Assignment 5: Data Visualization

Kamil Orozco

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 setting up the session with necessary packages
library(tidyverse); library(lubridate); library(here); library(cowplot)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
```

```
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
## here() starts at /Users/kariis/Documents/EDA2
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(ggplot2)
library(ggridges)
# verifying home directory
getwd()
```

```
## [1] "/Users/kariis/Documents/EDA2"
```

```
#reading in processed data files for Peter and Paul Lakes
Peter.Paul.chem.physics <-
  read.csv(here("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)

Niwot.Ridge.Litter <-
  read.csv(here("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)

#2 reading as date confirmation
class(Peter.Paul.chem.physics$sampledate)
```

```
## [1] "factor"
```

```
class(Niwot.Ridge.Litter$collectDate)
```

```
## [1] "factor"
```

```
#fixing the dates to be read as date
Peter.Paul.chem.physics$sampledate <- ymd(Peter.Paul.chem.physics$sampledate)
Niwot.Ridge.Litter$collectDate <- ymd(Niwot.Ridge.Litter$collectDate)
#reconfirming fixed dates
class(Peter.Paul.chem.physics$sampledate)
```

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

```
class(Niwot.Ridge.Litter$collectDate)
```

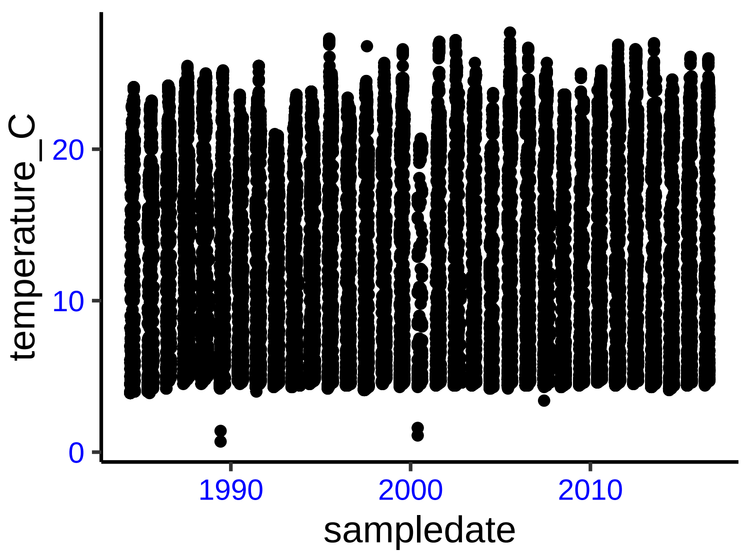
```
## [1] "Date"
```

Define your theme

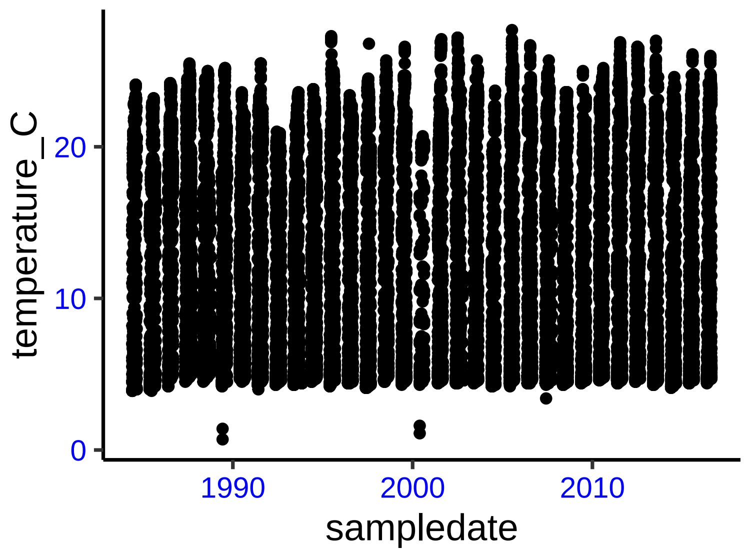
3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

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

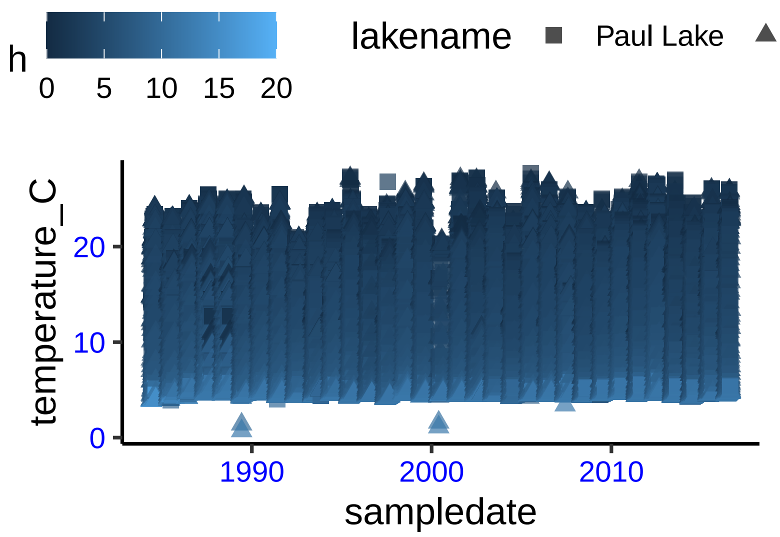
```
#3 building a theme
ThemeBuilder <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "blue"),
        legend.position = "top")

theme_set(ThemeBuilder)

LakesPlot <- ggplot(Peter.Paul.chem.physics) +
  geom_point(aes(x = sampledate, y = temperature_C))
print(LakesPlot)
```




```
LakesPlot2 <- ggplot(Peter.Paul.chem.physics) +
  geom_point(aes(x = sampledate, y = temperature_C)) +
  ThemeBuilder
print(LakesPlot2)
```




```
LakesPlot3 <-
  ggplot(Peter.Paul.chem.physics, aes(x = sampledate, y = temperature_C, color= depth, shape = lakename
  geom_point(alpha = 0.7, size = 2.5) +
  scale_shape_manual(values = c(15, 17)) +
  theme(legend.position = "right",
        legend.text = element_text(size= 12), legend.title = element_text(size = 12)) +
  ThemeBuilder
print(LakesPlot3)
```



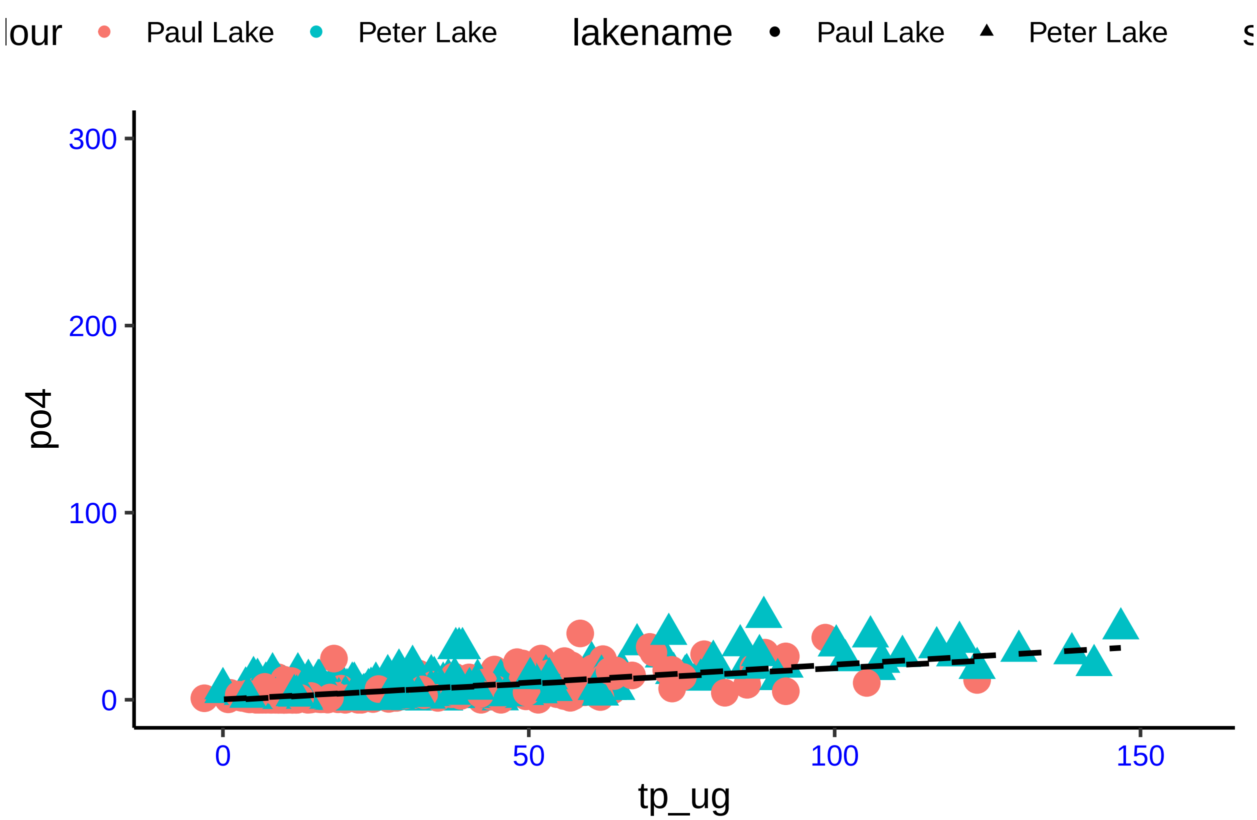
Create graphs

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 plotting total phosphorus and phosphate

PeterandPaulPhos <-
  ggplot(Peter.Paul.chem.physics, aes(x = tp_ug, y= po4, color ='red', 'blue', shape= lakename)) +
    geom_point(aes(color=lakename, size=2)) +   #distinguishing the lakes
    geom_smooth(method=lm, se=FALSE, col='black', linetype='dashed') +  #line of best fit
  ylim(0, 300) +    #detaching outlier
    ThemeBuilder   #applying my blue number original theme
print(PeterandPaulPhos)
```

```
## `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 4 rows containing missing values (`geom_smooth()`).
```

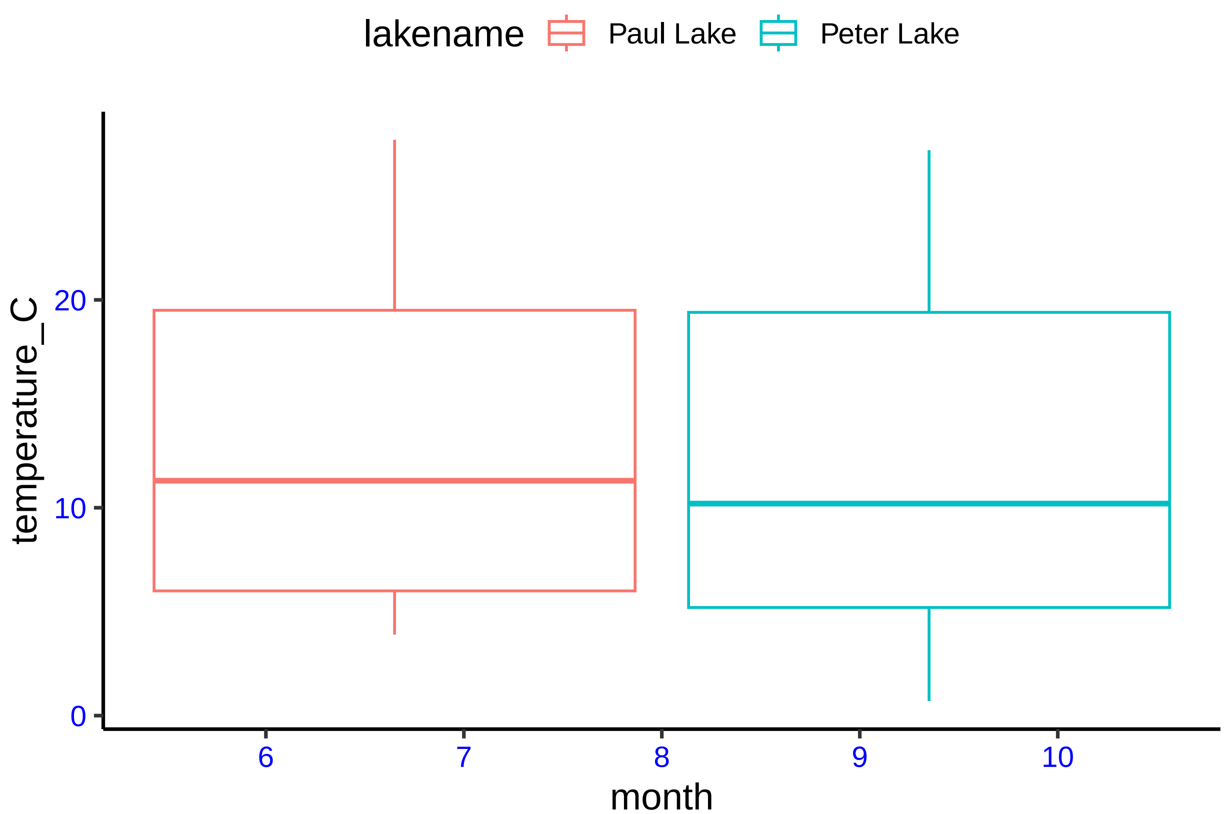


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: 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

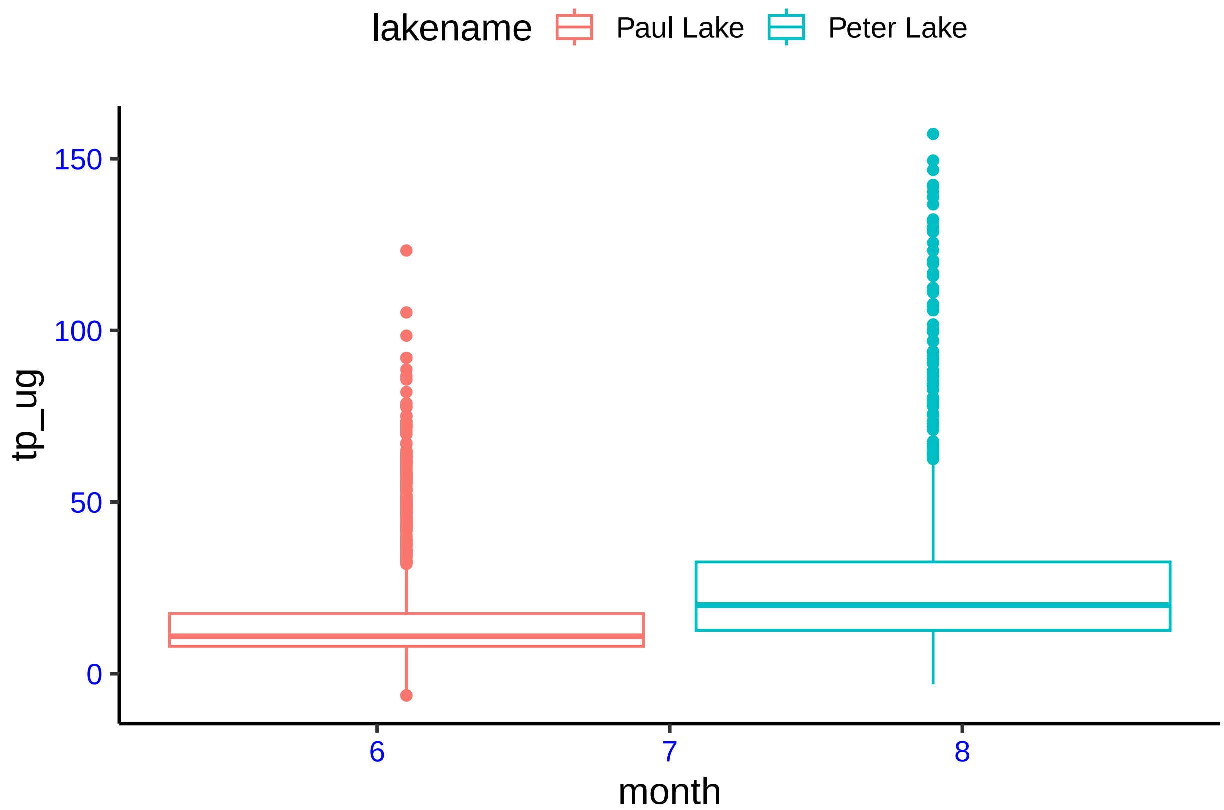
```
#5 #creating three separate boxplots
#part a
BoxPlotTemp <-
  ggplot(Peter.Paul.chem.physics, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename))
print(BoxPlotTemp)
```

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



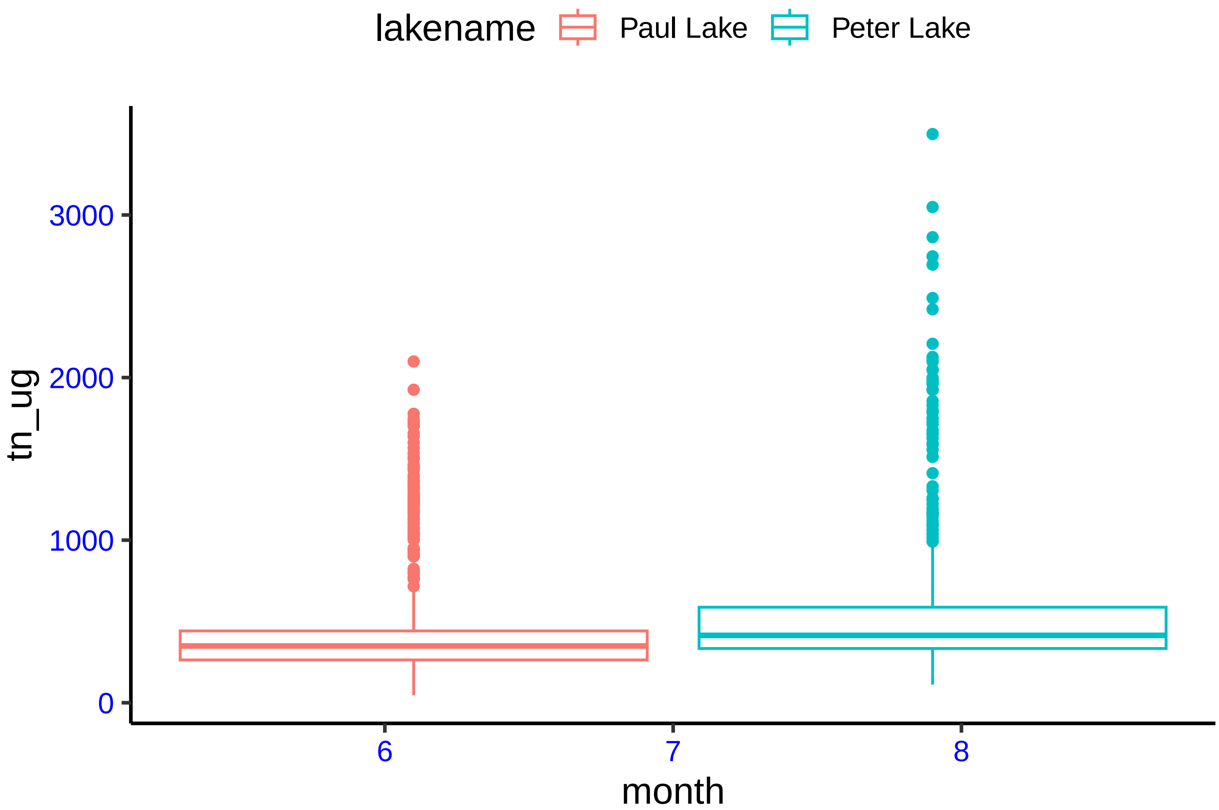

```
#part b
BoxPlotTP <-
  ggplot(Peter.Paul.chem.physics, aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename))
print(BoxPlotTP)
```

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




```
# part c
BoxPlotTN <-
  ggplot(Peter.Paul.chem.physics, aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename))
print(BoxPlotTN)
```

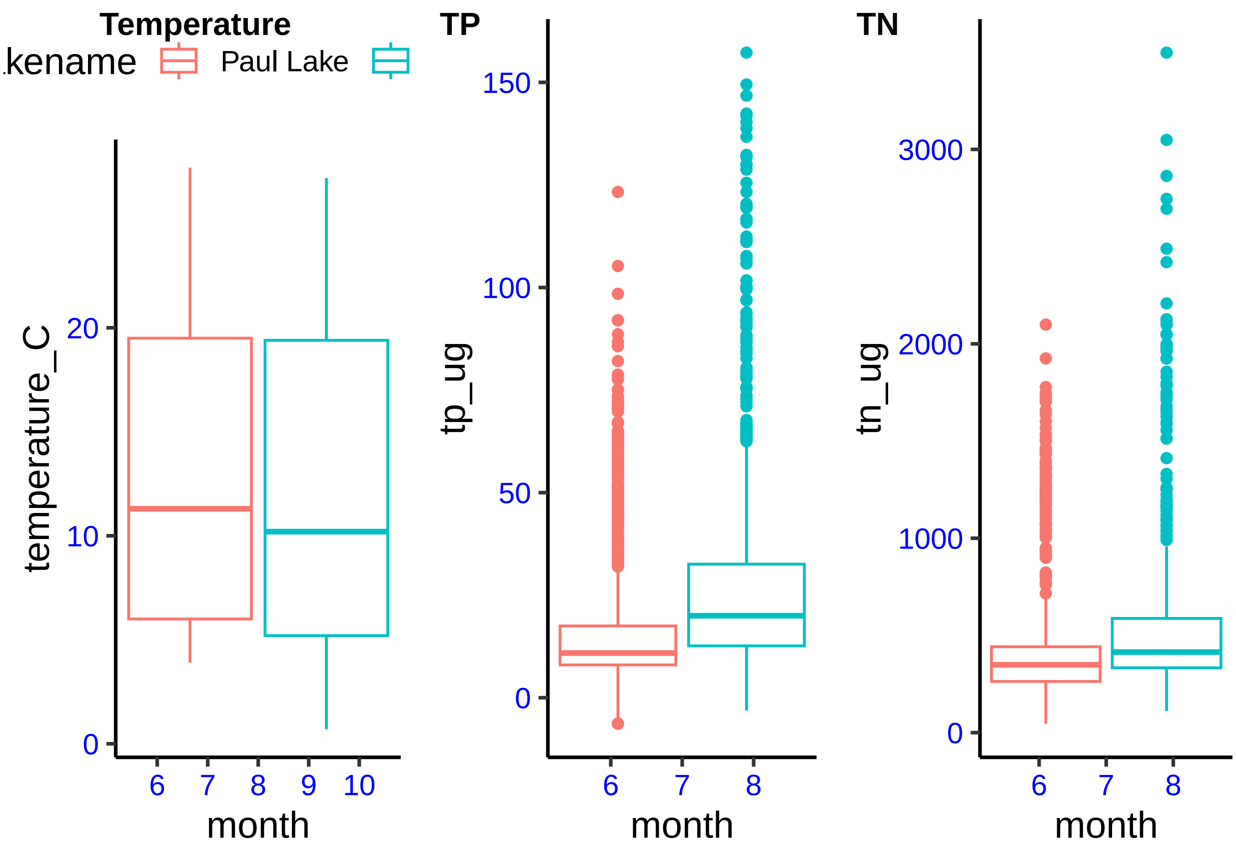
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#creating a cowplot
CombinedBoxPlots <-
  plot_grid(
    BoxPlotTemp + theme(legend.box.margin = margin(0, 0, 0, 12)), #keeping one graph's legend
    BoxPlotTP + theme(legend.position = "none"), #getting rid of these graphs' legends
    BoxPlotTN + theme(legend.position = "none"),
    nrow = 1, align = 'h', rel_heights = c(1.25, 1), labels = c('Temperature', 'TP', 'TN'), label_size =
```

```
## 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()`).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
print(CombinedBoxPlots)
```

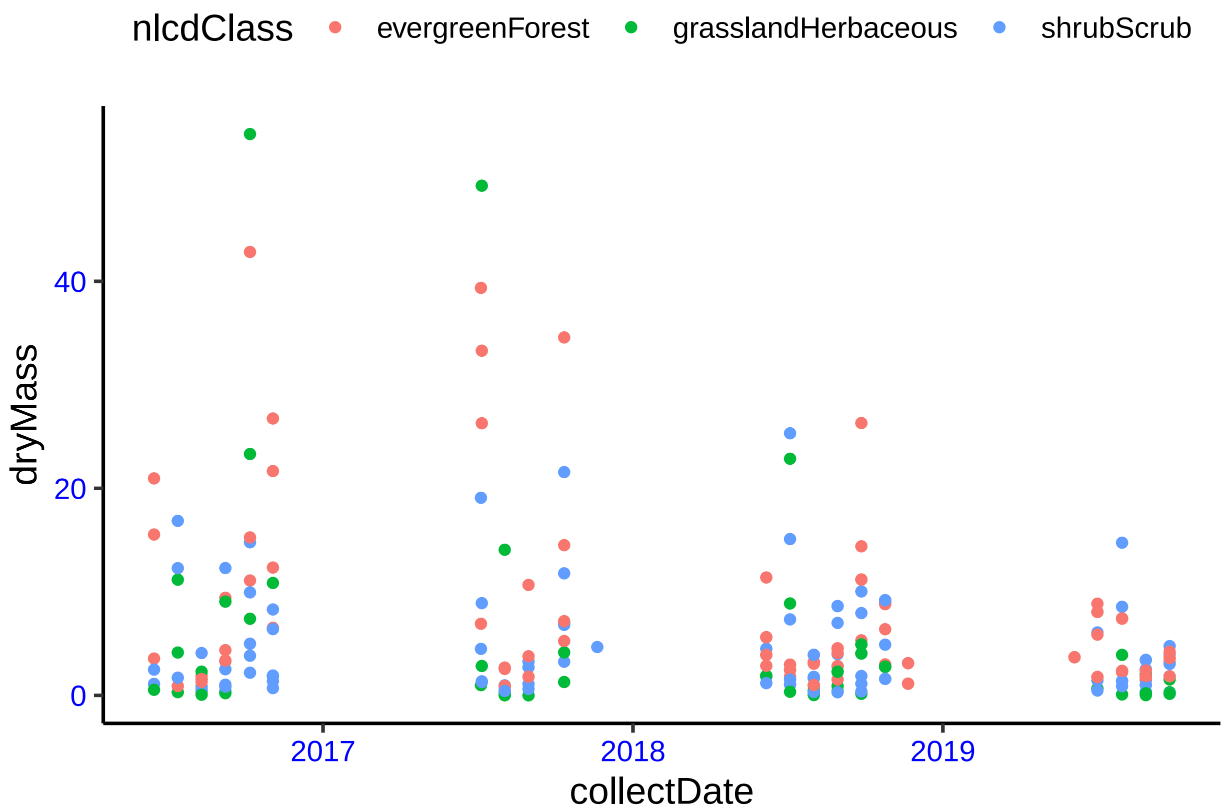


Question: What do you observe about the variables of interest over seasons and between lakes?

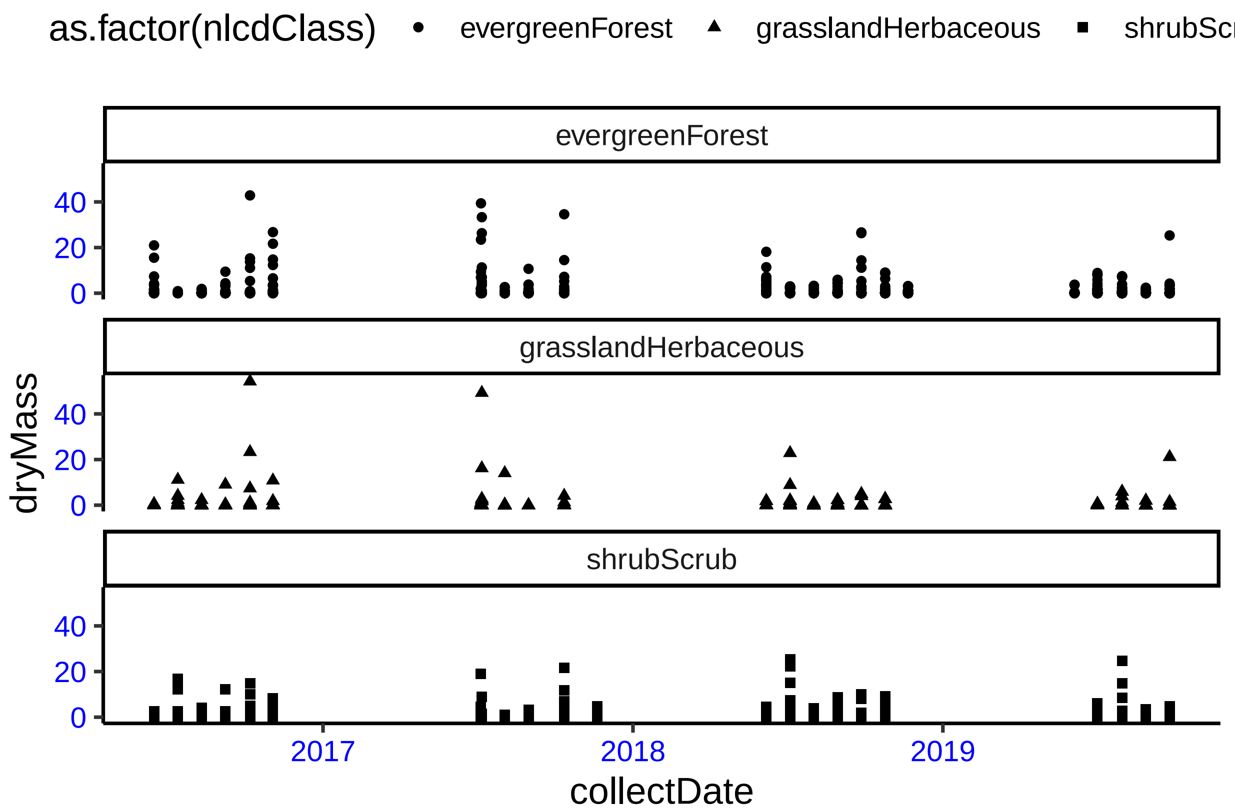
> Answer: Peter Lake generally sees an uptick in temperature in the later summer months/early fall (August-October) whereas Peter Lake sees higher TP and TN than Paul Lake; which tends to see its hot season in the early months of summer. This tells us that they have different nutrient activity but confirms that Peter Lake sees higher rates of nutrient activity than Paul Lake does.

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 Plot 'Needles' functional group, plot dry mass by date AND NLCD class has a color aes
NiwotRidgePlot <-
  ggplot(subset(Niwot.Ridge.Litter, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point()
print(NiwotRidgePlot)
```




```
#7  separating into 3 facets
NWRFaceted <-
    ggplot(Niwot.Ridge.Litter, aes(x = collectDate, y = dryMass, shape = as.factor(nlcdClass))) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(NWRFaceted)
```



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

> Answer: The graph from #7 is more effective in comparing the difference in dry mass of needles by varying tree type. It is this way because when data is faceted (#7), the data is divided into subsets of the same varibale (and plot type) which makes for a viewer friendly comparison of distribution relationships. Faceting data is mostly useful when comparing multiple variables in general!