Assignment 5: Data Visualization

Vincient Whatley

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);library(lubridate);library(here)
```

```
## -- 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 1.9.3     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 errors
## here() starts at /Users/vincient/Desktop/Projects/EDA_class
```

```
library(ggridges)
here()
```

```
## [1] "/Users/vincient/Desktop/Projects/EDA_class"
```

```
getwd()
```

```
## [1] "/Users/vincient/Desktop/Projects/EDA_class"
```

```
###
PeterPaul.chem.nutrients <-
  read.csv(here("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"), stri
NeonNiwo.LiterMass <-
  read.csv(here("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),stringsAsFactors = TRU

#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
NeonNiwo.LiterMass$collectDate <- ymd(NeonNiwo.LiterMass$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
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "azure")+
  theme(ggtitle("Your Title Here"))+
  theme(labs(y= "y axis name", x = "x axis name")),
        legend.position = "top") #alternative: legend.position + legend.justification
```

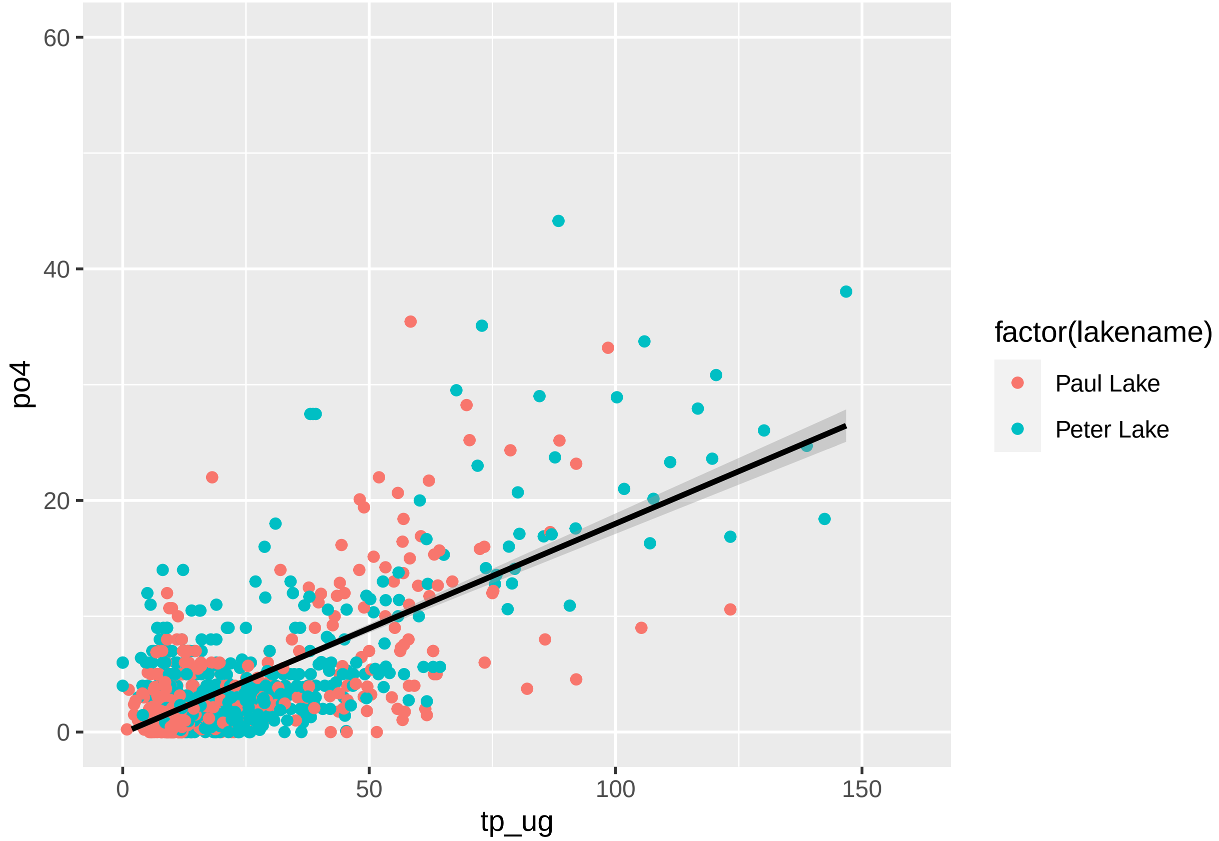
Create graphs

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
ggplot(PeterPaul.chem.nutrients, aes(x=tp_ug, y=po4, color= factor(lakename)))+
         xlim(0,160)+
         ylim(0,60)+
  geom_point()+
  geom_smooth(method= "lm", formula = y ~ x,, color = "black")
```

```
## Warning: Removed 21948 rows containing non-finite values (‘stat_smooth()‘).

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

## Warning: Removed 1 rows containing missing values (‘geom_smooth()‘).
```

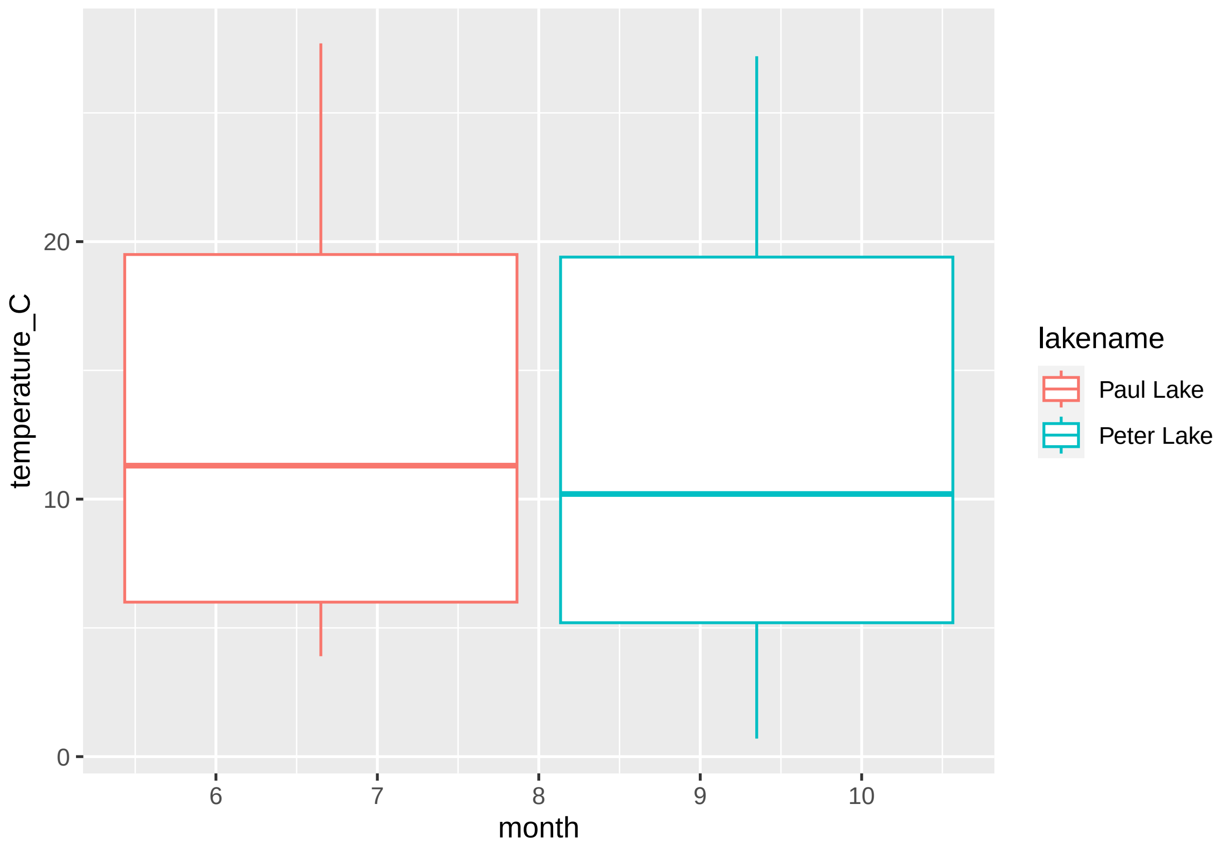


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * 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

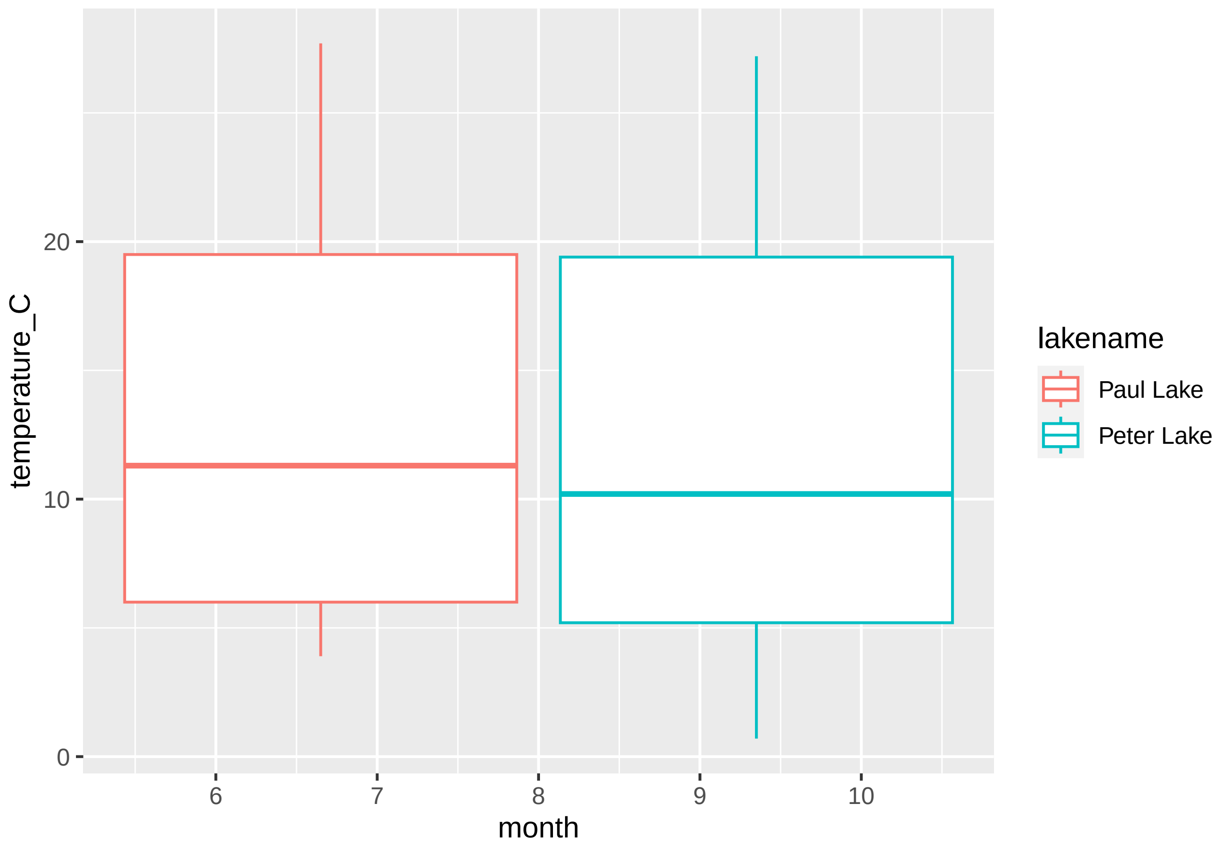
```
#5

PeterPaul_temp <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename))
print(PeterPaul_temp)
```

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



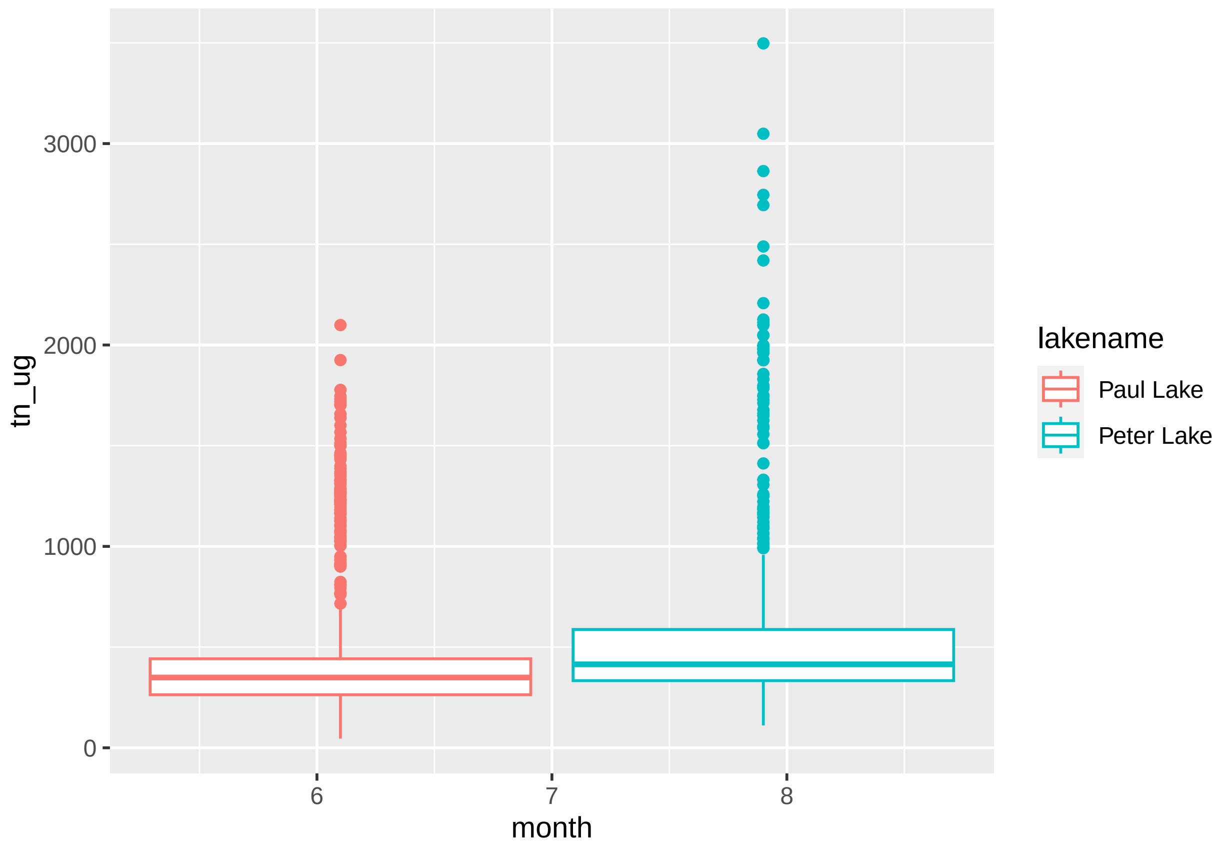
```
###
PeterPaul_TP <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename))
print(PeterPaul_temp)
```

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



```
###
PeterPaul_TN <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename))
print(PeterPaul_TN)
```

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

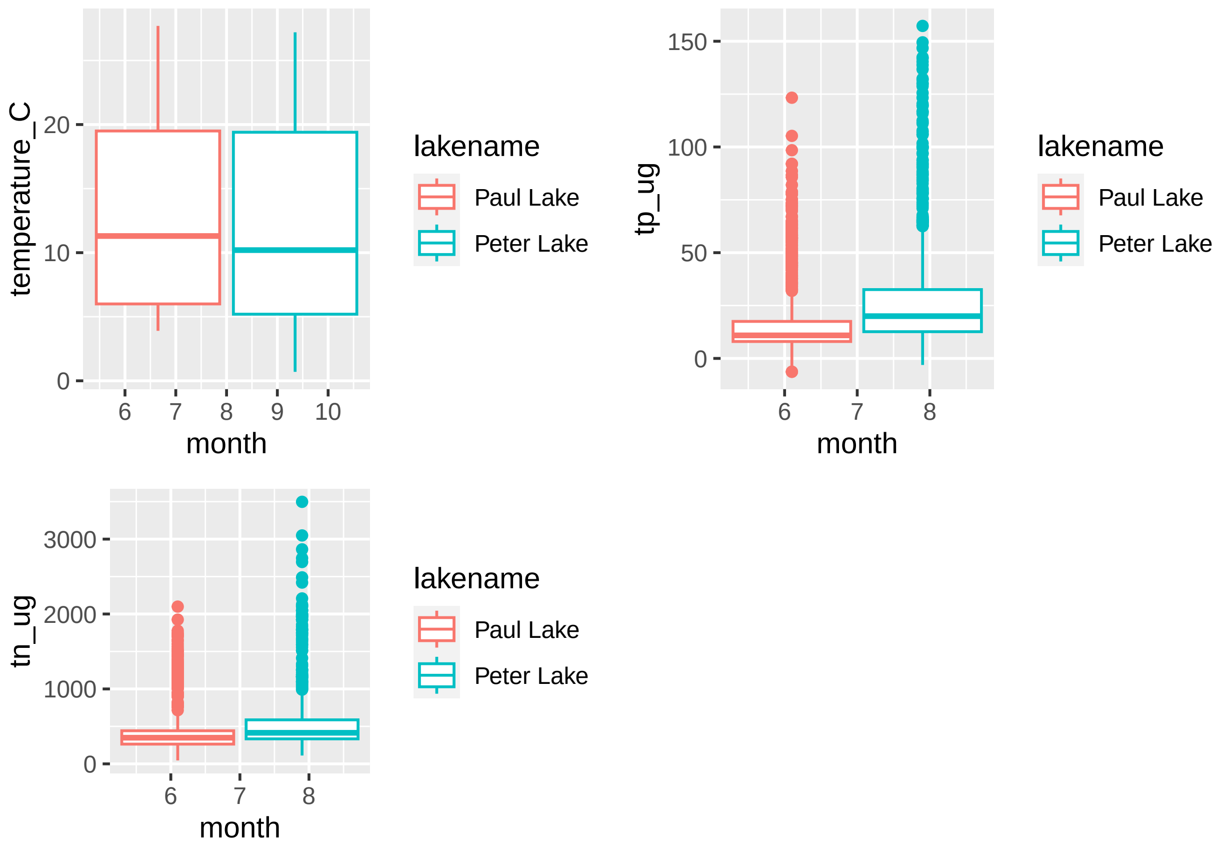


```r
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

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

```r
plot_grid(PeterPaul_temp, PeterPaul_TP, PeterPaul_TN, nrow = 2, align = 'h', rel_heights = c(1.25, 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()`).
```

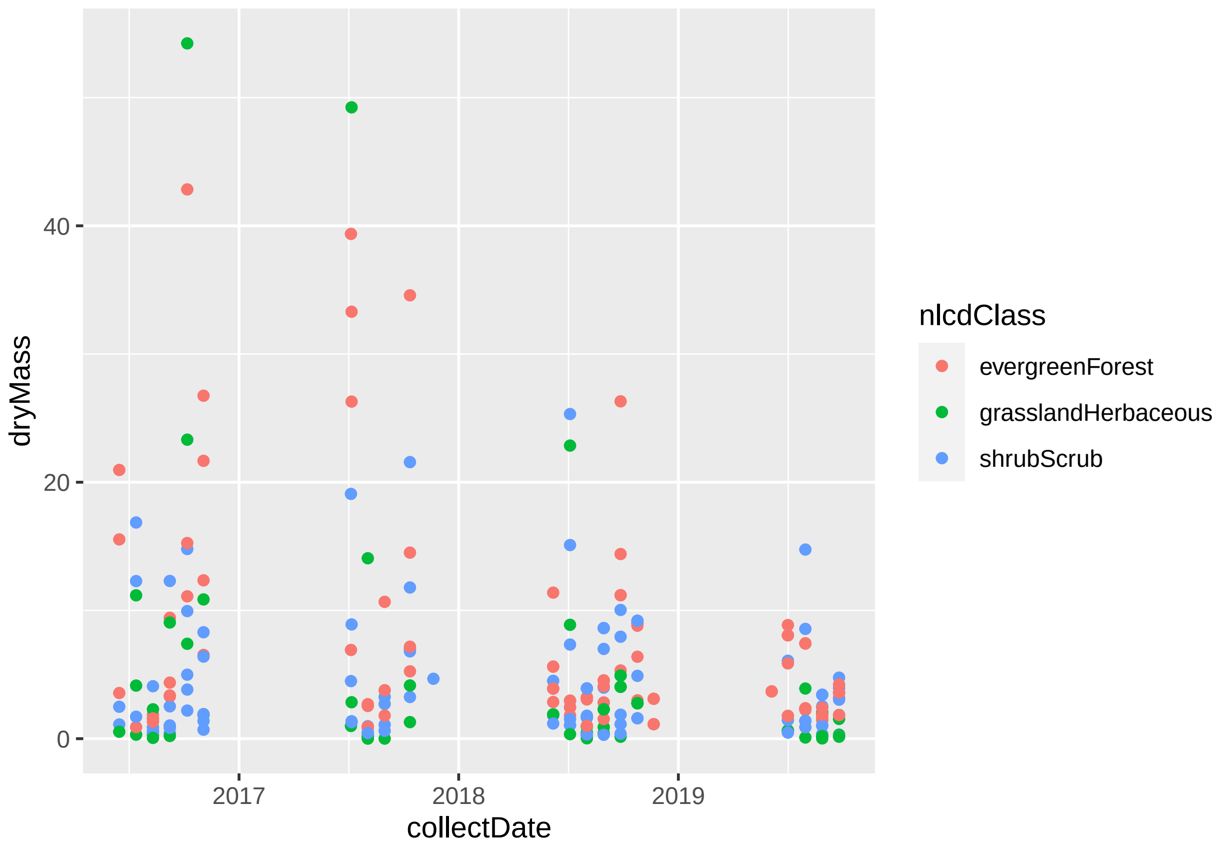


Question: What do you observe about the variables of interest over seasons and between lakes?

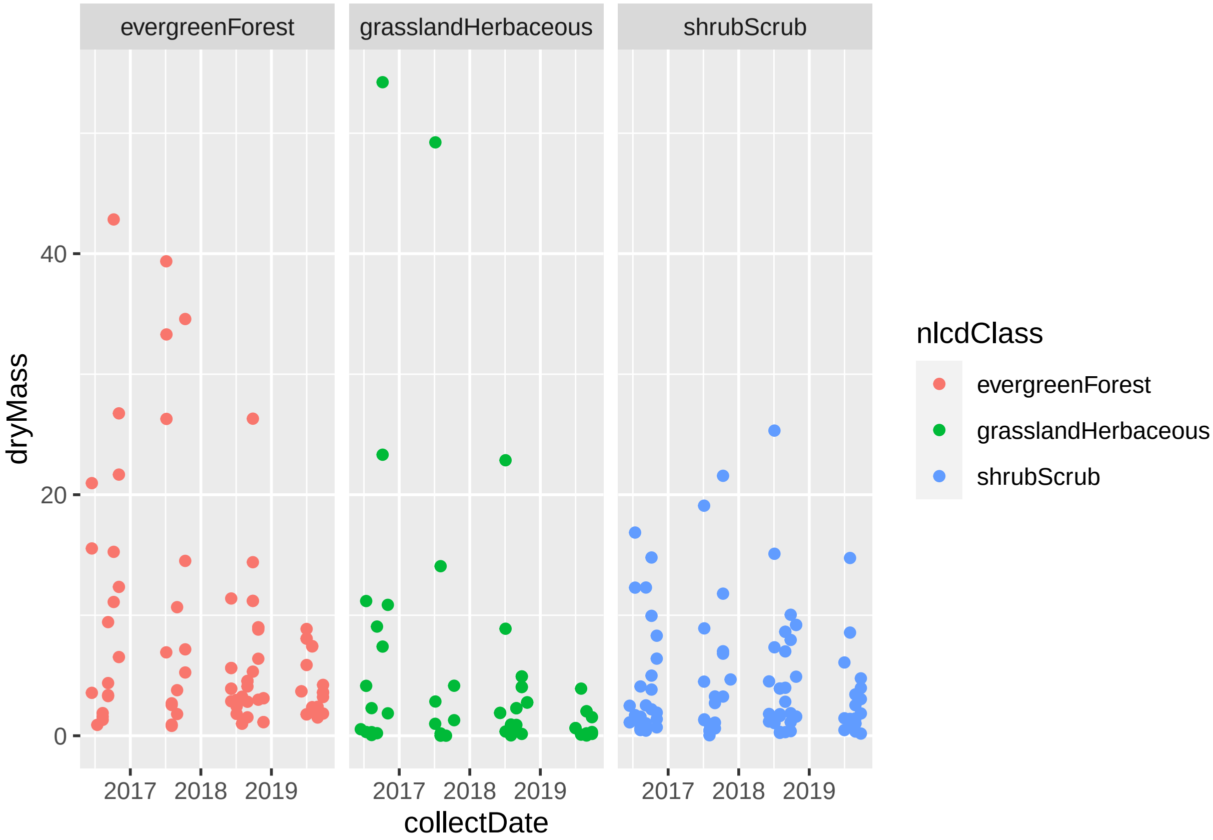
> Answer: That for the summer months lake Paul

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.

```r
#6
Filtered_neon <- filter(NeonNiwo.LiterMass, functionalGroup=='Needles')

ggplot(Filtered_neon)+
  aes(x=collectDate, y=dryMass, color= nlcdClass)+
    geom_point()
```



```
#7
ggplot(Filtered_neon)+
  aes(x=collectDate, y=dryMass, color= nlcdClass)+
    geom_point() +
 facet_wrap(vars(nlcdClass))
```



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

Answer:It is the plot we graph in number is 7,with the facet_wrap function we can now evaluate view the data in a more visually appealing form. One that has the has seperated nlcd class by the schrubScrup, evergreenforest and grasslandherbaceous.