Assignment 5: Data Visualization

Annabelle White

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) # Loading packages
```

```
## -- 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()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(here)
```

```
## here() starts at C:/Users/ardwh/OneDrive/Documents/NSOE/env872/EDA-Spring2023
```

```
library(cowplot)
```

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

```
library(ggplot2)
library(ggthemes)
```

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

```
here()
```

```
## [1] "C:/Users/ardwh/OneDrive/Documents/NSOE/env872/EDA-Spring2023"
```

```
Lake <- read.csv(
  file = here('Data','Processed',
              'NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
  stringsAsFactors = T
)

Litter <- read.csv(
  file = here('Data','Processed','NEON_NIWO_Litter_mass_trap_Processed.csv'),
  stringsAsFactors = T
)

#2
# Check date fields
class(Lake$sampledate) # Both are read as factors
```

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

```
class(Litter$collectDate) #year, month, day
```

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

```
# Convert format to dates
Lake$sampledate <- ymd(Lake$sampledate)
Litter$collectDate <- ymd(Litter$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

```
#3

starwars <- theme_base() +
  theme( # Pulled from John's theme template code
    line = element_line(color = "white"),
    rect = element_rect(fill = "black"),
    text = element_text(color = "yellow"),

    # Modified inheritance structure of text element
    #plot.title = element_text(color = "yellow"),
    #axis.title.x =      element_blank(),
    #axis.title.y =      element_blank(),
    #axis.text = element_text(color = "yellow"),

    # Modified inheritance structure of line element
    axis.ticks = element_line(color = "yellow"),
    panel.grid.major = element_line(color = "yellow",
                                    linetype = "dashed"),

    # Modified inheritance structure of rect element
    #plot.background = element_rect(fill = "black"),
    #panel.background = element_rect(fill = "black")
    #legend.key =        element_rect(),

    # Modifiying legend.position
    #legend.position = 'right',

    #complete = TRUE
    )
```

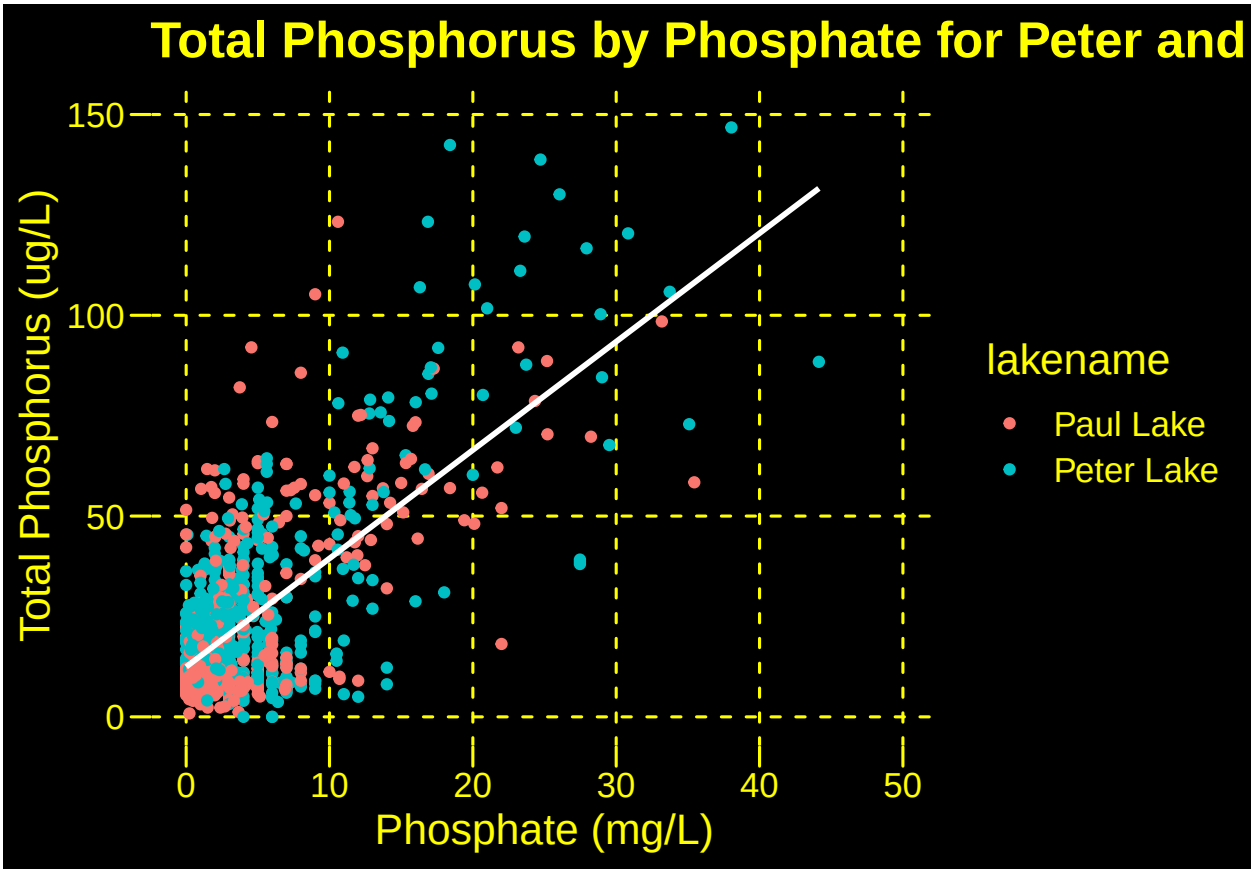
Create graphs

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

Pplot <- ggplot(Lake, aes(x = po4,
                          y = tp_ug,
                          color = lakename)) +
  geom_point() +
  starwars + # Add my theme

  # Add a line of best fit colored white
  stat_smooth(method = "lm",
              formula = y ~ x,
              se = FALSE,
              na.rm = TRUE,
              color = "white") +

  # Adjust axes limits to hide extreme values
  xlim(0, 50) +
  ylim(0, 150) +

  # Add labels and titles
  xlab("Phosphate (mg/L)") +
  ylab("Total Phosphorus (ug/L)") +
  ggtitle("Total Phosphorus by Phosphate for Peter and Paul Lakes")
print(Pplot)
```

```
## Warning: Removed 21948 rows containing missing values (‘geom_point()‘).
```

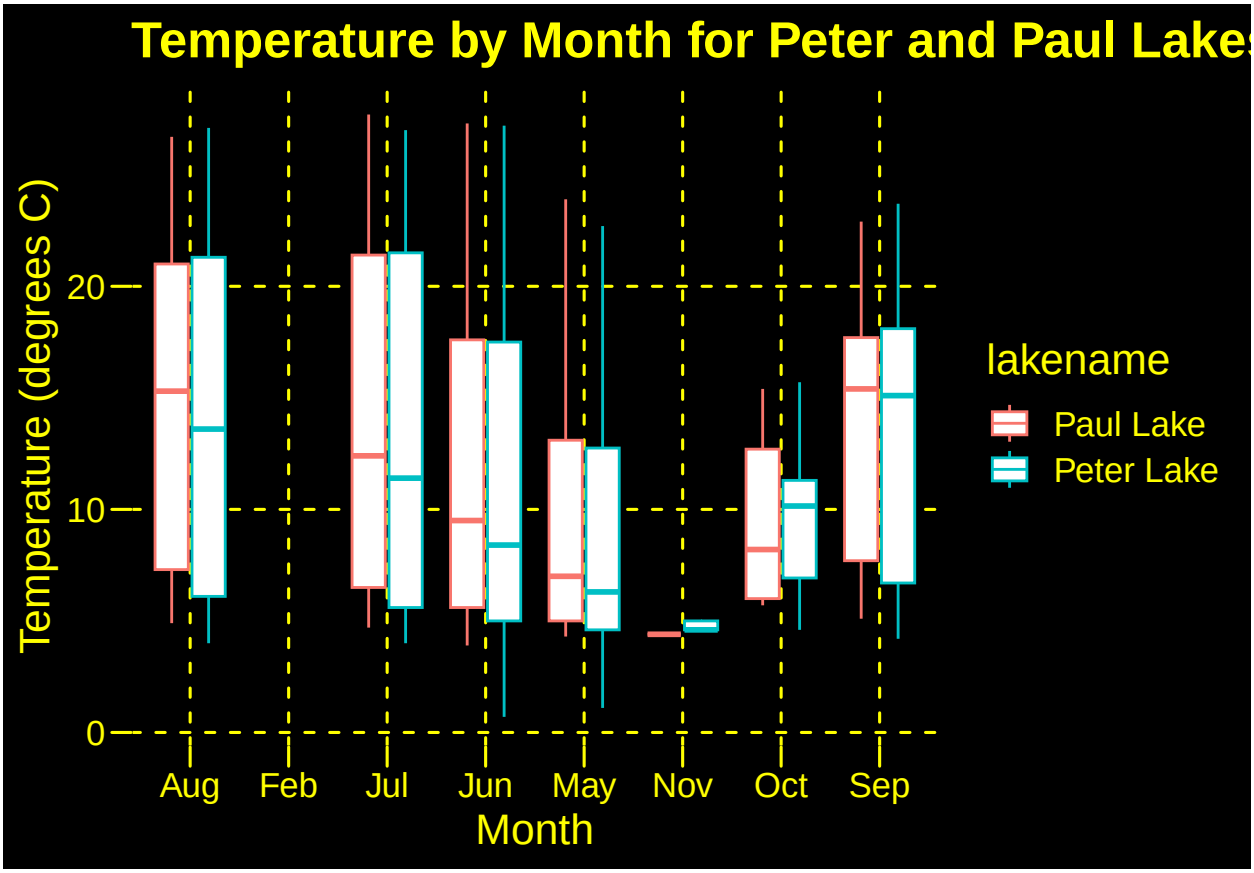


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

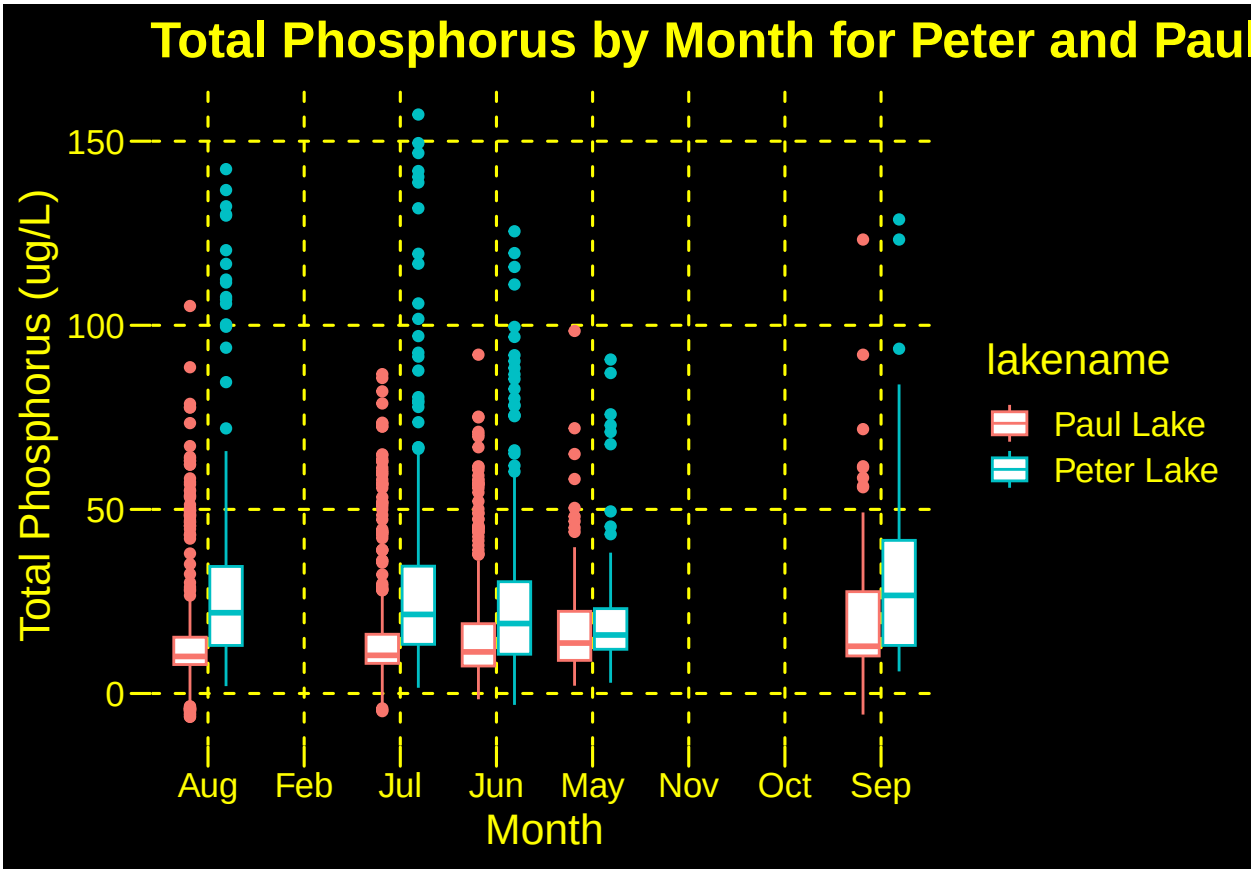
```
#5

# Separate boxplots

Tplot <- ggplot(Lake, aes(x = month.abb[month],
                          y = temperature_C,
                          color = lakename)) +
  geom_boxplot() +
  starwars + # Add my theme

  # Add labels and titles
  xlab("Month") +
  ylab("Temperature (degrees C)") +
  ggtitle("Temperature by Month for Peter and Paul Lakes")
print(Tplot)
```

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



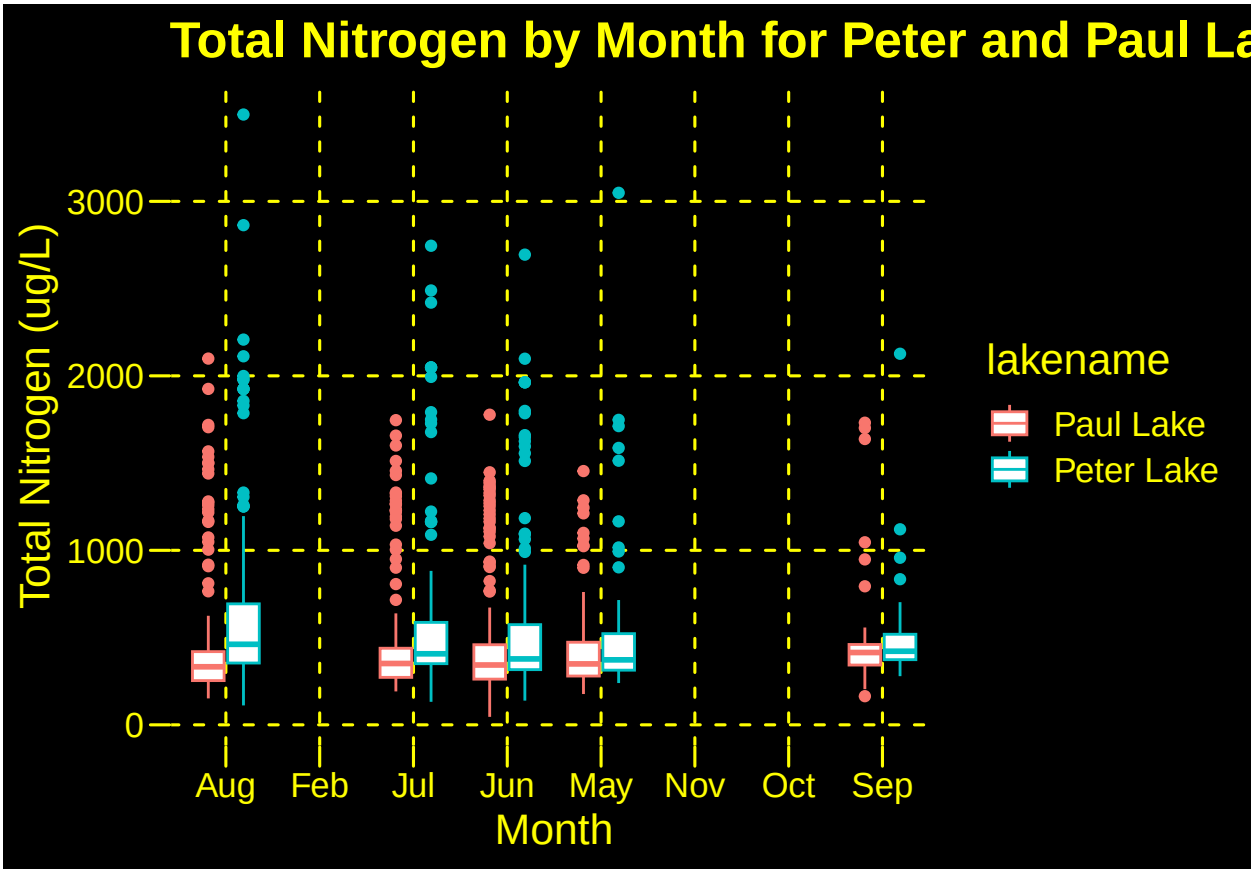

```
TPplot <- ggplot(Lake, aes(x = month.abb[month],
                           y = tp_ug,
                           color = lakename)) +
  geom_boxplot() +
  starwars + # Add my theme

  # Add labels and titles
  xlab("Month") +
  ylab("Total Phosphorus (ug/L)") +
  ggtitle("Total Phosphorus by Month for Peter and Paul Lakes")
print(TPplot)
```

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



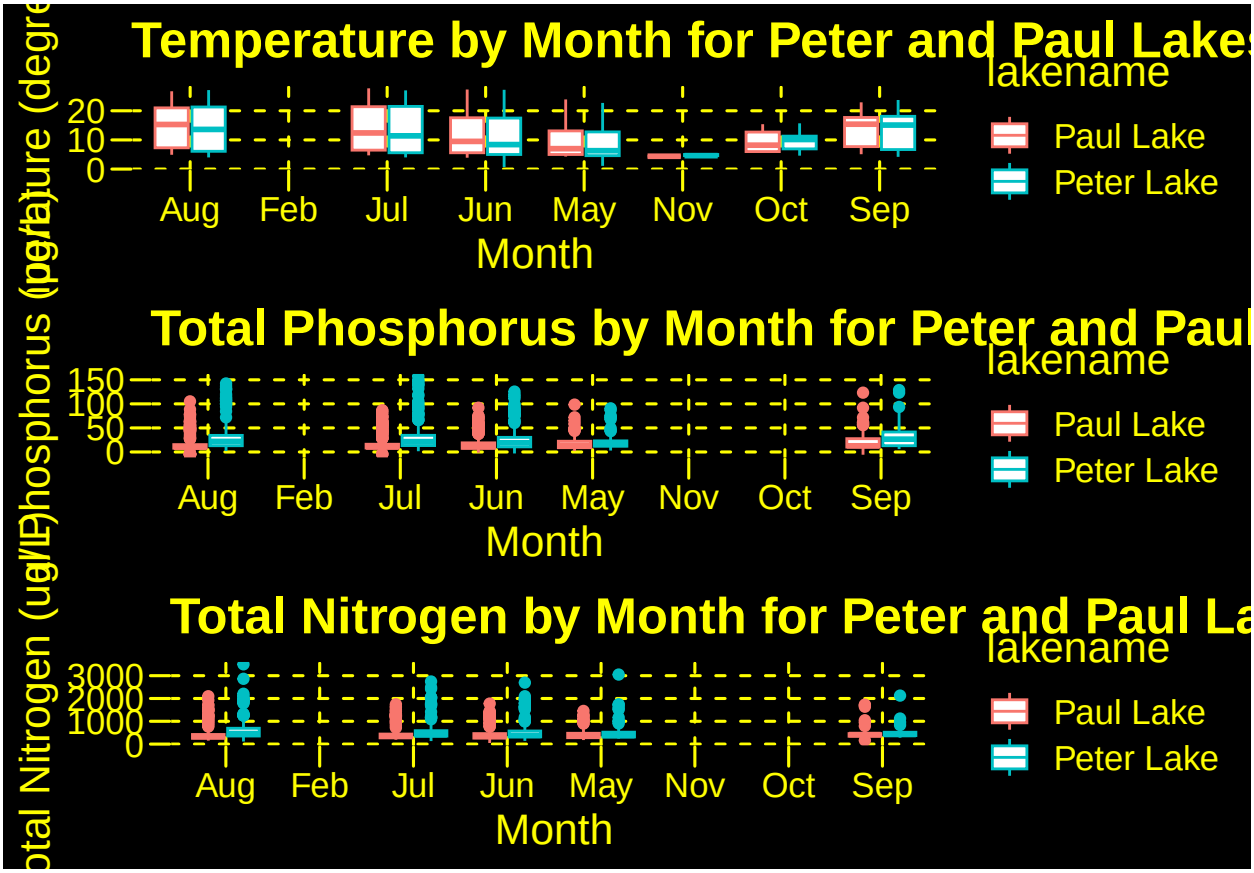

```
TNplot <- ggplot(Lake, aes(x = month.abb[month],
                           y = tn_ug,
                           color = lakename)) +
  geom_boxplot() +
  starwars + # Add my theme

  # Add labels and titles
  xlab("Month") +
  ylab("Total Nitrogen (ug/L)") +
  ggtitle("Total Nitrogen by Month for Peter and Paul Lakes")
print(TNplot)
```

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




```
# Cowplot

plot_grid(Tplot, TPplot, TNplot, nrow = 3, align = 'h')
```

```
## 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()‘).
```




```
# I didn't remove the legends. sue me
```

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

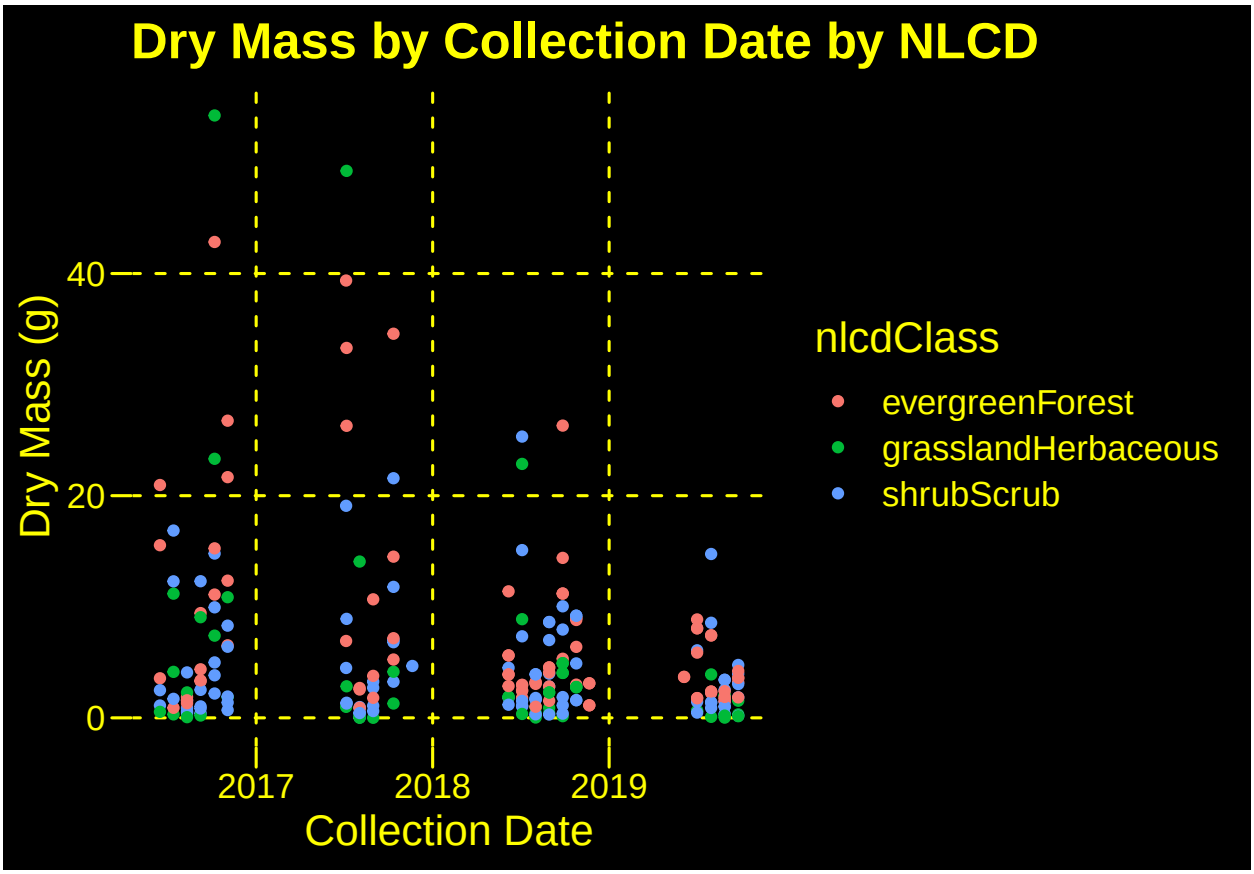
> Answer: These variables decrease from the summer into the winter, reaching lows in October and November. The nutrient variables are generally higher in Peter Lake than Paul Lake. With regards to temperature, Peter Lake has higher highs and lower lows than Paul Lake.

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.

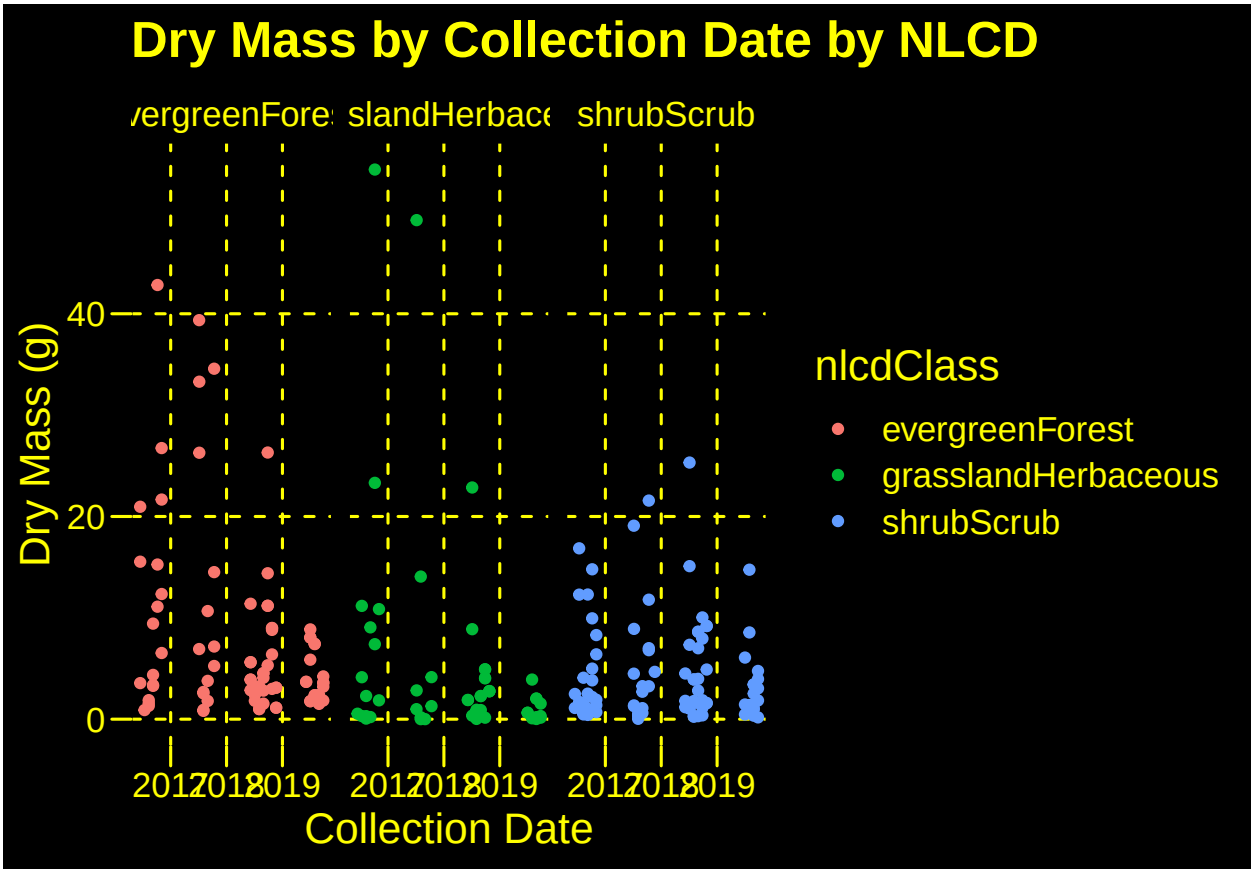
```
#6

NLCD_color <- Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(
      x = collectDate,
      y = dryMass,
      color = nlcdClass)
      ) +
  starwars + # Add my theme!
```

```
  geom_point() +
  #facet_wrap(vars(lakename)) +
  labs(
    title = "Dry Mass by Collection Date by NLCD",
    y = "Dry Mass (g)",
    x = "Collection Date"
  )
print(NLCD_color)
```




```
#7

NLCD_facet <- Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(
      x = collectDate,
      y = dryMass,
      color = nlcdClass)
      ) +
  starwars + # Add my theme!
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(
    title = "Dry Mass by Collection Date by NLCD",
    y = "Dry Mass (g)",
    x = "Collection Date"
```

```
  )
print(NLCD_facet)
```



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

Answer: Plot 7 (faceted) is more visually effective for me. When the points are jumbled altogether in plot 6, it's easy to see an overall downward trend, but that's it. Grouping the NLCD classes separately allows me to track variation over time within each class, including averages and outliers.