# Assignment 5: Data Visualization

Hannah Nelson

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
library(tidyverse)
```

```
## -- 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 1.9.2     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()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(here)
```

```
## here() starts at /Users/hannahnelson/Desktop/env872/EDA-Fall2023
```

```r
library(cowplot)
```

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

```r
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```r
#2
here()
```

```
## [1] "/Users/hannahnelson/Desktop/env872/EDA-Fall2023"
```

```r
Lakes <-
read.csv(here(
"./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"
  ),
  stringsAsFactors = TRUE)

glimpse(Lakes)
```

```
## Rows: 23,008
## Columns: 15
## $ lakename         <fct> Paul Lake, Paul Lake, Paul Lake, Paul Lake, Paul Lake,~
## $ year4            <int> 1984, 1984, 1984, 1984, 1984, 1984, 1984, 1984, 1984, ~
## $ daynum           <int> 148, 148, 148, 148, 148, 148, 148, 148, 148, 148, 148,~
## $ month            <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ sampledate       <fct> 1984-05-27, 1984-05-27, 1984-05-27, 1984-05-27, 1984-0~
## $ depth            <dbl> 0.00, 0.25, 0.50, 0.75, 1.00, 1.50, 2.00, 3.00, 4.00, ~
## $ temperature_C    <dbl> 14.5, NA, NA, NA, 14.5, NA, 14.2, 11.0, 7.0, 6.1, 5.5,~
## $ dissolvedOxygen  <dbl> 9.5, NA, NA, NA, 8.8, NA, 8.6, 11.5, 11.9, 2.5, 1.6, 0~
## $ irradianceWater  <dbl> 1750.0, 1550.0, 1150.0, 975.0, 870.0, 610.0, 420.0, 22~
## $ irradianceDeck   <dbl> 1620, 1620, 1620, 1620, 1620, 1620, 1620, 1620, 1620, ~
## $ tn_ug            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ tp_ug            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ nh34             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ no23             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ po4              <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```r
Litter <- read.csv(here
("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

glimpse(Litter)
```

```
## Rows: 1,692
## Columns: 13
## $ plotID            <fct> NIWO_062, NIWO_061, NIWO_062, NIWO_064, NIWO_058, NIW~
## $ trapID            <fct> NIWO_062_050, NIWO_061_169, NIWO_062_050, NIWO_064_10~
## $ collectDate       <fct> 2016-06-16, 2016-06-16, 2016-06-16, 2016-06-16, 2016-~
## $ functionalGroup   <fct> Seeds, Other, Woody material, Seeds, Needles, Leaves,~
## $ dryMass           <dbl> 0.000, 0.270, 0.120, 0.000, 1.110, 0.000, 0.000, 0.00~
## $ qaDryMass         <fct> N, N, N, N, Y, N, N, N, N, N, N, Y, N, N, N, N, N, Y,~
## $ subplotID         <int> 31, 41, 31, 32, 32, 32, 40, 40, 40, 40, 40, 31, 31, 3~
## $ decimalLatitude   <dbl> 40.05114, 40.04762, 40.05114, 40.04737, 40.04872, 40.~
## $ decimalLongitude <dbl> -105.5858, -105.5861, -105.5858, -105.5840, -105.5872~
## $ elevation         <dbl> 3477.0, 3413.4, 3477.0, 3373.2, 3446.4, 3446.4, 3509.~
## $ nlcdClass         <fct> shrubScrub, evergreenForest, shrubScrub, evergreenFor~
## $ plotType          <fct> tower, tower, tower, tower, tower, tower, tower, towe~
## $ geodeticDatum     <fct> WGS84, WGS84, WGS84, WGS84, WGS84, WGS84, WGS84, WGS8~
```

## 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

```r
#3
hn_theme <- theme_base() +
  theme(plot.background = element_rect(color= "grey", fill = "white"),
  plot.title = element_text(color= "navy"),
  axis.title.x = element_text(color= "navy"),
  axis.title.y = element_text(color= "navy"),
  legend.position = "bottom")

#set default theme to my theme
theme_set(hn_theme)
```

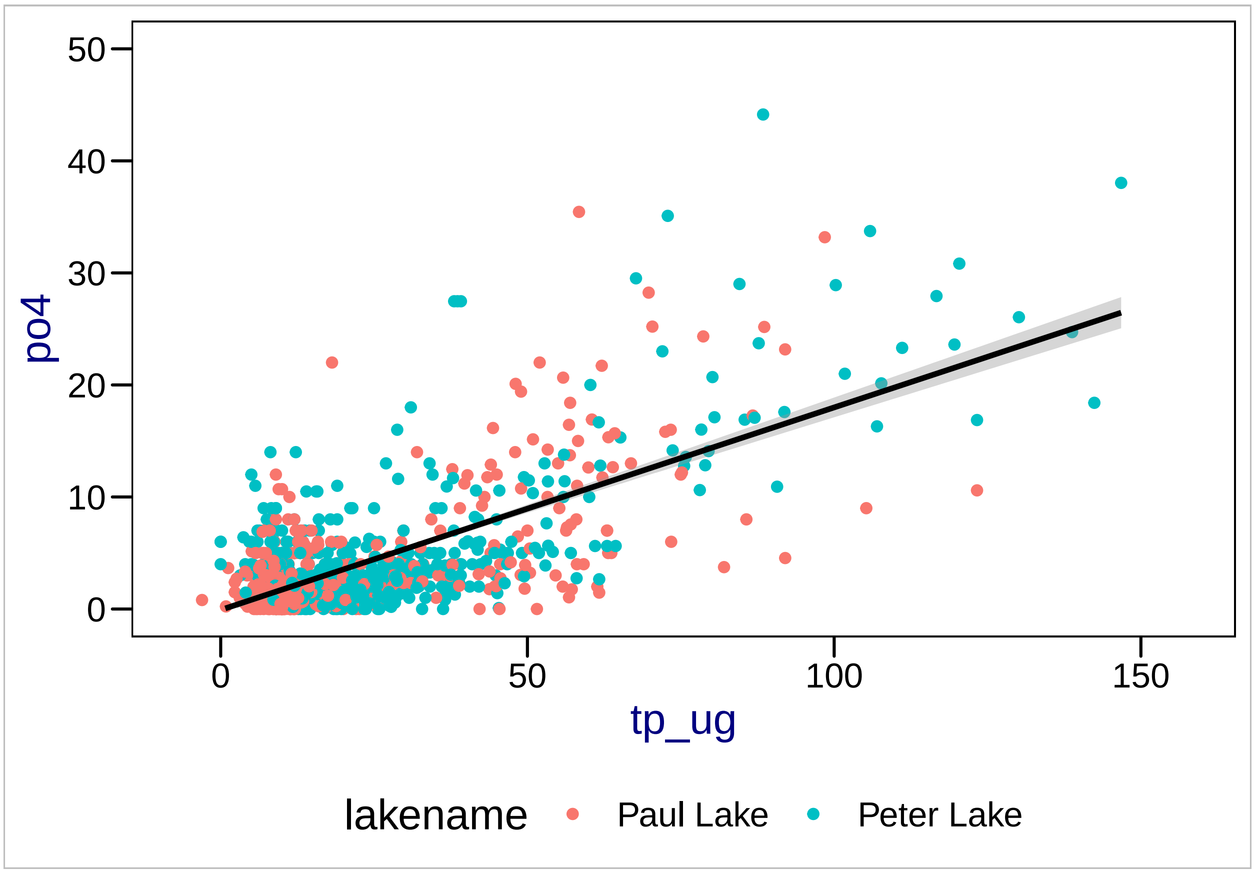
## Create graphs

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()`).

```r
#4
ggplot(Lakes, aes(x=tp_ug,
                  y=po4,
                  color=lakename))+
      geom_point()+
      ylim(0, 50)+
     geom_smooth(method="lm", color="black")
```

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

## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21947 rows containing missing values (`geom_point()`).

## Warning: Removed 2 rows containing missing values (`geom_smooth()`).
```

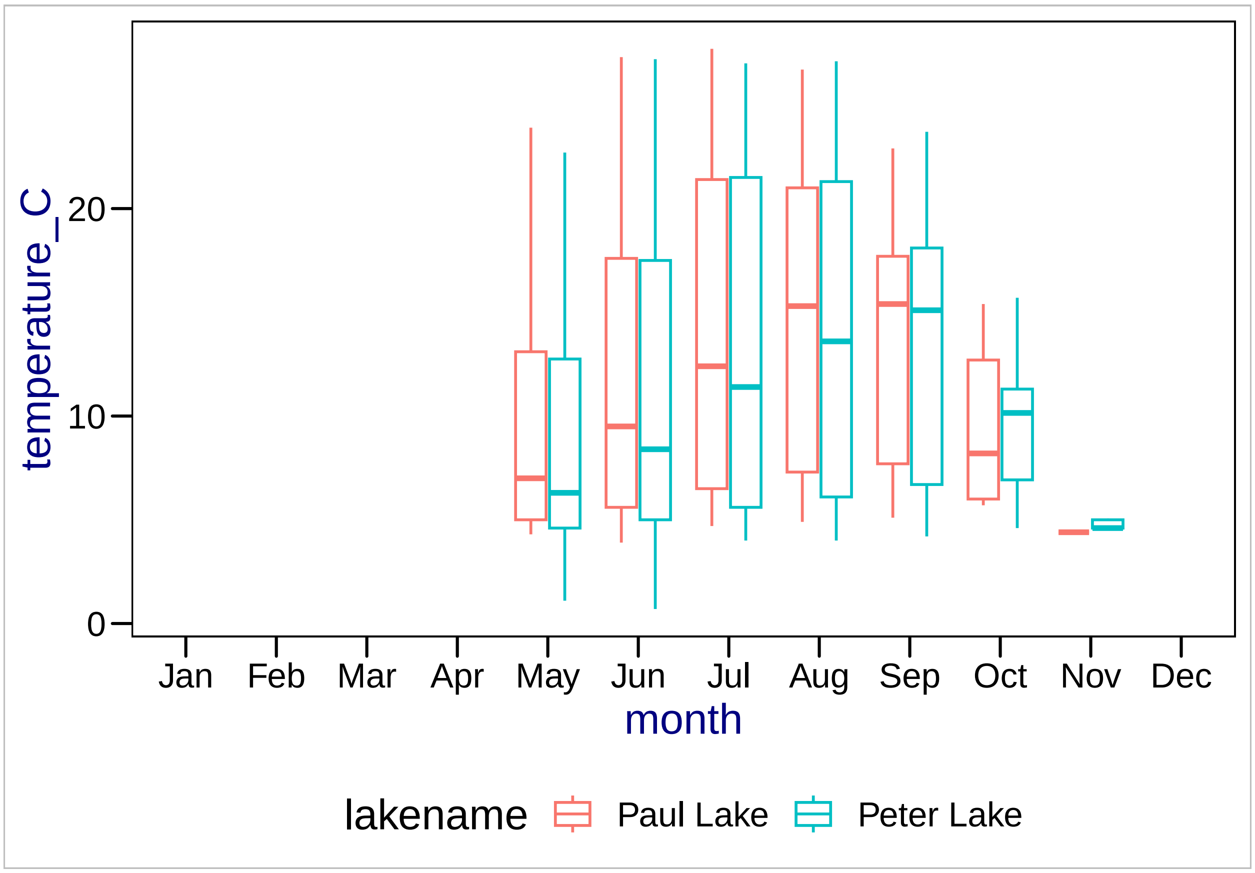



5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

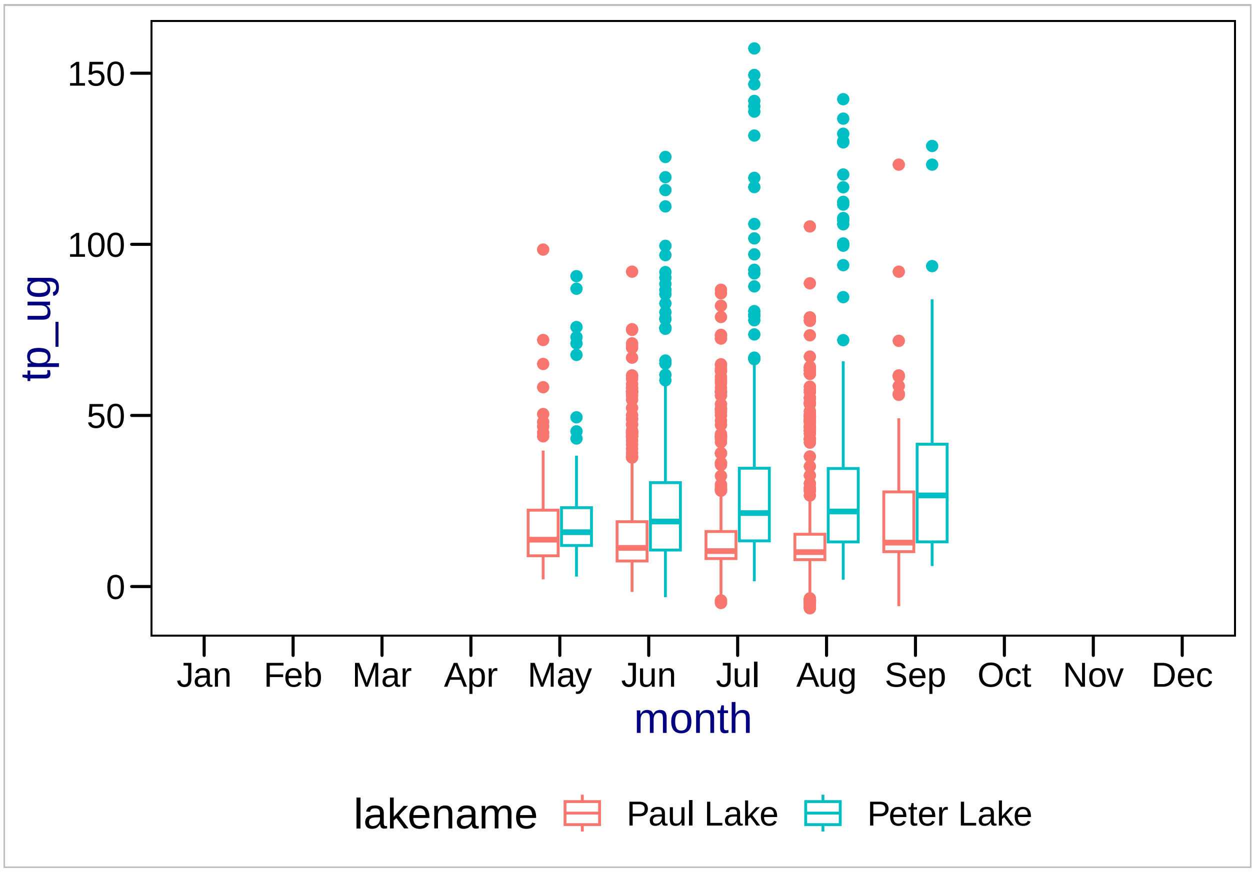
```r
#5
#a)
temp <- ggplot(Lakes, aes(x=factor(month,levels=1:12, labels=month.abb),
                  y=temperature_C,
                  color=lakename))+
  geom_boxplot()+
  scale_x_discrete(name="month", drop=FALSE)

temp
```

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



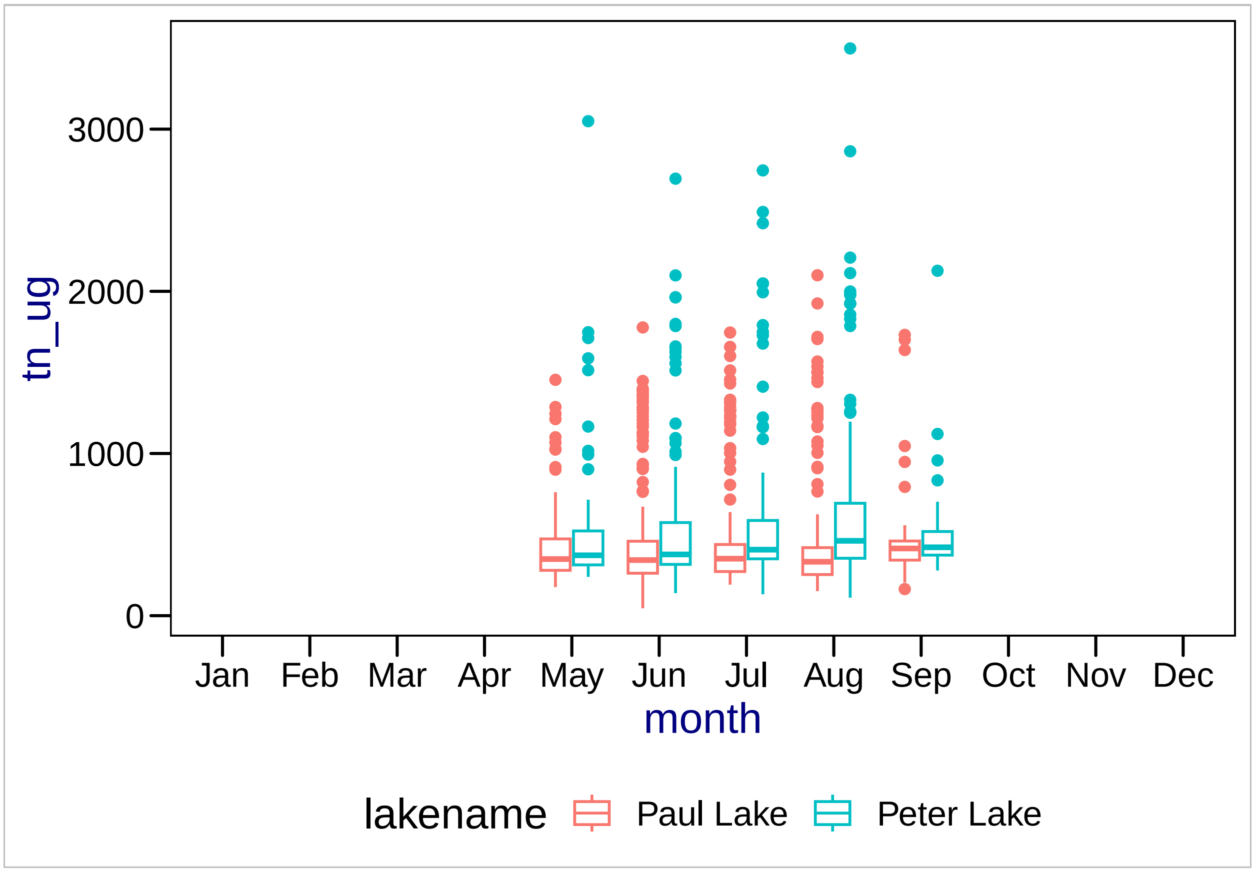

```r
#b)
TP <- ggplot(Lakes, aes(x=factor(month,levels=1:12, labels=month.abb),
                  y=tp_ug,
                  color=lakename))+
  geom_boxplot()+
  scale_x_discrete(name="month", drop=FALSE)

TP
```

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



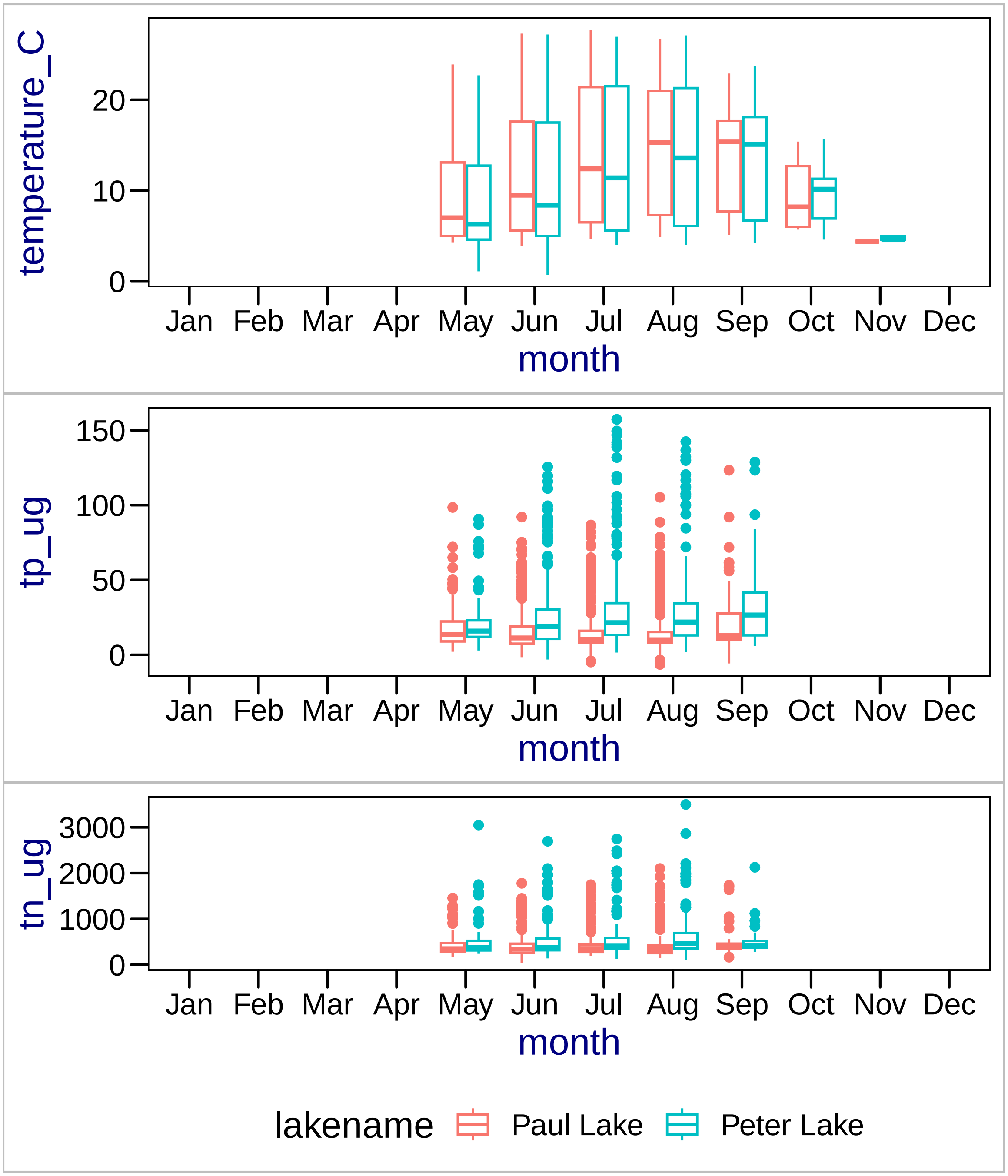

```
#c)
TN <- ggplot(Lakes, aes(x=factor(month,levels=1:12, labels=month.abb),
                    y=tn_ug,
                    color=lakename))+
  geom_boxplot()+
  scale_x_discrete(name="month", drop=FALSE)

TN
```

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




```
#removing legends from two plots so only one legend will be present on cowplot
temp2 <- temp + theme(legend.position = "none")

TP2 <- TP + theme(legend.position = "none")
```

```
#cowplot aligned vertically
plot_grid(temp2, TP2, TN, nrow=3, align="v")
```




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

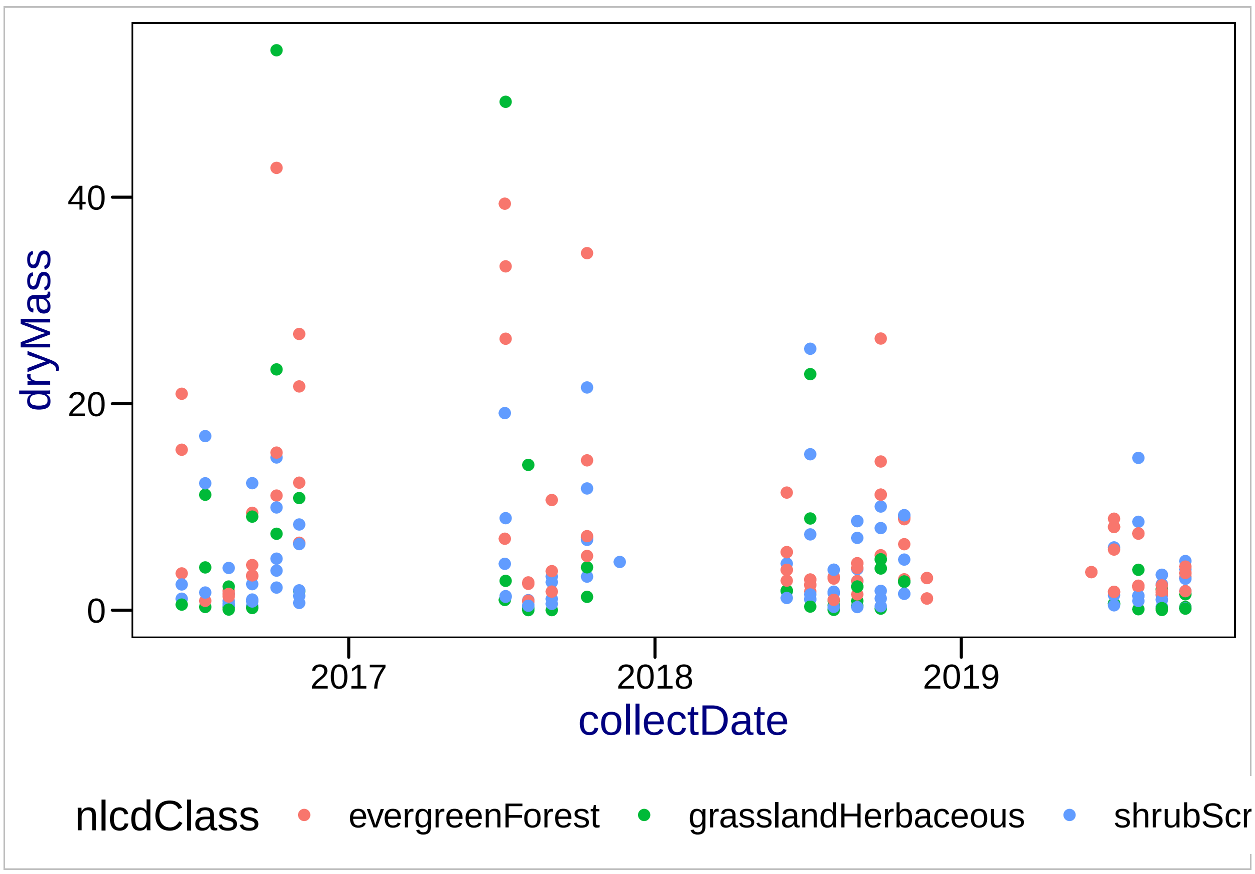
> Answer: Average temperature is highest in August and September for both Peter and Paul lakes with a larger range in August. The range of temperatures for both lakes is very small in November. Data for total nitrogen for Peter and Paul lakes looks similar overall with average levels and range and values for Peter Lake trending a bit higher. While phosphorus numbers are much lower than nitrogen numbers, the pattern of the data between lakes is similar to that of the nirgoen data. The highest means for ntirogen and phosphorous levels in Peter and Paul lakes are occuring in August and September.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group.

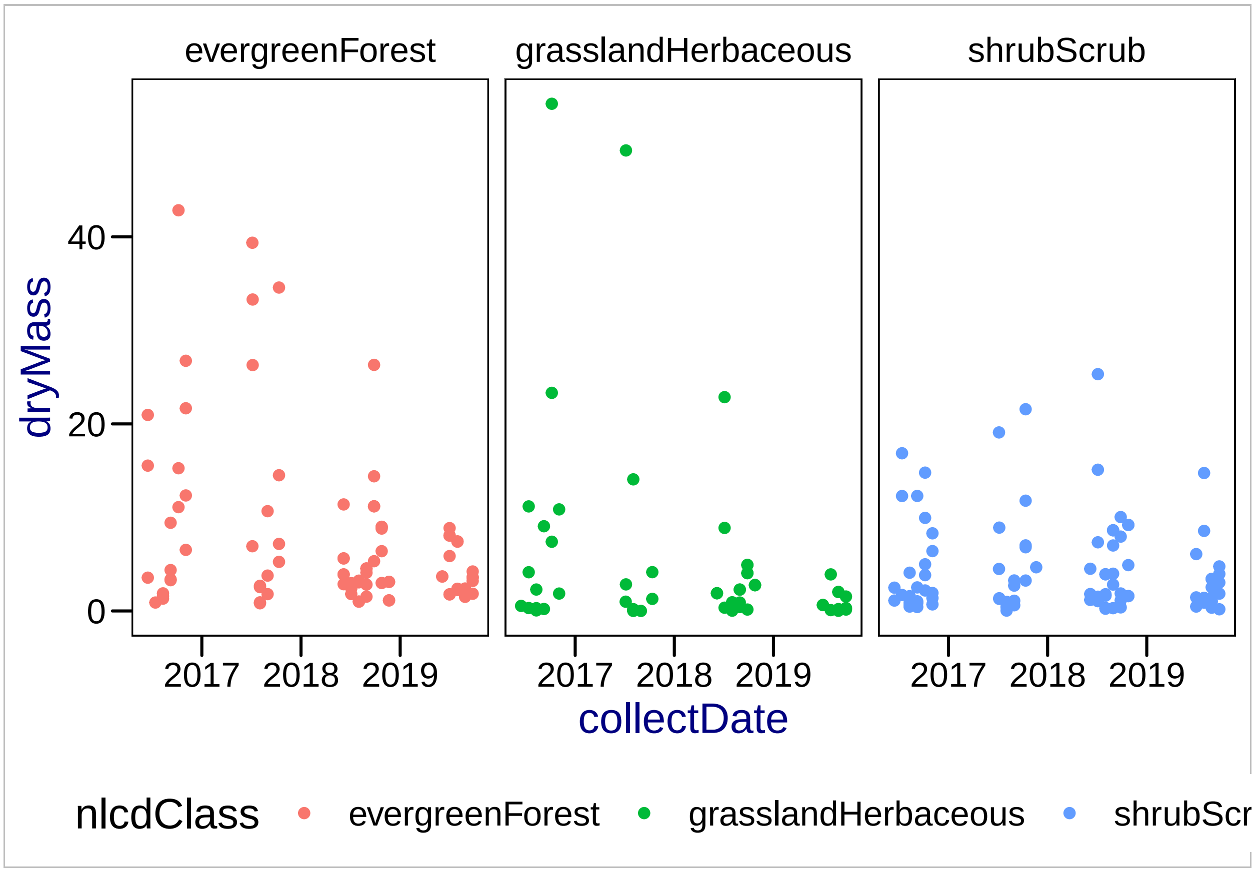
Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

#change collectDate class to date
Litter$collectDate <- ymd(Litter$collectDate)


Needles <- Litter %>%
  filter(functionalGroup == "Needles")

ggplot(Needles, aes(x=collectDate,
                    y=dryMass,
                   color=nlcdClass))+
      geom_point()
```




```
#7
ggplot(Needles, aes(x=collectDate,
                    y=dryMass,
                   color=nlcdClass))+
      geom_point()+
      facet_wrap(vars(nlcdClass))
```




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

> Answer: Plot 7 is more effective because it allows you to more effectively visually compare all three classes. In plot 6, many of the data points are on top of each other or very close to each other, and that makes it hard to see all of the data points. Plot 7 shows the data points more clearly, and it is easy to compare the classes when they are side by side.