Assignment 5: Data Visualization

Andreana Chou

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.

```r
#1

library(tidyverse)     #load libraries
```

```
## -- 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 error
```

```
library(lubridate)
library(cowplot)
```

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

```
library(here)
```

```
## here() starts at C:/Users/andre/Documents/R Studio Files/EDE_Fall2023
```

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

#read in NTL-LTER dataset with here()

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

#read in Niwot dataset with here()

#2

Niwot$collectDate <- ymd(Niwot$collectDate)     #convert to date format with ymd()
class(Niwot$collectDate)     #verify format
```

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

```
NTL_LTF$sampledate <- ymd(NTL_LTF$sampledate)   #convert to date format with ymd()
class(NTL_LTF$sampledate)    #verify format
```

```
## [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

my_theme <- theme(panel.background = element_rect(color = "lightblue",
                                                  fill = "white"),
                  panel.grid.major=element_line(color="lightblue",
                                                linewidth=0.5),
                  legend.title = element_text(color = "black"))

#panel.background sets up plot background
#panel.grid.major sets up a grid and axis ticks
#legend.title sets up the legend

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.

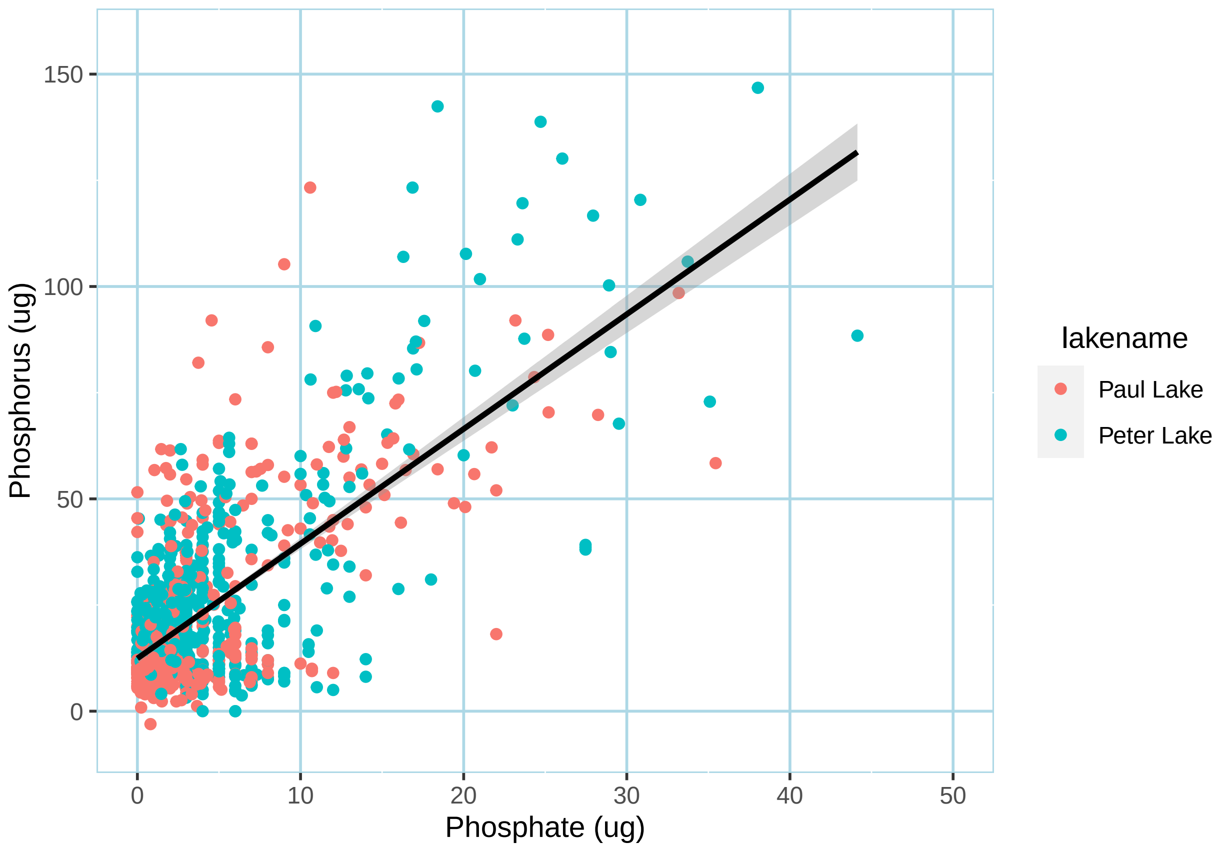
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

Plot_4 <- NTL_LTF %>% ggplot(aes(x=po4, y=tp_ug)) +
  geom_point(aes(color=lakename)) +
  xlim(0, 50) +
  geom_smooth(method="lm", color="black") +
  xlab("Phosphate (ug)") +
  ylab("Phosphorus (ug)")

#used geom_point() to create scatter plot when both variables are numeric
#set color=lakename as an argument within geom_point() aesthetics
#added geom_smooth() to create line of best fit (method="lm") and set color to black

Plot_4
```

```
## ‘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()‘).
```



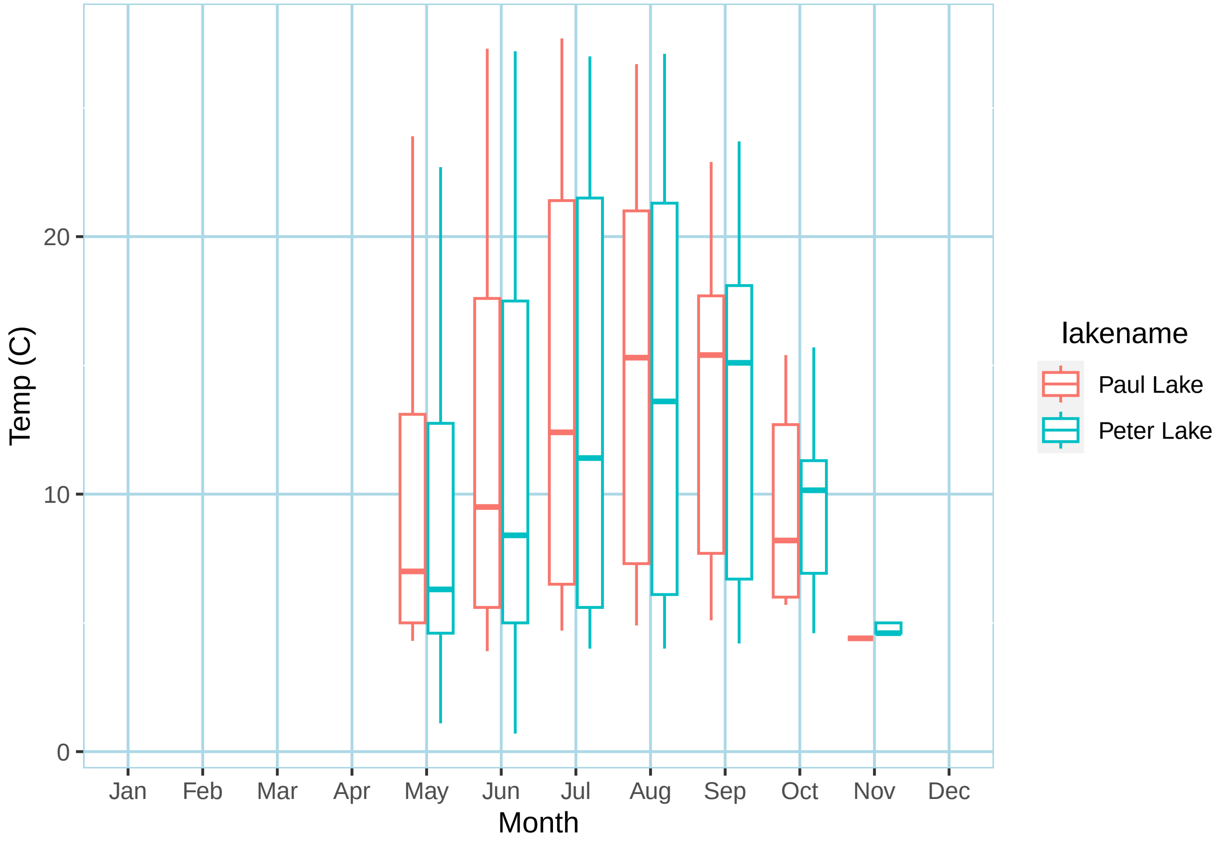
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

Plot_5_temp <- NTL_LTF %>% ggplot(aes(x=factor(NTL_LTF$month, levels=1:12,
                                               labels=month.abb),
                                      y=temperature_C, color=lakename)) +
  geom_boxplot() +
  scale_x_discrete(name="Month", drop=FALSE) +
  ylab("Temp (C)")
Plot_5_temp
```

```
## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.
```

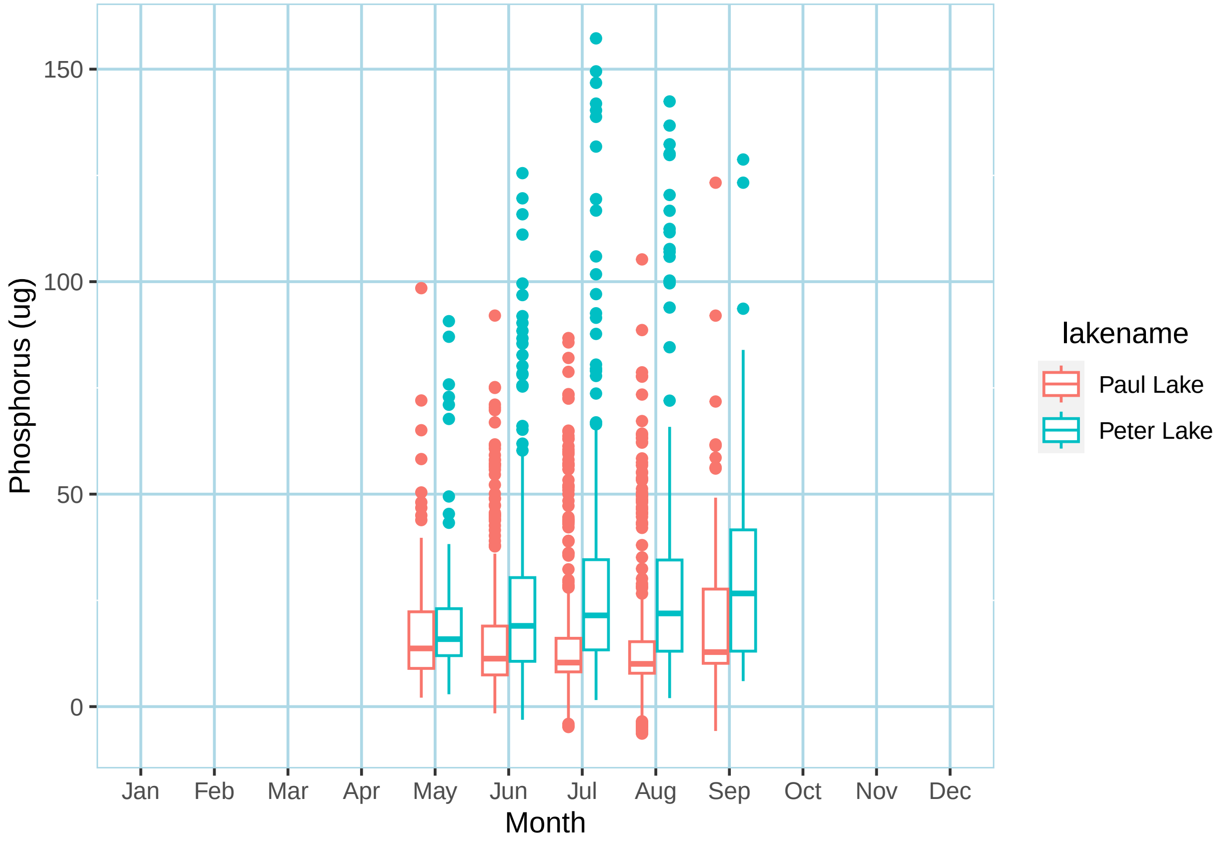
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



```
#boxplot for temperature using geom_boxplot()
#converted months to factor before setting to x axis, ensured months were in levels
  #1-12 for all 12 months, and labeled using their month abbreviations
#set color=lakename as an argument within ggplot aesthetics

Plot_5_TP <- NTL_LTF %>% ggplot(aes(x=factor(NTL_LTF$month, levels=1:12,
                                       labels=month.abb),
                                  y=tp_ug, color=lakename)) +
  geom_boxplot() +
  scale_x_discrete(name="Month", drop=FALSE) +
  ylab("Phosphorus (ug)")
Plot_5_TP
```

```
## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.
```

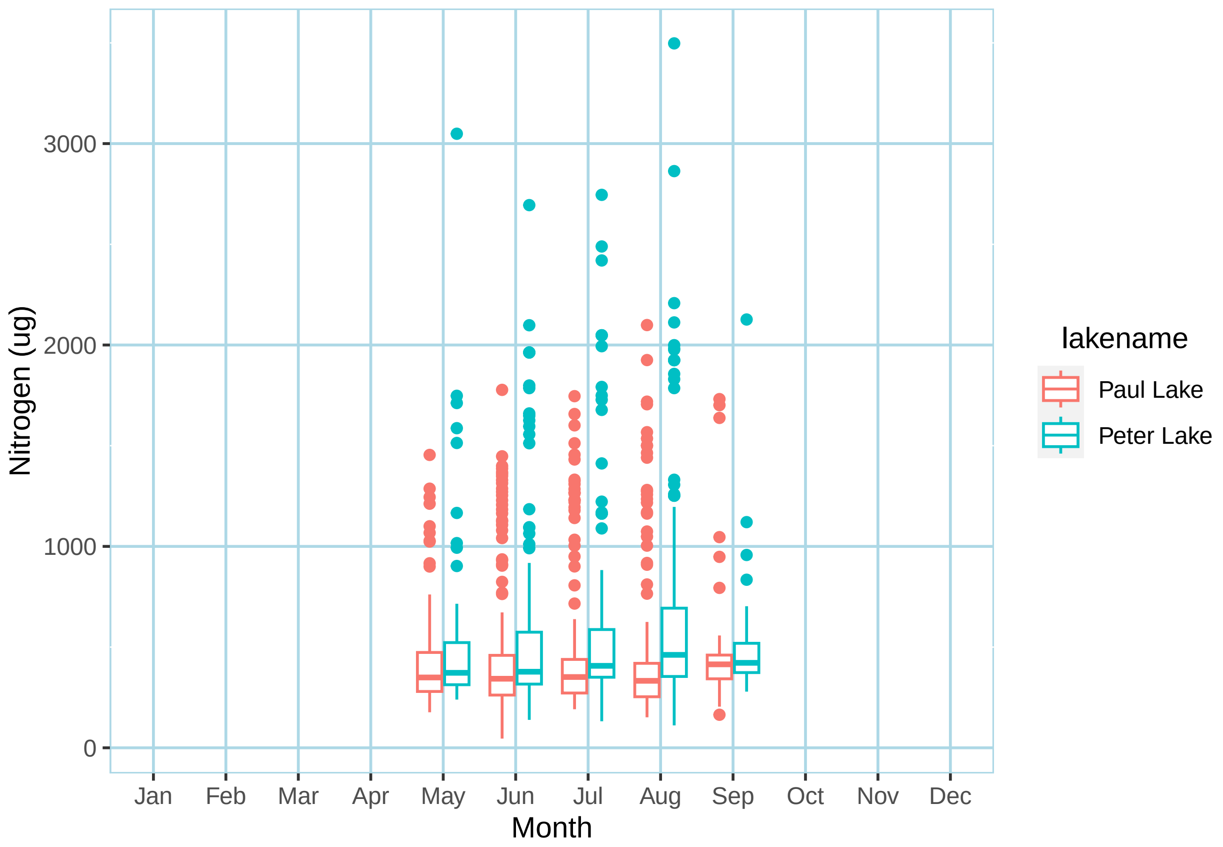
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```



```
#boxplot for TP using geom_boxplot()
#converted months to factor before setting to x axis, ensured months were in levels
  #1-12 for all 12 months, and labeled using their month abbreviations
#set color=lakename as an argument within ggplot aesthetics

Plot_5_TN <- NTL_LTF %>% ggplot(aes(x=factor(NTL_LTF$month, levels=1:12,
                                        labels=month.abb),
                                  y=tn_ug, color=lakename)) +
  geom_boxplot() +
  scale_x_discrete(name="Month", drop=FALSE) +
  ylab("Nitrogen (ug)")
Plot_5_TN
```

```
## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.
```

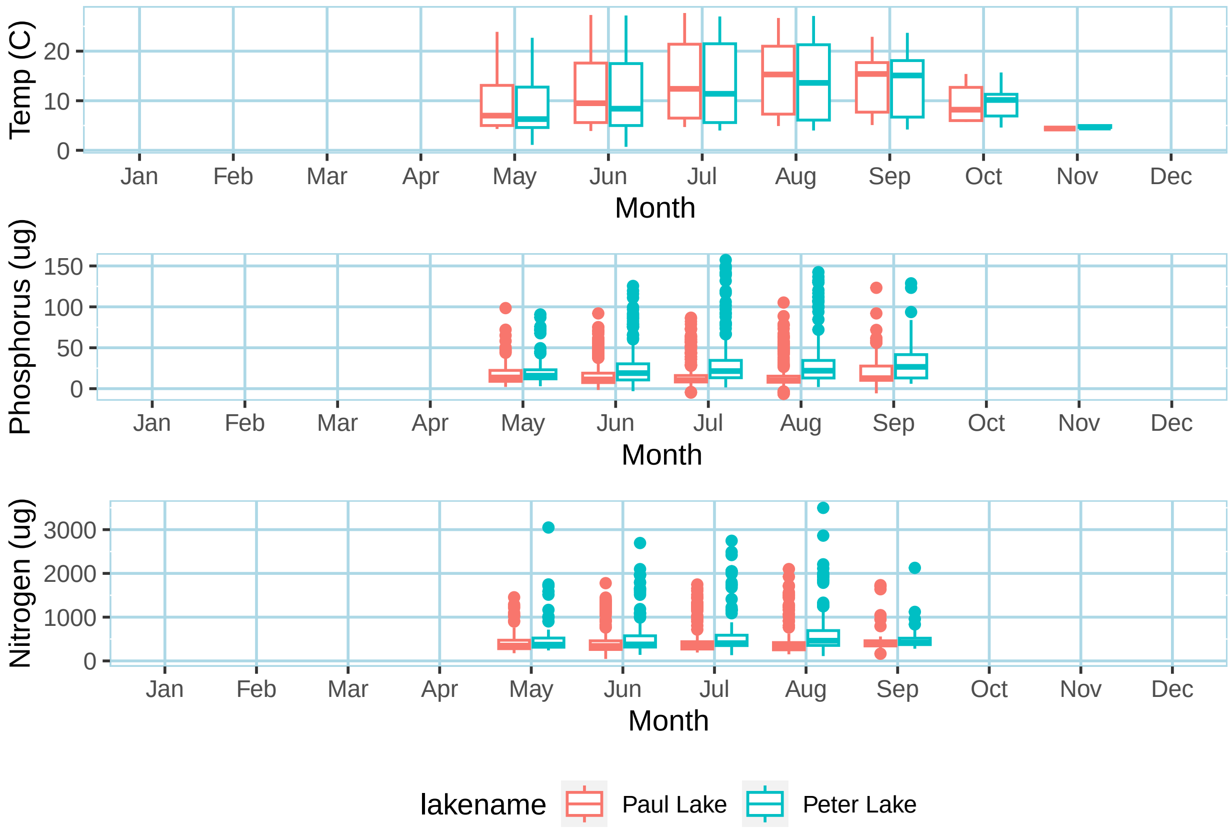
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```



```
#boxplot for TN using geom_boxplot()
#converted months to factor before setting to x axis, ensured months were in levels
  #1-12 for all 12 months, and labeled using their month abbreviations
#set color=lakename as an argument within ggplot aesthetics

combo_5 <- plot_grid(Plot_5_temp + theme(legend.position="none"),
                     Plot_5_TP + theme(legend.position="none"),
                     Plot_5_TN + theme(legend.position="bottom"),
                     nrow=3, align = 'h', rel_heights=c(1, 1, 1.5))
```

```
## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.

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

## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.

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

## Warning: Use of ‘NTL_LTF$month‘ is discouraged.
## i Use ‘month‘ instead.

## 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.
```

```
combo_5
```



```
#combined all three plots into one using plot_grid(), aligning them in one row horizontally
#removed repeating legends by using theme(legend.position="none")
#assigned to object combo_5

#legend_5 <- get_legend(Plot_5_temp + theme(legend.box.margin = margin(0, 0, 0, 20)))

#extracted legend from Plot_5_temp using get_legend() and assigned to object legend_5

#Plot_5 <- plot_grid(combo_5, legend_5, rel_widths = c(5, 0.5))
#Plot_5

#created final plot using plot_grid to combine combo_5 and legend_5 objects, set relative
#widths of graphs and legends using rel_widths at a ratio of 10:1
```

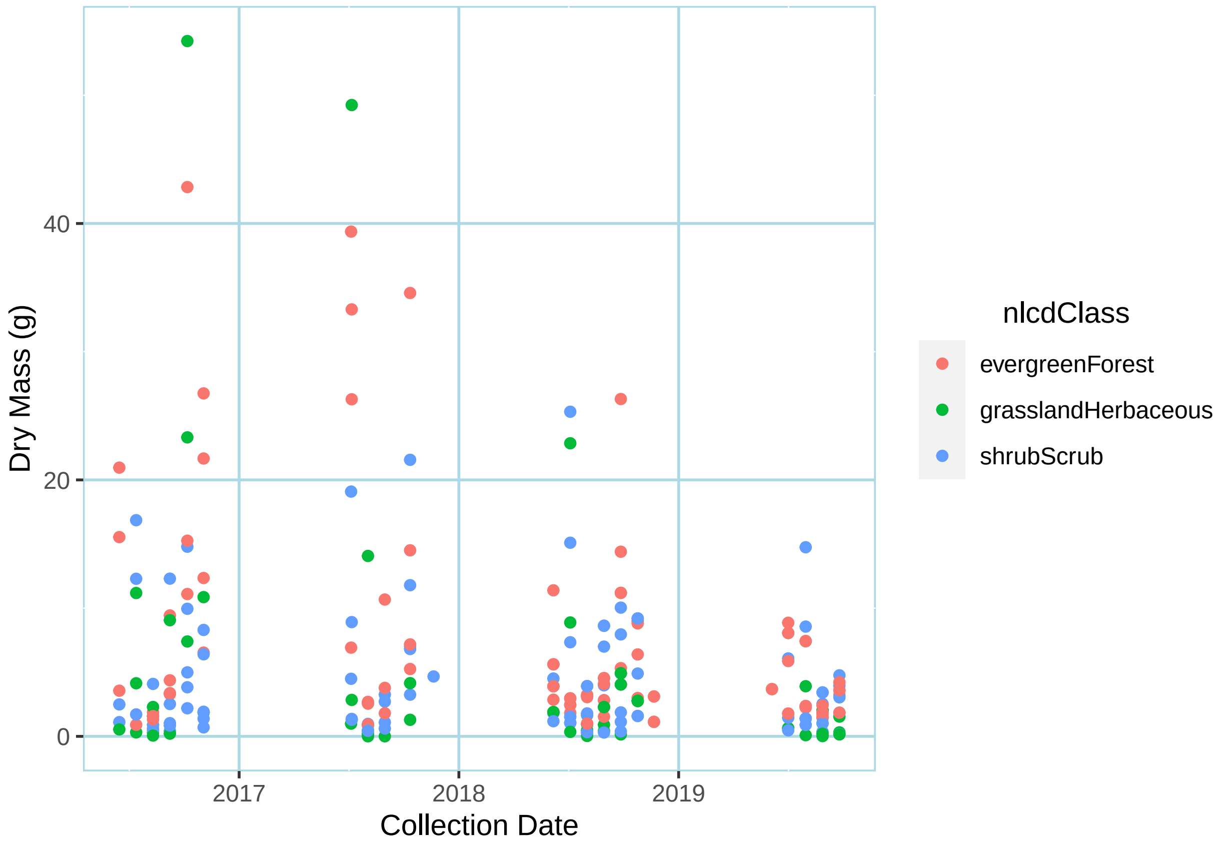
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The temperatures trend higher during the summer, with Paul Lake having higher average temperatures than Peter Lake. The trends for TP data across seasons diverged between the lakes: Peter Lake average TP trended higher during summer months while Paul Lake average TP trended lower during summer months. The distribution of TN values remained relatively

constant across all months, though Peter Lake retained slightly higher average TN values than Paul Lake. Both lakes had significant number of outliers in their TP and TN data across the months.

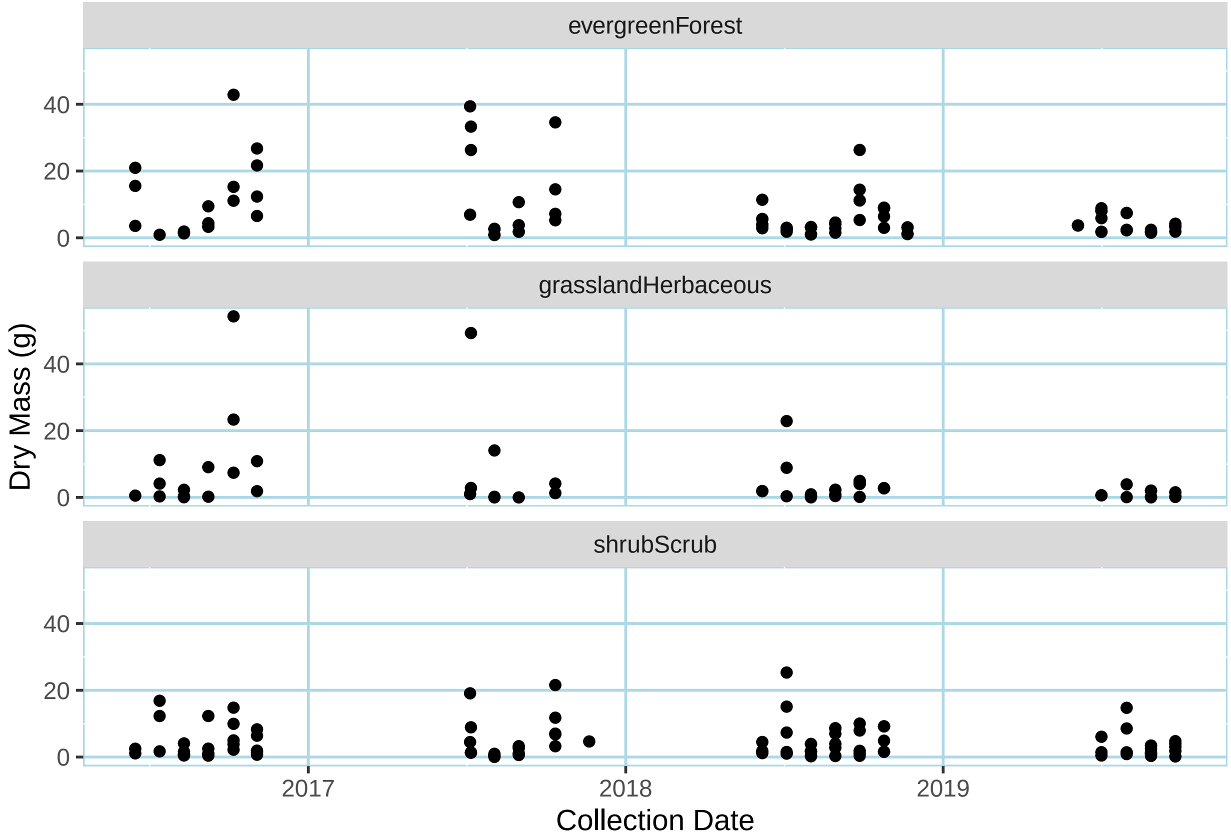
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

Plot_6 <- Niwot %>% filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point() +
  xlab("Collection Date") +
  ylab("Dry Mass (g)")

Plot_6
```



```
#filtered dataset for Needles functional group prior to applying ggplot()
#set color=nlcdClass as an argument within ggplot aesthetics
#used geom_point() to create scatter plot
```

```
#7

Plot_7 <-Niwot %>% filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate, y=dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow=3) +
  xlab("Collection Date") +
  ylab("Dry Mass (g)")

Plot_7
```



```
#filtered dataset for Needles functional group prior to applying ggplot()
#used geom_point() to create scatter plot
#facet_wrap() set nlcdClass as a variable within vars() to create three separate plots
  #based on nlcdClass, nrow=3 ensured plots were stacked on top of each other
```

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

> Answer: The facet plot (7) is more effective because the data points are more clearly defined. The color plot (6) overlays the data points over each other, so even though there is color labeling, we cannot see the full spread of data across all three classes.