Assignment 5: Data Visualization

Allison Barbaro

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
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: ’lubridate’
##
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
library(here)
```

```
## here() starts at /home/guest/R/EDA-Spring2023
```

```
library(cowplot)
```

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

```
library(ggplot2)

getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

```
Peter.Paul.nutrients <-
  read.csv("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
Niwo.Litter <-
  read.csv("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
Peter.Paul.nutrients$sampledate <-ymd(Peter.Paul.nutrients$sampledate)
Niwo.Litter$collectDate <-ymd(Niwo.Litter$collectDate)
```

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
my_theme.1 <-
  theme(
    line = element_line(
      color='pink',
      linewidth =2
    ),
    legend.background = element_rect(
      color='grey',
    ),
    legend.title = element_text(
      color='blue'
    )

  )

theme(my_theme.1)
```

```
## List of 1
##  $ line:List of 3
##   ..$ line             :List of 6
##   .. ..$ colour       : chr "pink"
##   .. ..$ linewidth    : num 2
##   .. ..$ linetype     : NULL
##   .. ..$ lineend      : NULL
##   .. ..$ arrow        : logi FALSE
##   .. ..$ inherit.blank: logi FALSE
##   .. ..- attr(*, "class")= chr [1:2] "element_line" "element"
##   ..$ legend.background:List of 5
##   .. ..$ fill         : NULL
##   .. ..$ colour       : chr "grey"
##   .. ..$ linewidth    : NULL
##   .. ..$ linetype     : NULL
##   .. ..$ inherit.blank: logi FALSE
##   .. ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##   ..$ legend.title     :List of 11
##   .. ..$ family       : NULL
##   .. ..$ face         : NULL
##   .. ..$ colour       : chr "blue"
##   .. ..$ size         : NULL
##   .. ..$ hjust        : NULL
##   .. ..$ vjust        : NULL
##   .. ..$ angle        : NULL
##   .. ..$ lineheight   : NULL
##   .. ..$ margin       : NULL
##   .. ..$ debug        : NULL
##   .. ..$ inherit.blank: logi FALSE
##   .. ..- attr(*, "class")= chr [1:2] "element_text" "element"
##   ..- attr(*, "class")= chr [1:2] "theme" "gg"
##   ..- attr(*, "complete")= logi FALSE
##   ..- attr(*, "validate")= logi TRUE
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

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()`).

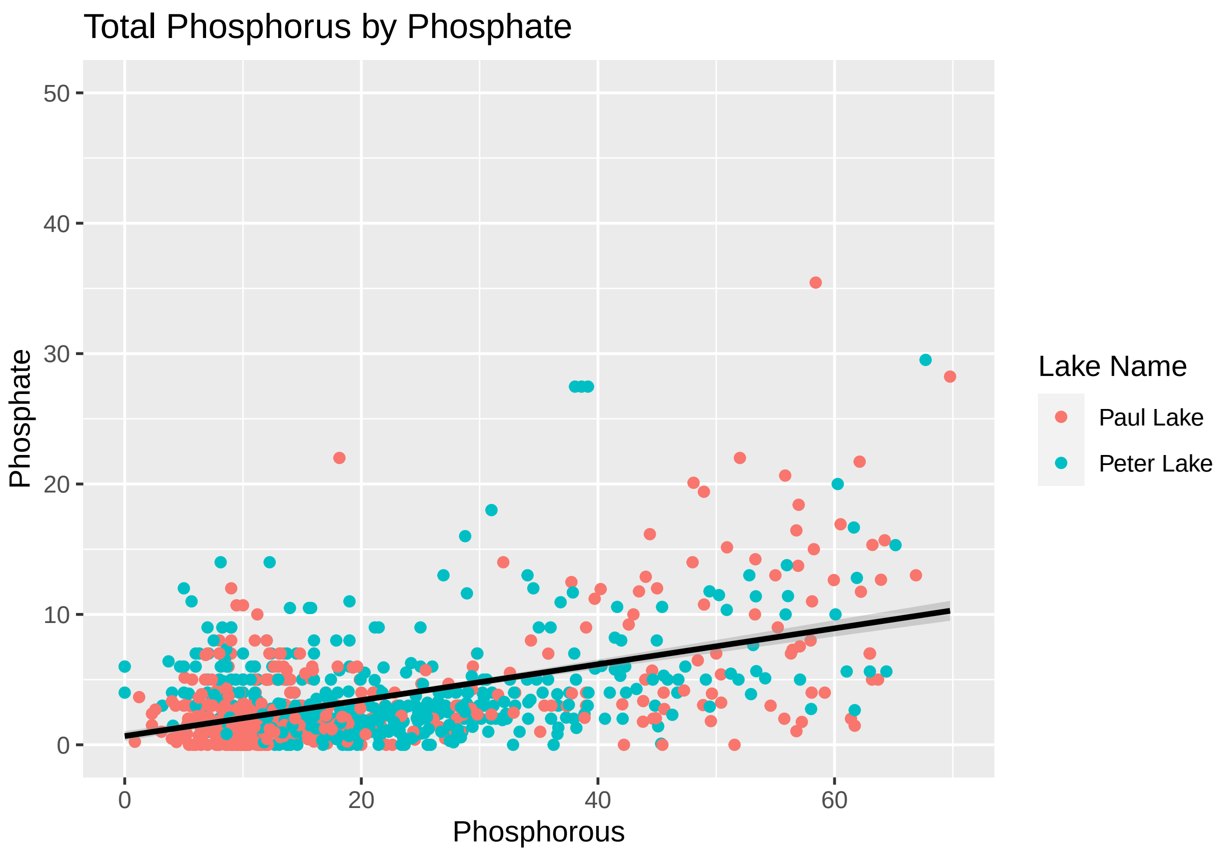
```
#4
Phos.plot.1 <- Peter.Paul.nutrients %>%
  ggplot(aes(
    x=tp_ug,
    y=po4,
    color=lakename
  )) +
  geom_point() +
   labs(title = "Total Phosphorus by Phosphate",
       x ="Phosphorous",
       y = "Phosphate") +
  geom_smooth(method=lm, color='black') +
  xlim(0, 70) +
  ylim(0, 50) +
  scale_color_discrete(name = "Lake Name")

print(Phos.plot.1)
```

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

```
## Warning: Removed 21996 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21996 rows containing missing values (`geom_point()`).
```



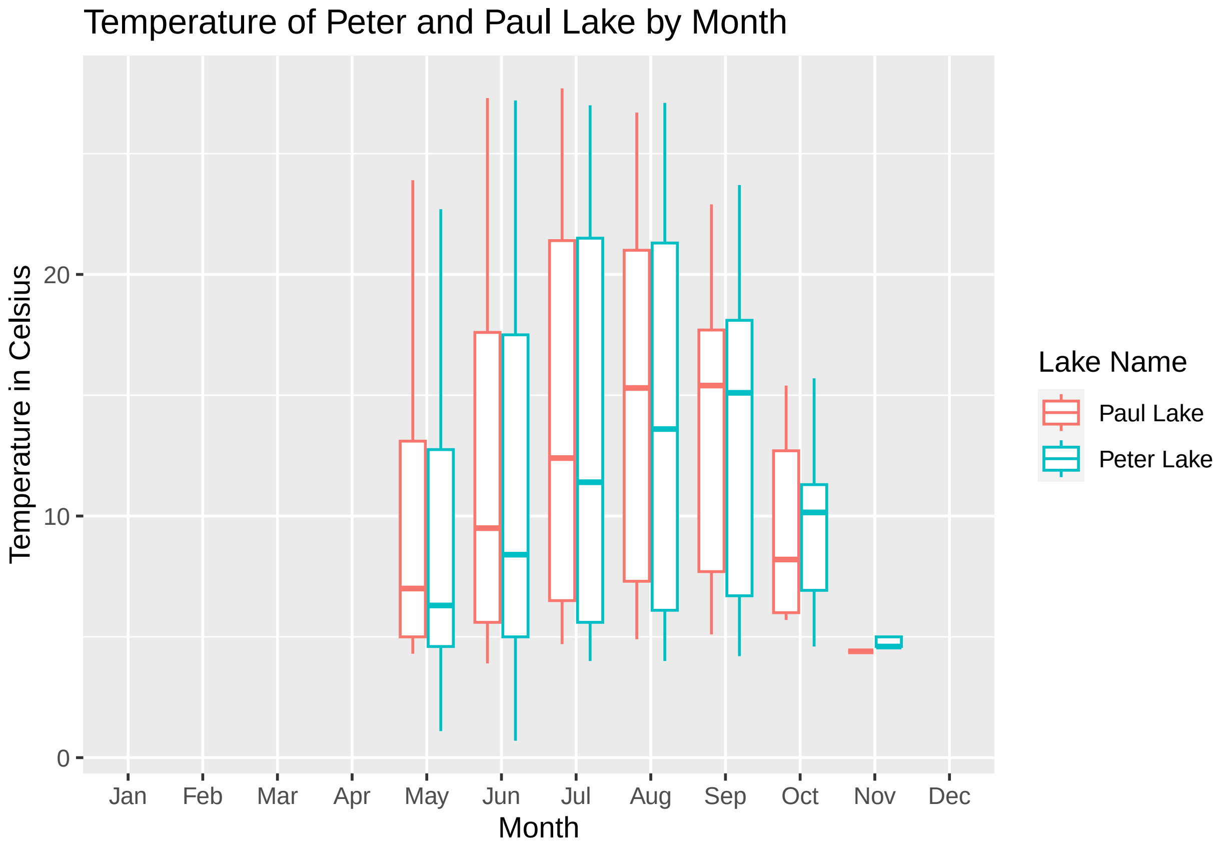
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: 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

Month.plot <- ggplot(Peter.Paul.nutrients) +
  geom_boxplot(aes(x= factor(month, levels=1:12, labels=month.abb), y = temperature_C, color = lakename
  labs(x = "Month", y = "Temperature in Celsius", title = "Temperature of Peter and Paul Lake by Month"
  scale_color_discrete(name = "Lake Name") + scale_x_discrete(drop=FALSE)
print(Month.plot)
```

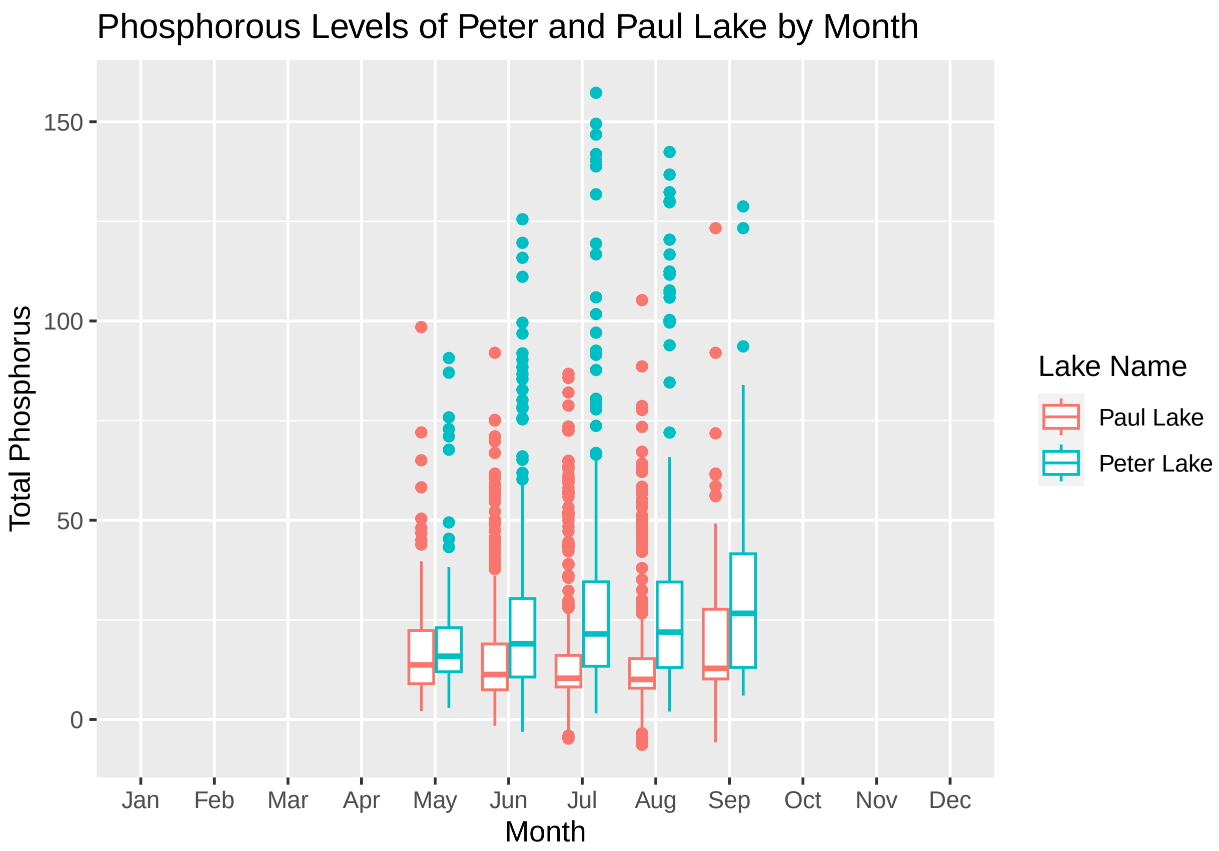
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



```
M.P <- Month.plot + theme(legend.position = "none")

Month.plot.1 <- ggplot(Peter.Paul.nutrients) +
  geom_boxplot(aes(x= factor(month, levels=1:12, labels=month.abb), y = tp_ug, color = lakename)) +
  labs(x = "Month", y = "Total Phosphorus", title = "Phosphorous Levels of Peter and Paul Lake by Month
  scale_color_discrete(name = "Lake Name") + scale_x_discrete(drop=FALSE)
print(Month.plot.1)
```

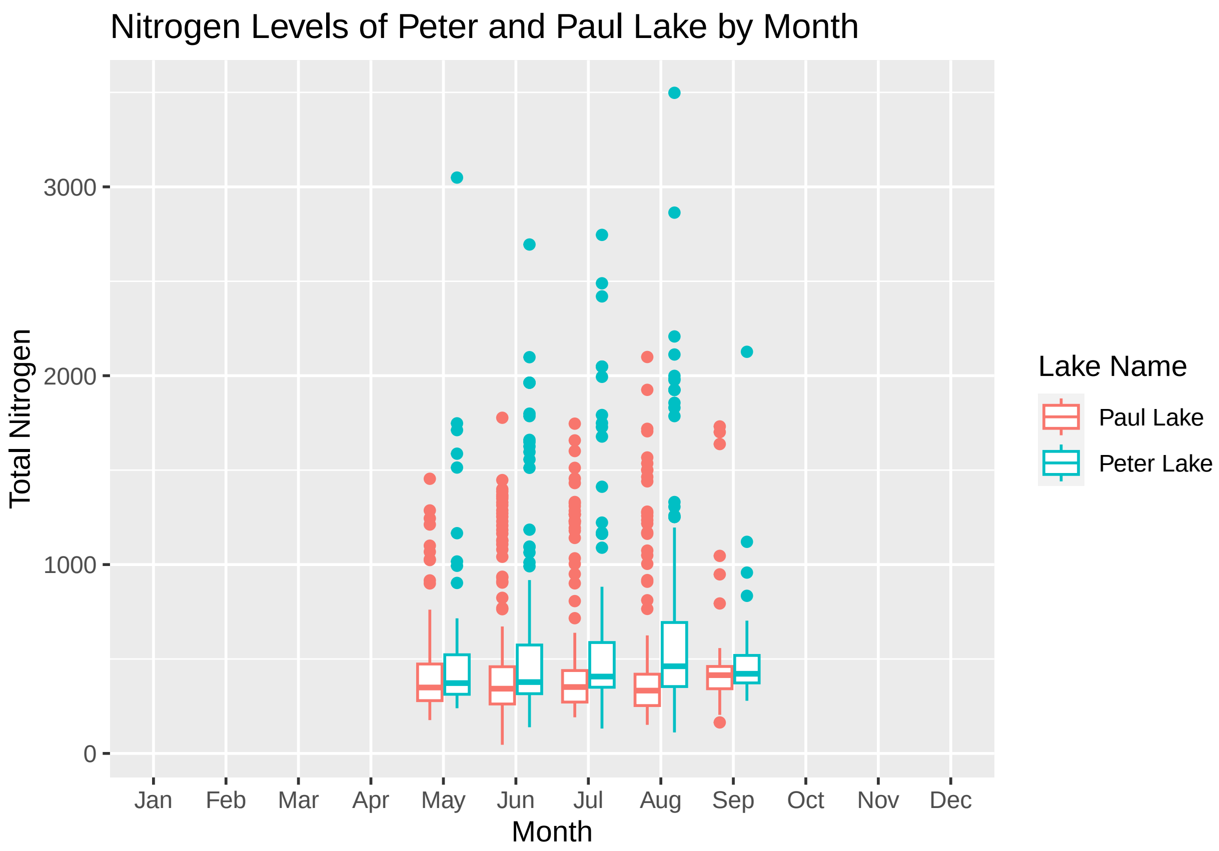
```
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```



```
M.P.1 <- Month.plot.1 + theme(legend.position = "none")

Month.plot.2 <- ggplot(Peter.Paul.nutrients) +
  geom_boxplot(aes(x= factor(month, levels=1:12, labels=month.abb), y = tn_ug, color = lakename)) + labs
  scale_color_discrete(name = "Lake Name") + scale_x_discrete(drop=FALSE)
print(Month.plot.2)
```

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



```
M.P.2 <- Month.plot.2 + theme(legend.position = "none")

legend <- get_legend(Month.plot.1)
```

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

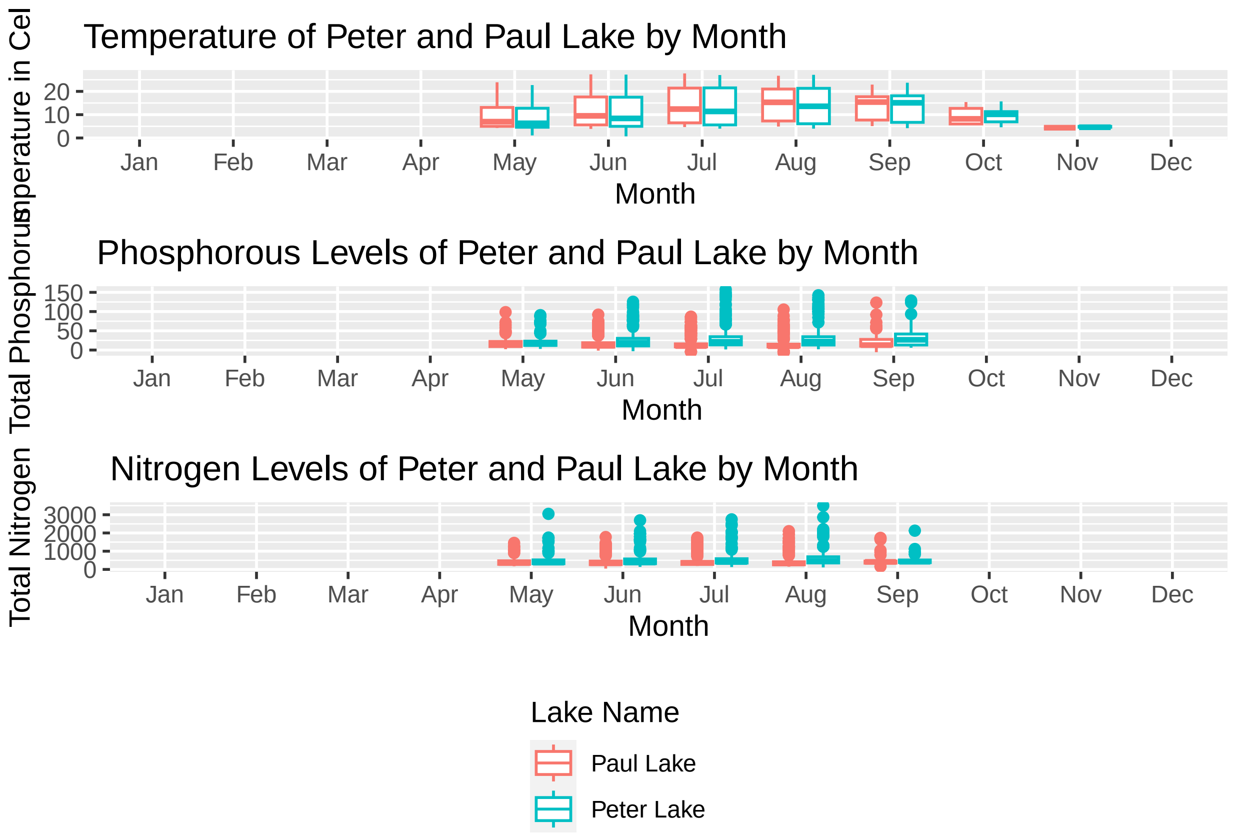
```
cowplot.main <- plot_grid(M.P, M.P.1, M.P.2, legend, ncol = 1)
```

```
## 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()‘).
```

```
print(cowplot.main)
```



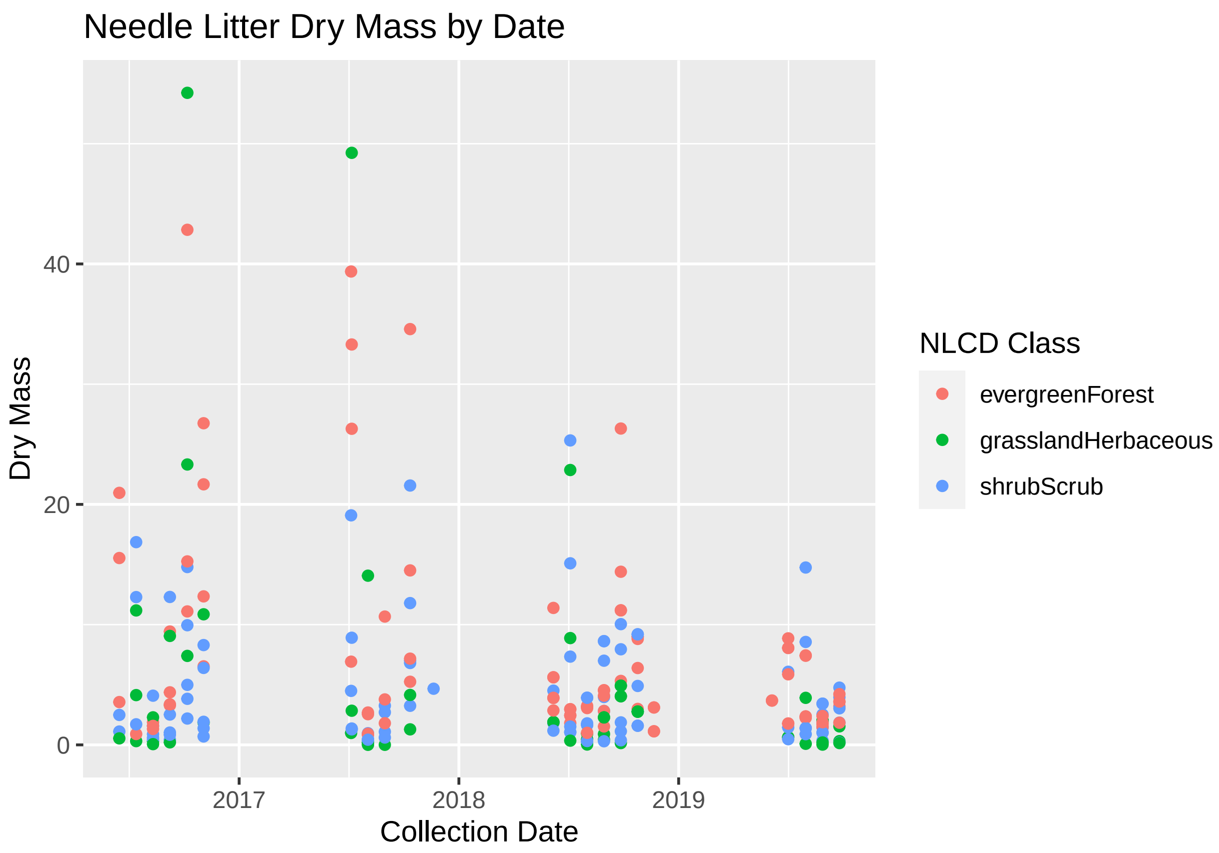
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The variables are most present during the summer months, and largely absent during winter. Temperature is the variable that is most similar between the two lakes, while P and N are both more present in Peter Lake than Paul Lake.

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.

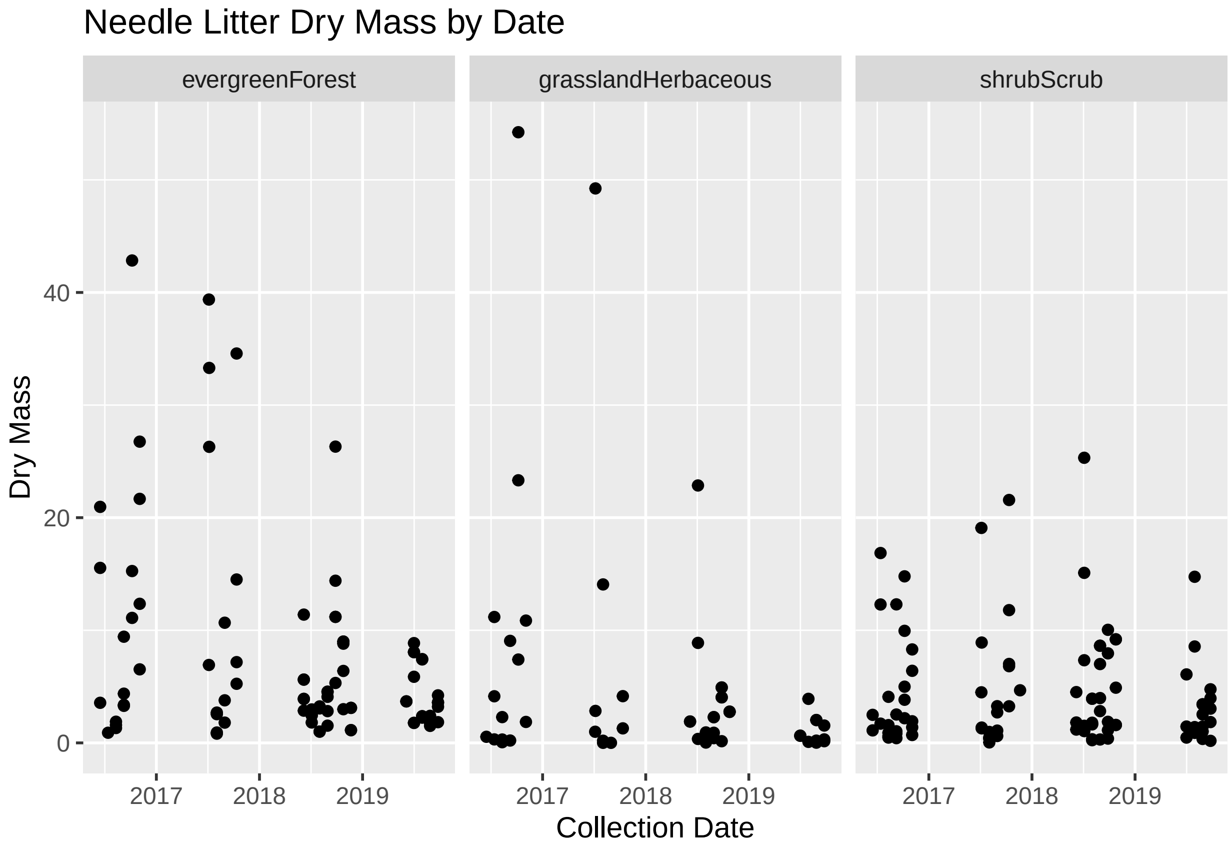
```
#6
Needles.Plot <- Niwo.Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(x = "Collection Date", y = "Dry Mass", title = "Needle Litter Dry Mass by Date") + scale_color_di

print(Needles.Plot)
```



```
#7
Needles.Plot.facets <- Niwo.Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  geom_point() +
  labs(x = "Collection Date", y = "Dry Mass", title = "Needle Litter Dry Mass by Date") +   scale_color
  facet_wrap(vars(nlcdClass))

print(Needles.Plot.facets)
```



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

> Answer: Plot 7 is more effective because the different NLCD classes are more clearly separated. Plot 6 there is too much overlap of data.