Assignment 5: Data Visualization

Brynn Rotbart

Fall 2024

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)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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(cowplot)
```

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

```r
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```r
library(ggplot2)

getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

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

Niwot_Ridge_Litter <- read.csv(
  file=here(
    "Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)
#2
class(Peter_Paul_chem$sampledate)
```

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

```r
class(Niwot_Ridge_Litter$collectDate)
```

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

```r
Peter_Paul_chem <- Peter_Paul_chem %>% mutate(sampledate = ymd(sampledate))
Niwot_Ridge_Litter <-Niwot_Ridge_Litter %>% mutate(
  collectDate= ymd(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
my_super_awesome_theme <- theme(
plot.background = element_rect("#9CCAC6"),
plot.title= element_text("#516ED2"),
)

theme_set(my_super_awesome_theme)
```

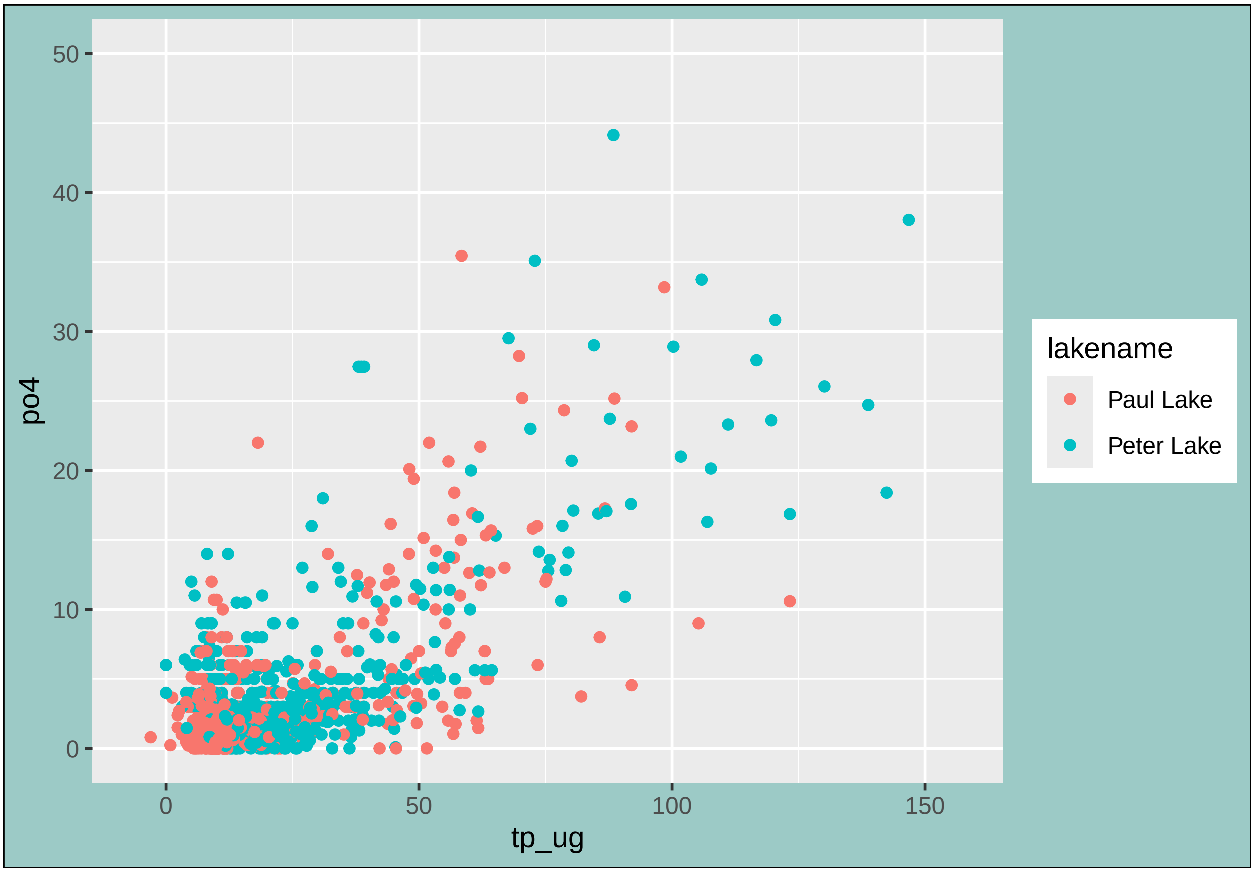
Create graphs

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 line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Phos_phate_graph <-
  ggplot(Peter_Paul_chem) +
  geom_point(aes(
    x=tp_ug,
    y=po4,
    color=lakename))+
    ylim(0,50)
print(Phos_phate_graph)
```

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



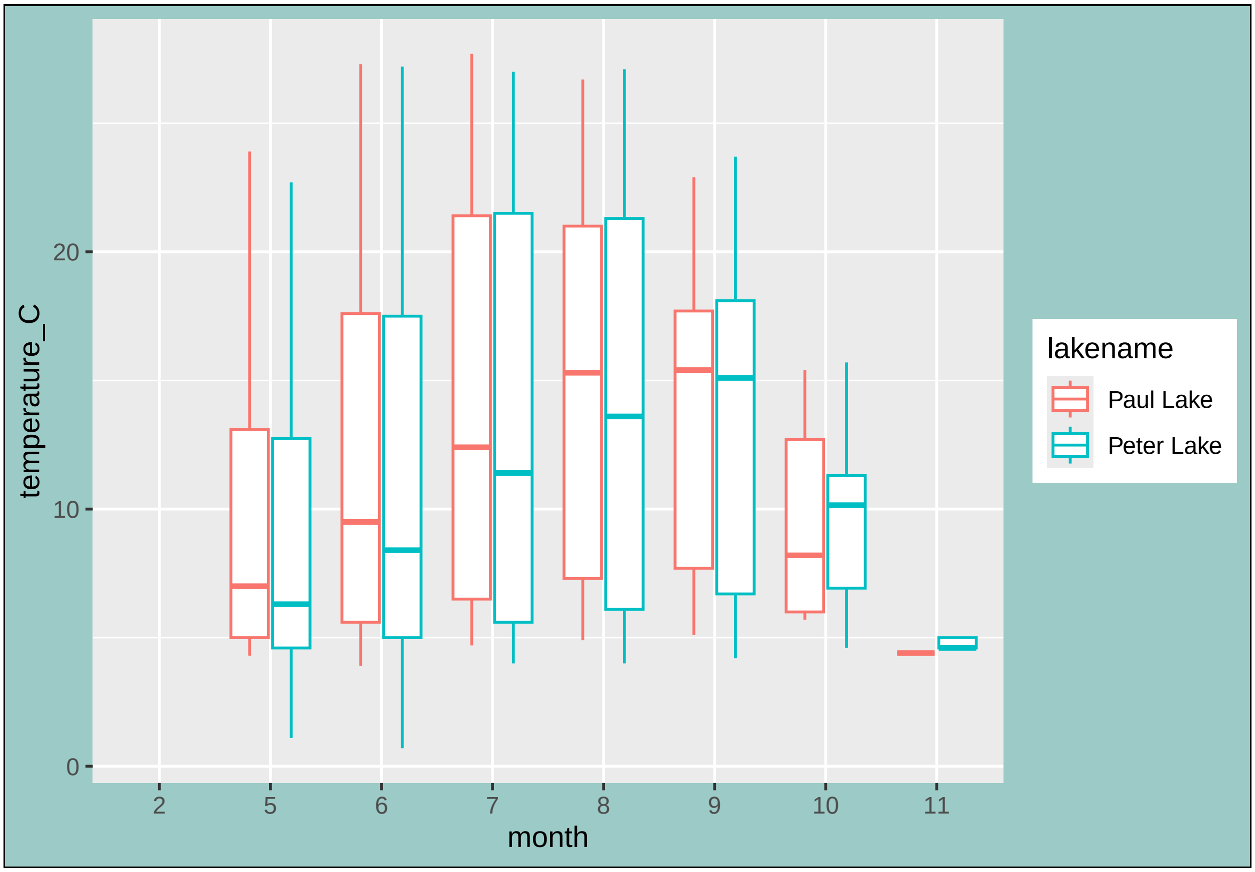
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
Peter_Paul_chem$month <- factor(Peter_Paul_chem$month)
Peter_Paul_chem_Temp <-
  ggplot(Peter_Paul_chem)+
  geom_boxplot(aes(
    x=month,
    y=temperature_C,
    color=lakename
  ))

print(Peter_Paul_chem_Temp)
```

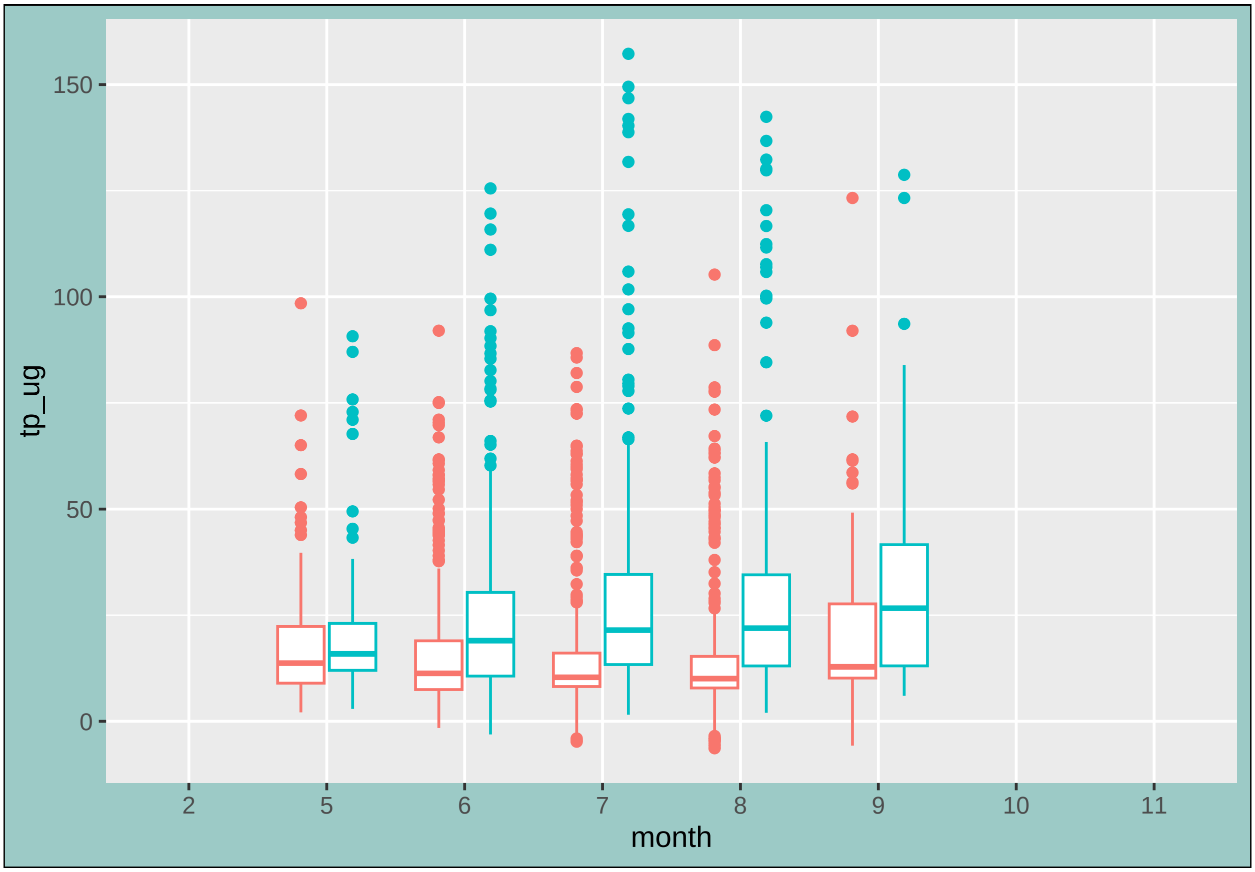
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```r
Peter_Paul_chem_Tp <-
  ggplot(Peter_Paul_chem)+
  geom_boxplot(aes(
    x=month,
    y=tp_ug,
    color=lakename
  ))+
  theme(legend.position = "none")

print(Peter_Paul_chem_Tp)
```

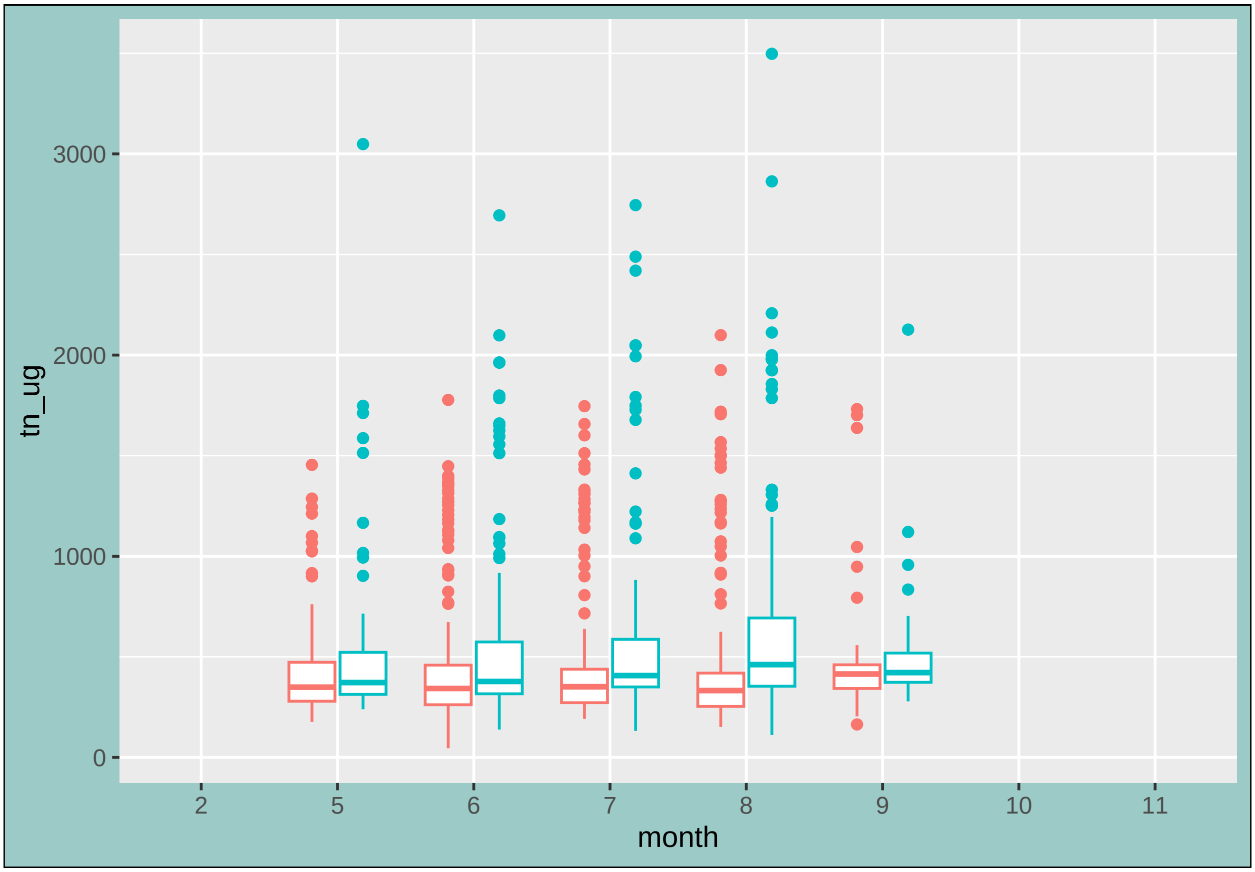
```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
Peter_Paul_chem_Tn <-
  ggplot(Peter_Paul_chem)+
  geom_boxplot(aes(
    x=month,
    y=tn_ug,
    color=lakename)
    )+
  theme(legend.position = "none")

print(Peter_Paul_chem_Tn)
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



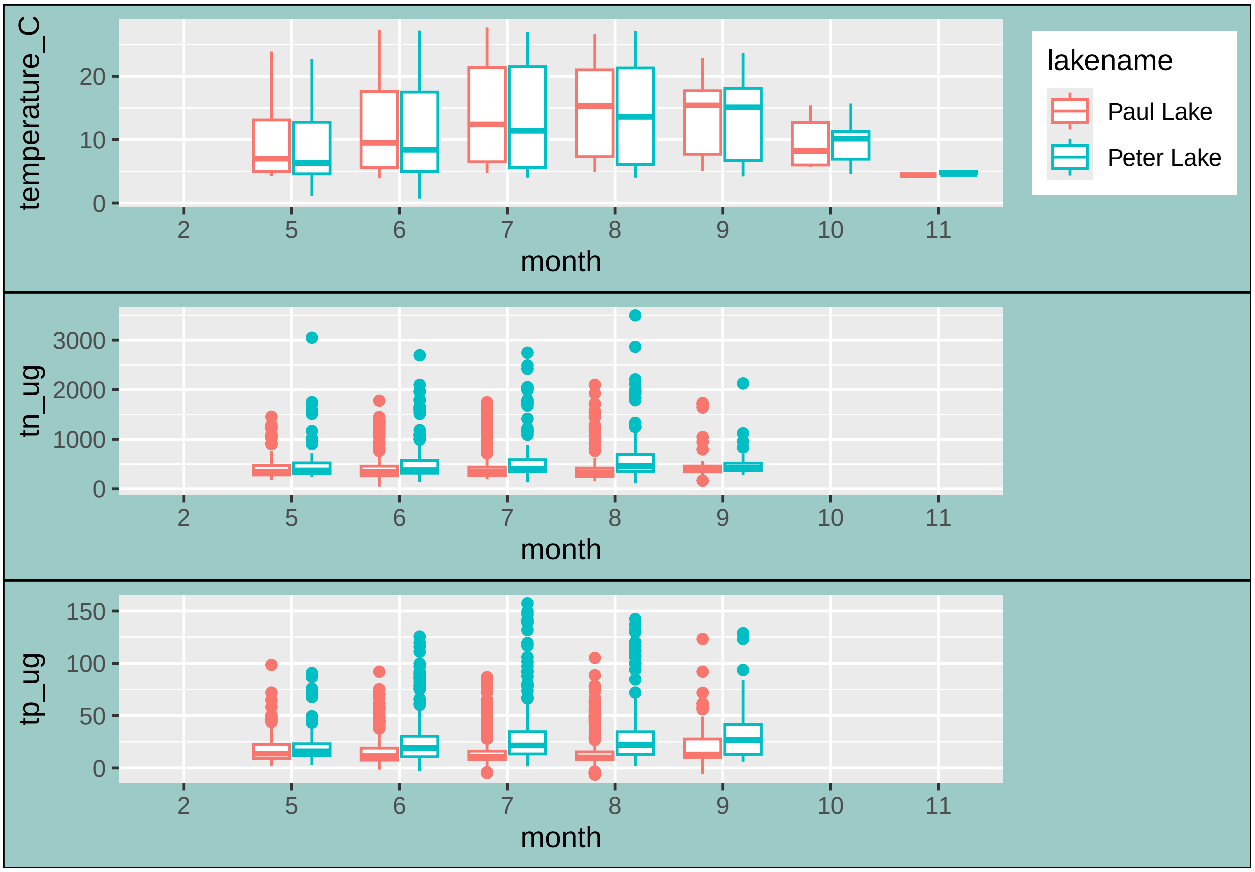

```r
peter_paulplot_grid<- plot_grid(
  Peter_Paul_chem_Temp, Peter_Paul_chem_Tn, Peter_Paul_chem_Tp,
          nrow=3,
          align='v')
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
print(peter_paulplot_grid)
```



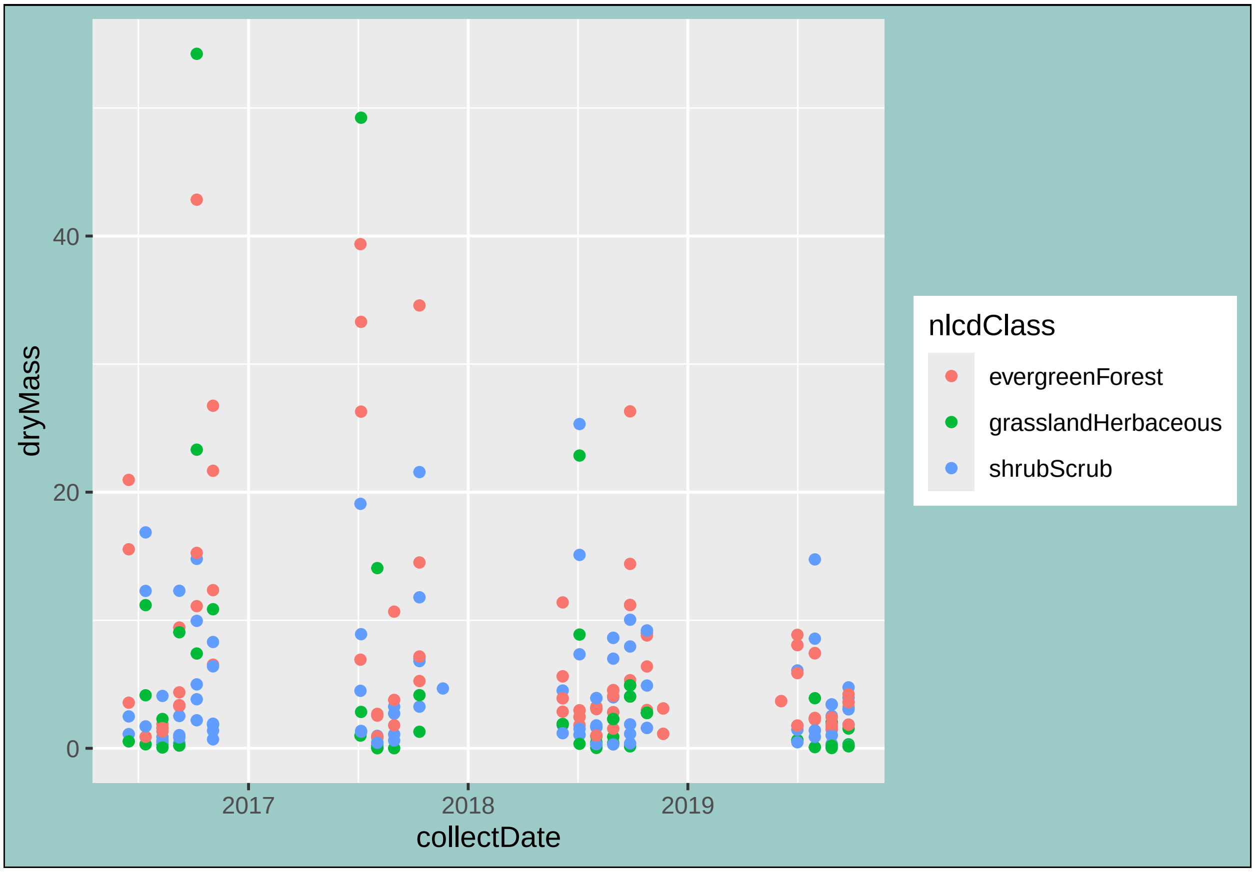
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Both the temperature and the nitrogen and phopshorous load increase over the summer, as expected. This makes sense because temperature increases in the summer, there is more sunlight, therefore more photosynthesis, higher algal blooms and more bioecological activity from marine animals. The temperature has a wider range in the summer, but the variance between nitrogen and phosphorous loads seems more discrete, not changing much from other measurements. However, both measurements seems to completely absent in the colder months. Peter has far more outliers than Paul lake when it comes to the Redfield Ratio and N & P measurements as well.

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.

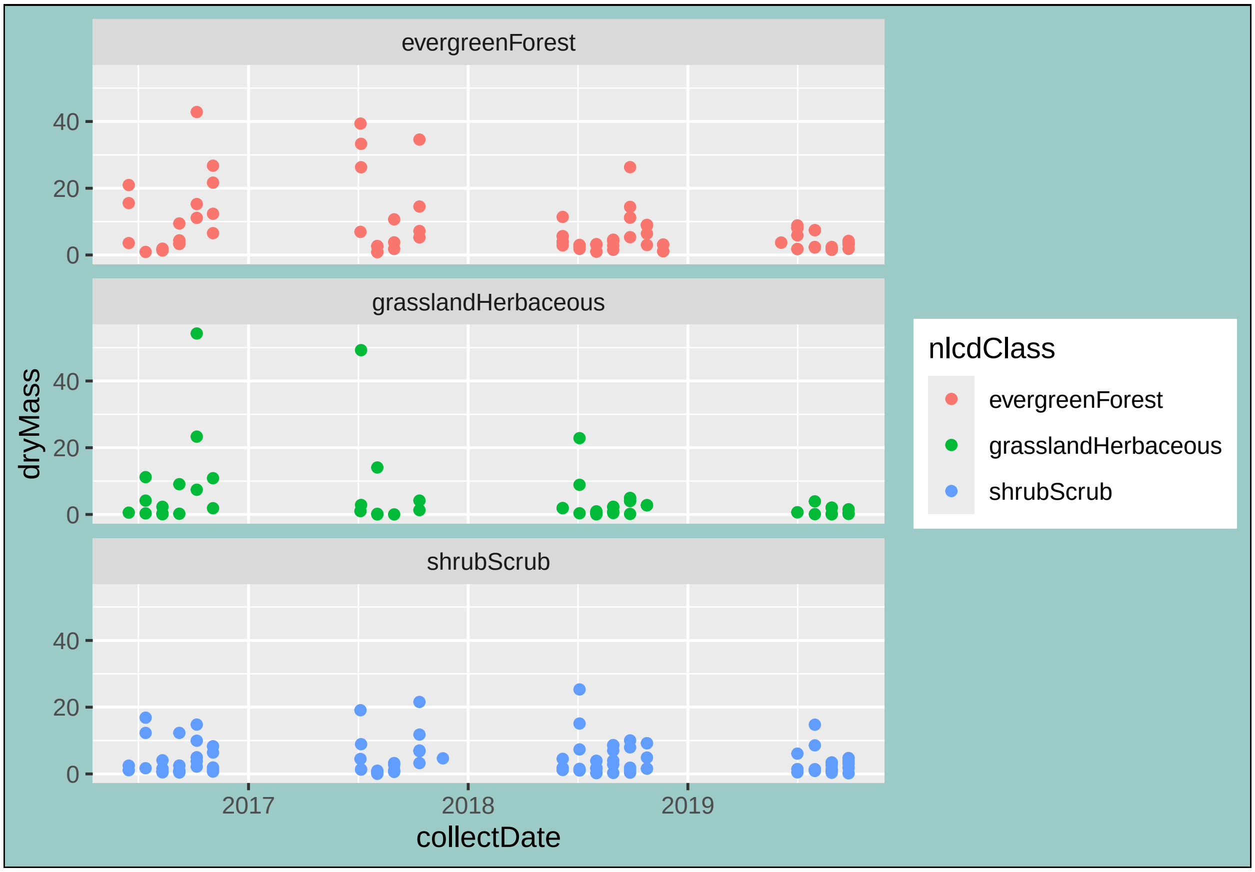
```
#6
dry_mass_needle <-
  Niwot_Ridge_Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot()+
  geom_point(aes(
    x=collectDate,
    y=dryMass,
```

```
    color=nlcdClass
  ))
print(dry_mass_needle)
```




```
#7
dry_mass_needle_facet <-
  Niwot_Ridge_Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(
    x=collectDate,
    y=dryMass,
    color=nlcdClass
    ))+
  geom_point()+
    facet_wrap(
      facet=vars(nlcdClass),
      nrow=3, ncol=1)

print(dry_mass_needle_facet)
```



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

Answer: Plot 7 is more effective because you can differenciate between the NLCD classes far easier and the dots do not overlap one another. If we are trying to measure the dry mass of needles, it makes more sense to seperate them by NLCD class in order to measure them effectively in years to come and compare results across ecosystem types.