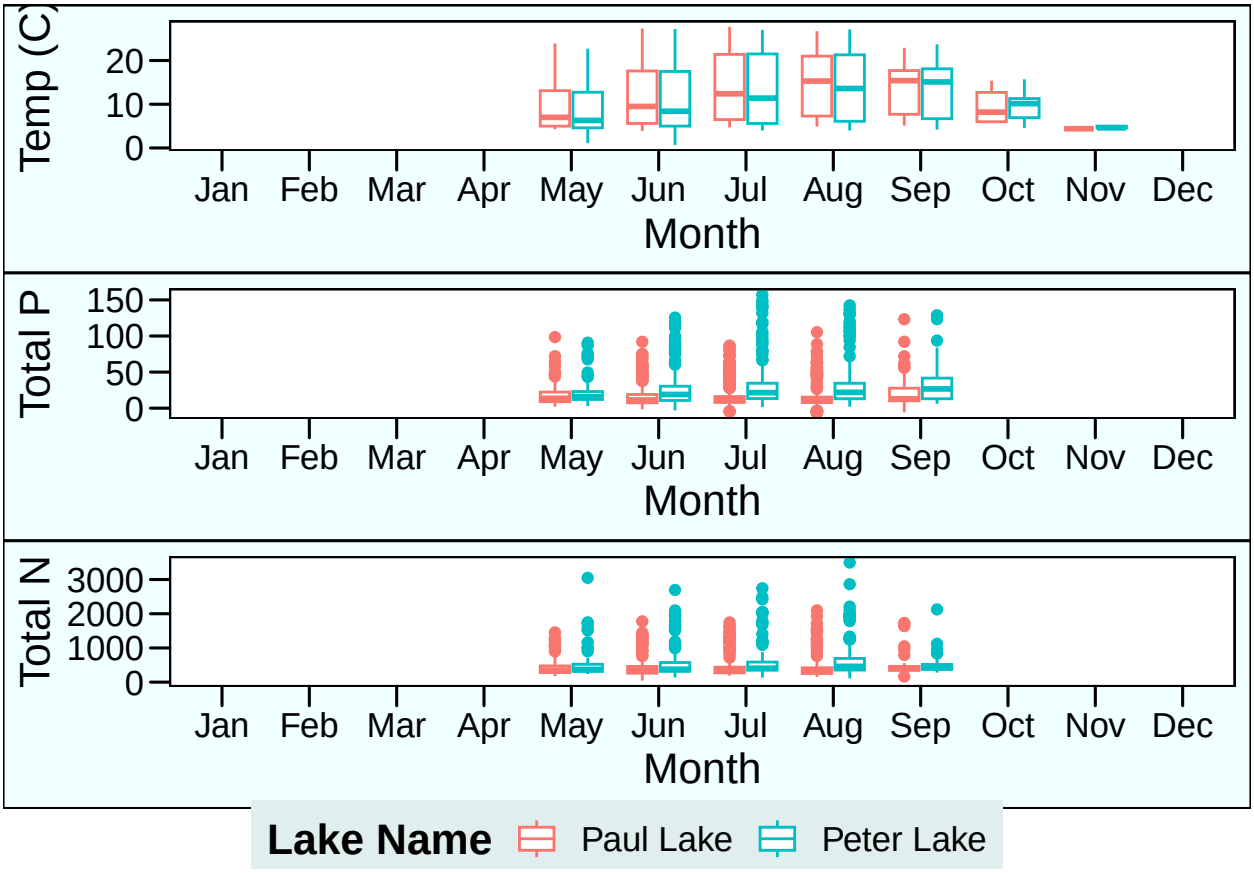
```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).

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

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

```
#Combine plots A-C with the legend
final_plot<-plot_grid(combined_plot,legend, ncol=1,rel_heights = c(0.93, 0.07))

final_plot
```



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

> Answer: As expected for lakes in Northern Michigan, the temperatures vary greatly across the seasons. The two lakes have relatively similar median and max temperatures. However, their minimums varied a lot more between the lakes. Depth of measurement is not taken into account at all for this chart but is a variable that could explain the large temperature interquartile ranges. It is interesting that both lakes had a large number of outliers for phosphorus and nitrogen. Despite the lakes being interconnected, Peter Lake had a consistently higher median phosphorus and nitrogen content throughout the year, with outlier values peaking in July and August. Phosphorus had a larger IQR range than nitrogen, especially in September at the end of summer.

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)

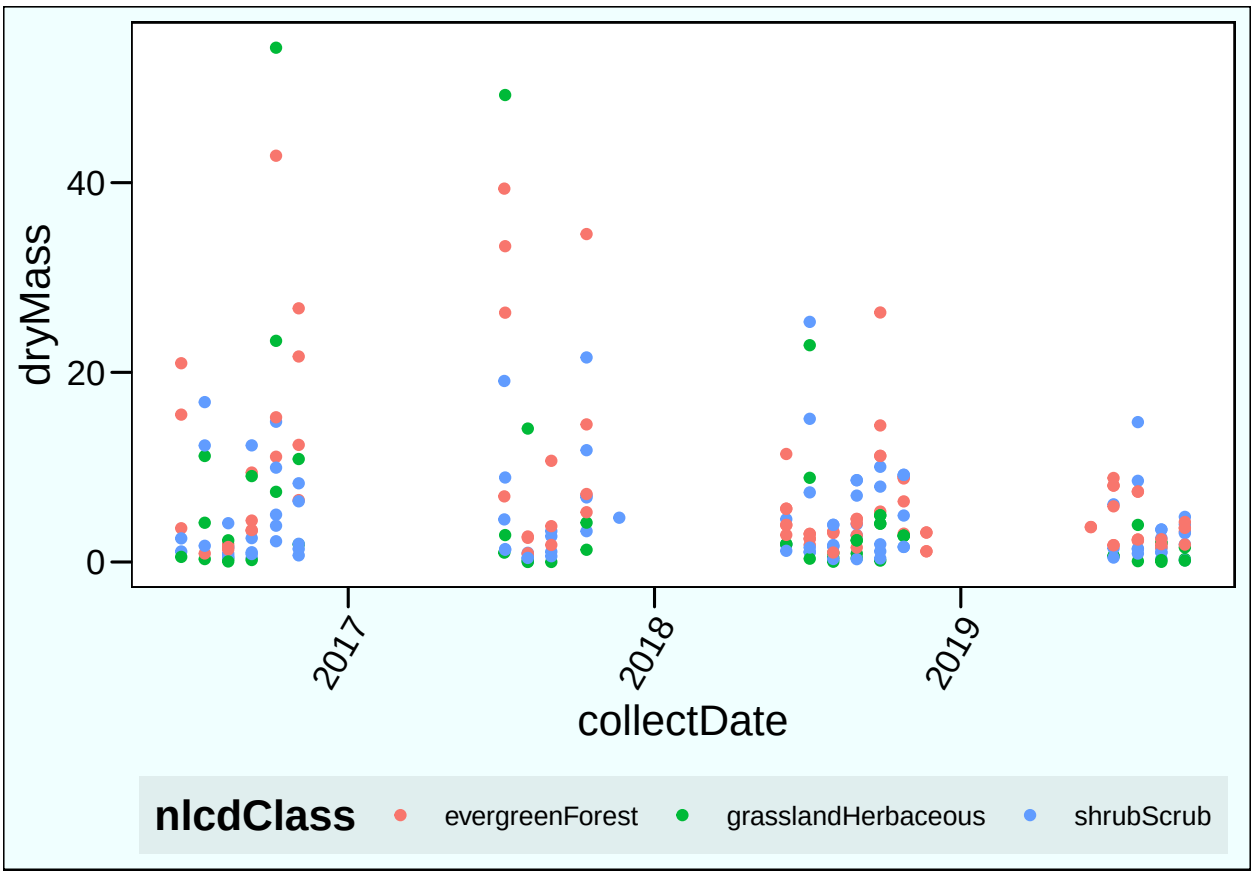
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6 Create plot of litter dataset color coded by NLCD

needles_plot1 <-
  ggplot(filter(litter,functionalGroup=="Needles"),
         aes(x=collectDate,
             y=dryMass,
             color=nlcdClass))+
```

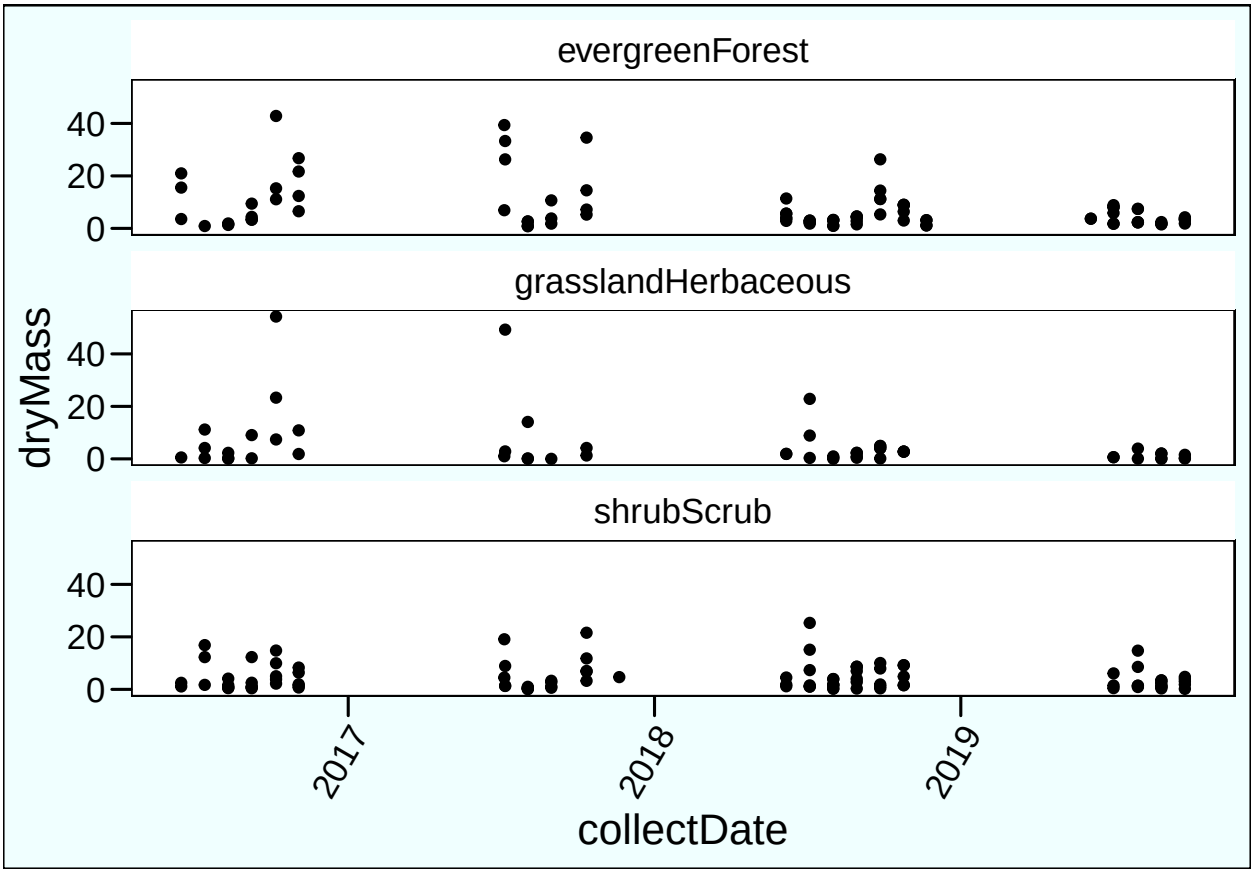
```
          geom_point()+
    theme(legend.position = "bottom")+
  theme(axis.text.x = element_text(angle = 60,
                                   hjust = 1))+
  theme(legend.text=element_text(size=10))

needles_plot1
```




```
#7 Update above chart to separate by facet
needles_plot2 <-
ggplot(filter(litter,functionalGroup=="Needles"),
         aes(x=collectDate,
             y=dryMass))+
           geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  theme(legend.position = "bottom")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))

needles_plot2
```



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

Answer: I believe that the needles_plot1, or the color coded version created in number 6, is more effective in this scenario. The single color coded plot is a lot easier to compare the dry mass weight across land use types as they share the same y axis.