Assignment 5: Data Visualization

Jiawei Liang

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct 14th @ 5:00pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse, lubridate, & cowplot packages. 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_PeterP`
version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processe`
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 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- 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(dplyr)
library(cowplot)
```

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

```
getwd()
```

```
## [1] "C:/Users/Jiawei Liang/Documents/EDA-Fall2022/Assignments"
```

```
setwd('c:/Users/Jiawei Liang/Documents/EDA-Fall2022/Data/Processed/')
NTL_LTER_LAKE_PeterPaul <-read.csv('NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv',stringsAs
NEON_NIWO <- read.csv('NEON_NIWO_Litter_mass_trap_Processed.csv', stringsAsFactors = TRUE)
#2
NTL_LTER_LAKE_PeterPaul$sampledate <- as.Date(NTL_LTER_LAKE_PeterPaul$sampledate, format = "%m/%d/%y")
NEON_NIWO$collectDate <- as.Date(NEON_NIWO$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme.

```
#3
theme_default <- theme_set(theme_bw())
theme_set(theme_default)
#theme_update(panel.grid.minor = element_line(colour = "red"))
```

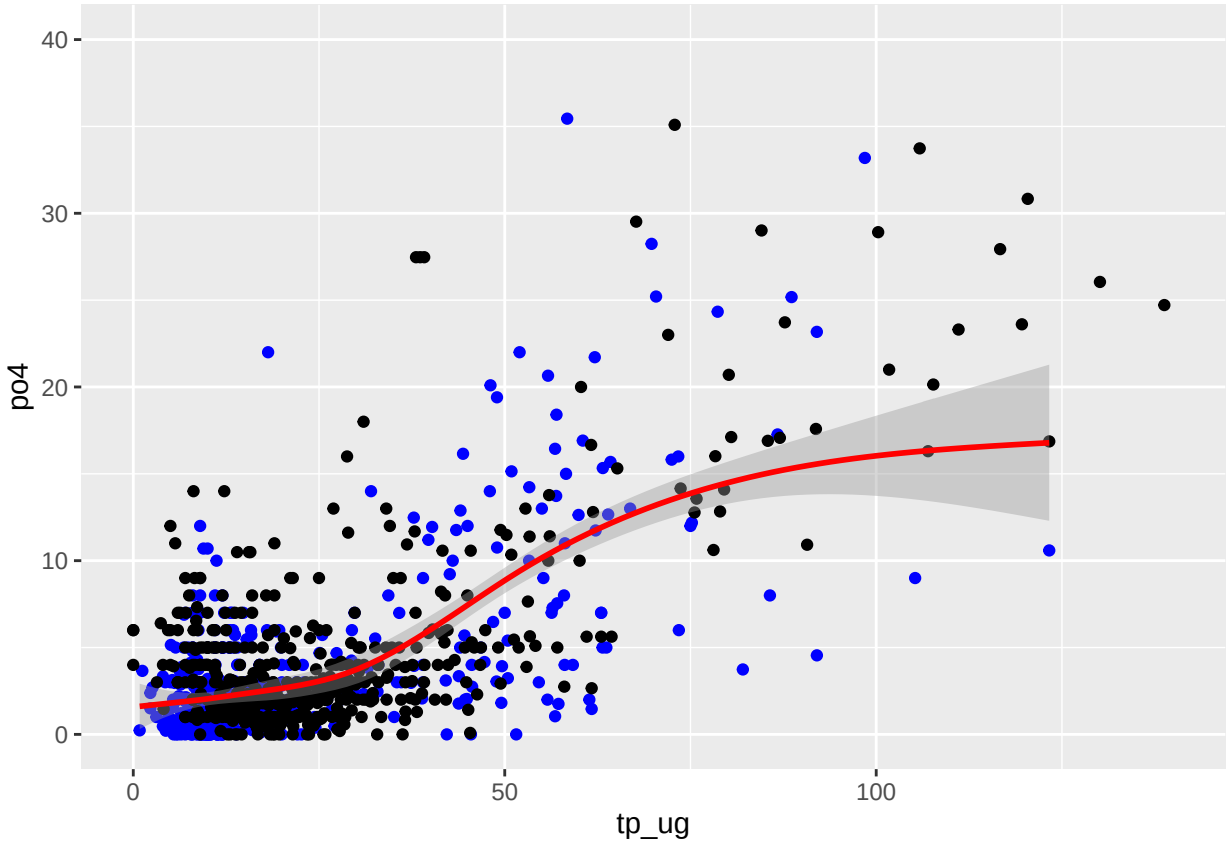
Create graphs

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()`).

```r
#4summary(NTL_LTER_LAKE_PeterPaul$lakename)
NTL_LTER_LAKE_PeterPaul1 <- filter(NTL_LTER_LAKE_PeterPaul, lakename == "Paul Lake")

NTL_LTER_LAKE_PeterPaul2 <- filter(NTL_LTER_LAKE_PeterPaul, lakename == "Peter Lake")

NTL_LTER_LAKE_PeterPaul3 <- ggplot(NULL, aes(x = tp_ug, y = po4)) +
  geom_point(data=NTL_LTER_LAKE_PeterPaul1, color = "blue") +
  geom_point(data=NTL_LTER_LAKE_PeterPaul2) +
  geom_smooth(data=NTL_LTER_LAKE_PeterPaul1, color = "red") +
  xlim( 0 , 140 ) +
  ylim( 0 , 40 )
print(NTL_LTER_LAKE_PeterPaul3)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 10525 rows containing non-finite values (stat_smooth).

## Warning: Removed 10525 rows containing missing values (geom_point).

## Warning: Removed 11426 rows containing missing values (geom_point).
```



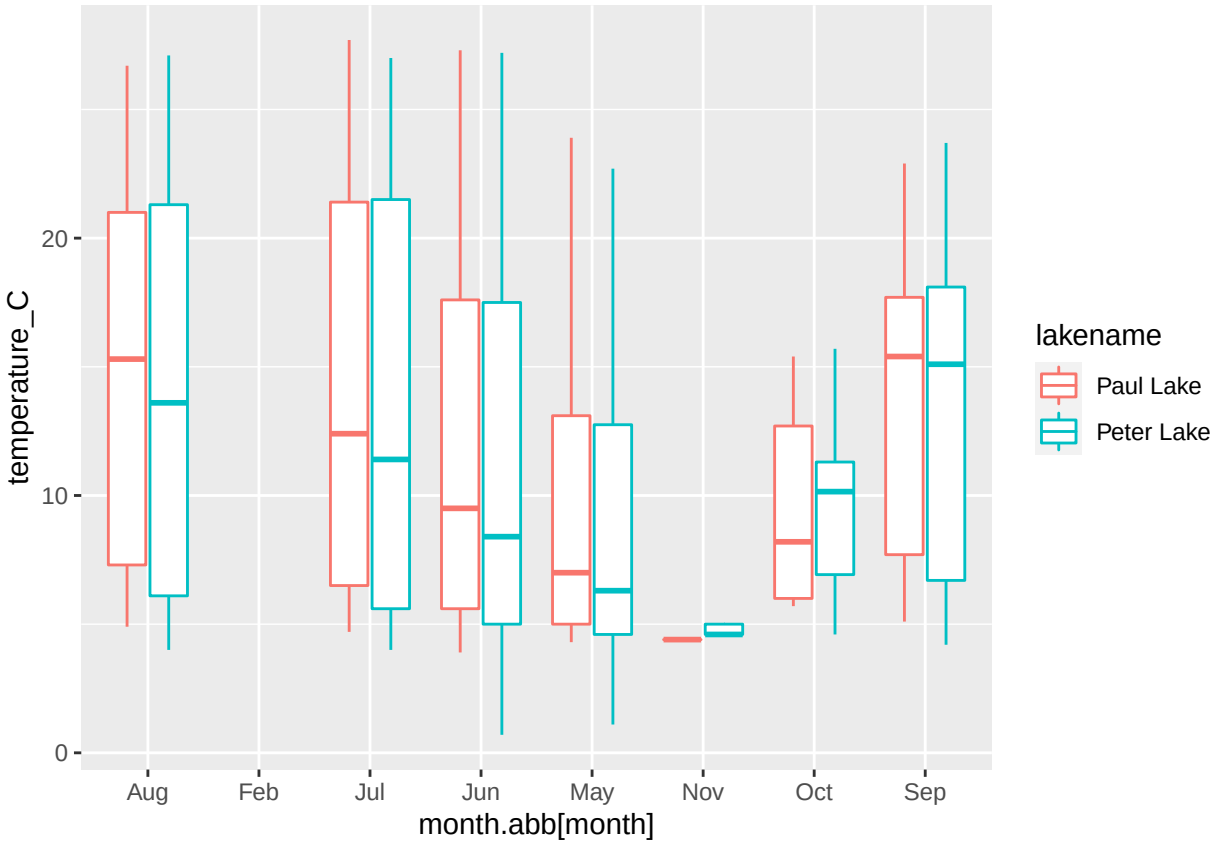
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

(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

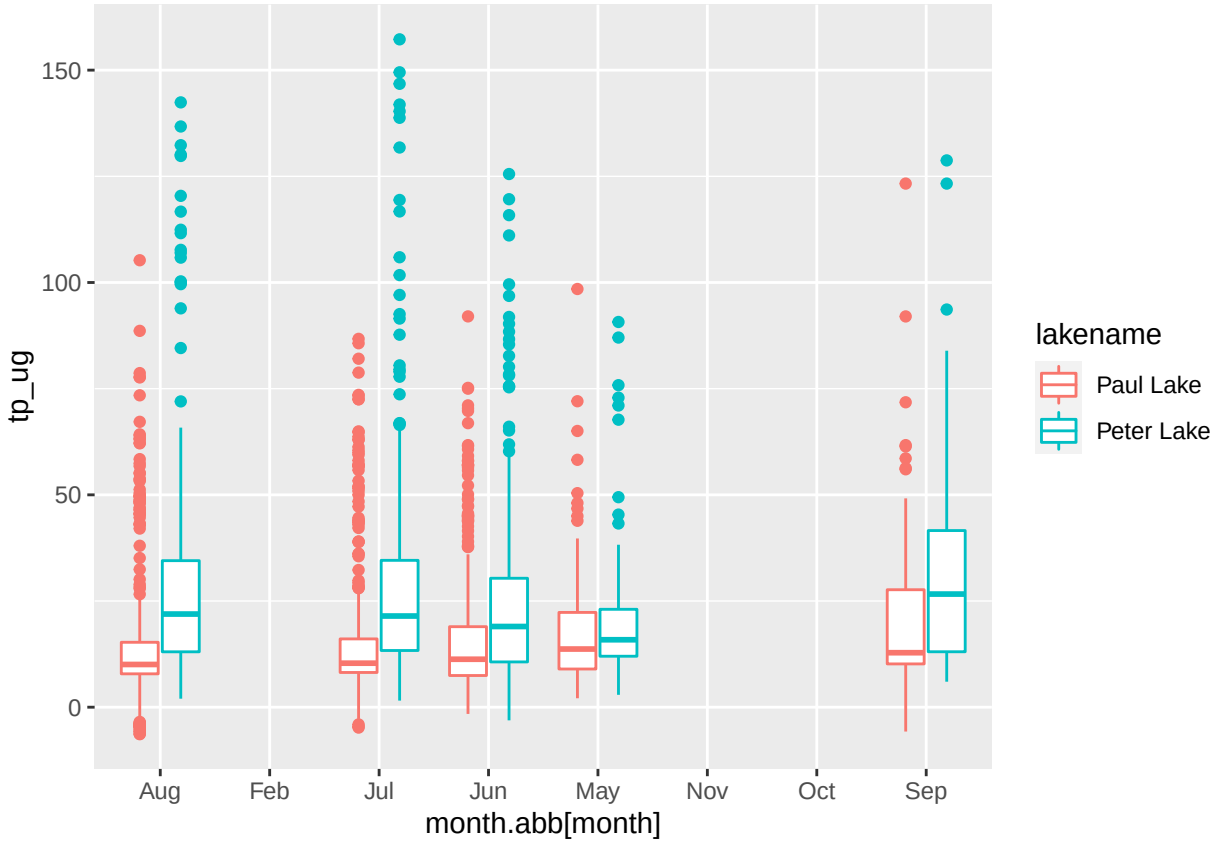
```
#5
Tem <-
  ggplot(NTL_LTER_LAKE_PeterPaul, aes(x = month.abb[month], y = temperature_C)) +
  geom_boxplot(aes(color = lakename))
print(Tem)
```

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



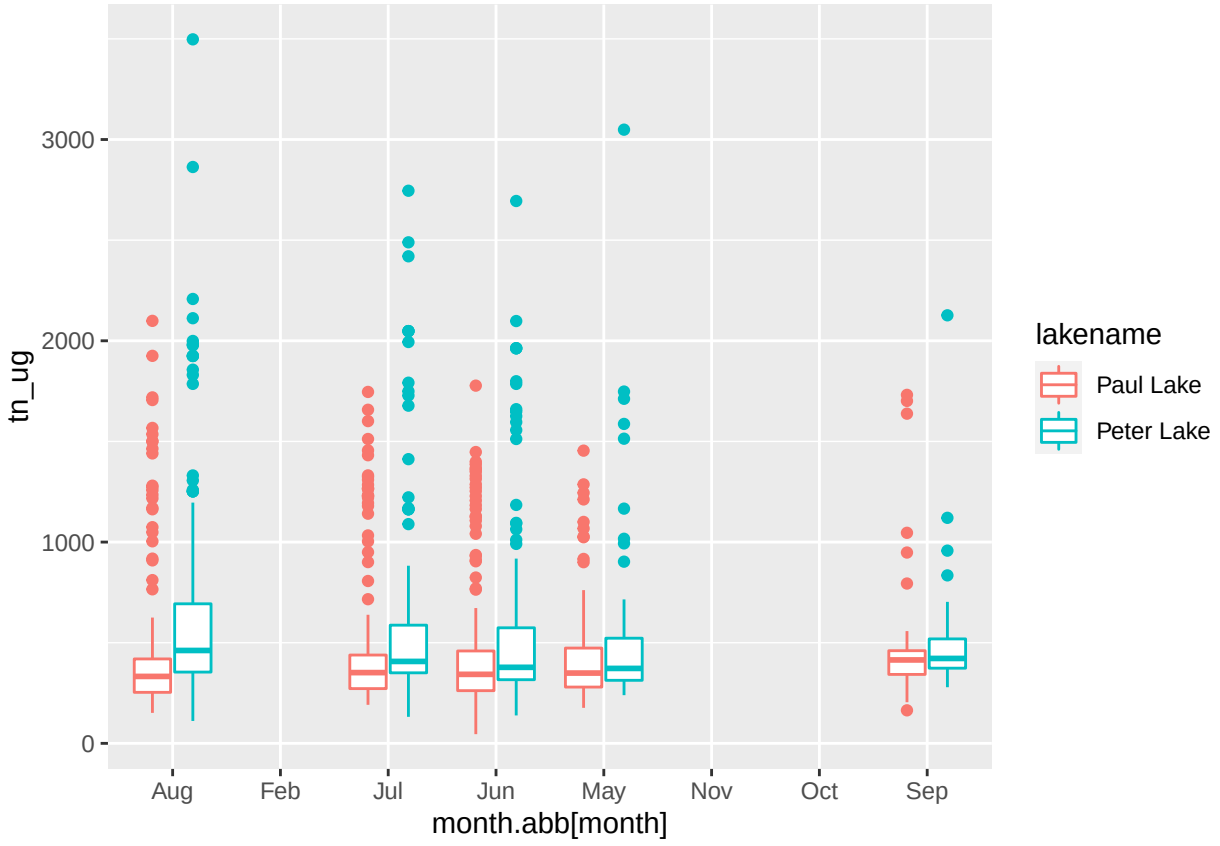

```
TP_ug <-
  ggplot(NTL_LTER_LAKE_PeterPaul, aes(x = month.abb[month], y = tp_ug)) +
  geom_boxplot(aes(color = lakename))
print(TP_ug)
```

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




```
TN_ug  <-
  ggplot(NTL_LTER_LAKE_PeterPaul, aes(x = month.abb[month], y = tn_ug)) +
  geom_boxplot(aes(color = lakename))
print(TN_ug)
```

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

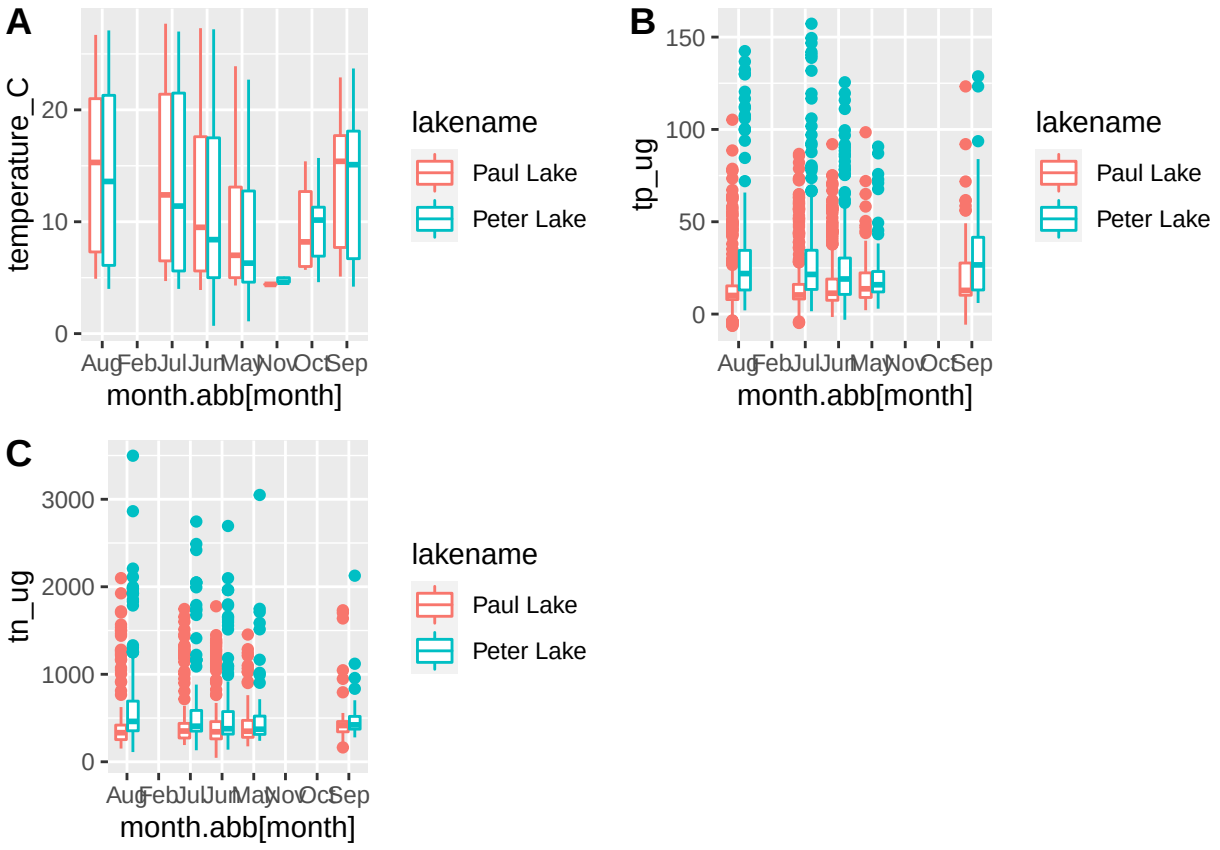



```
plot_grid(Tem, TP_ug, TN_ug, labels=c("A", "B", "C"), ncol = 2, nrow = 2)
```

```
## 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).
```



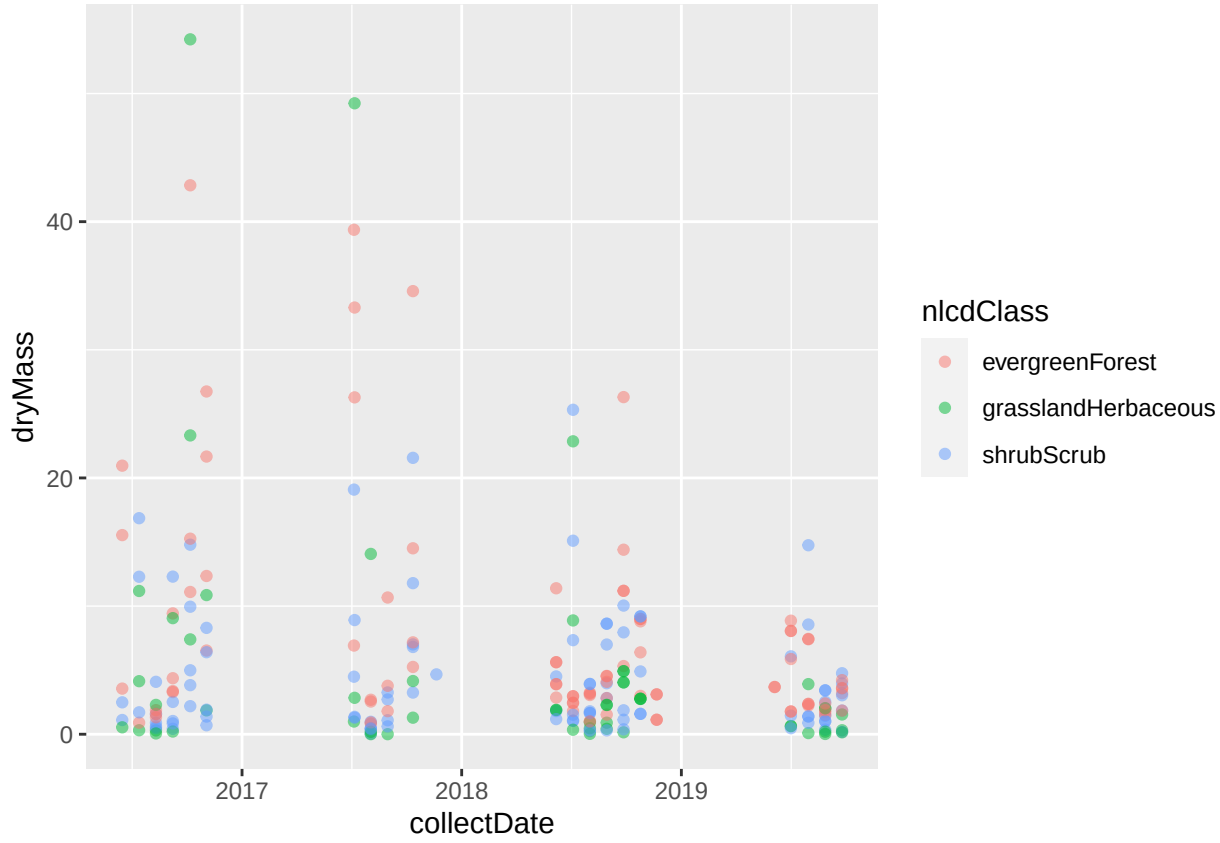
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: For temperature, Paul Lake and Peter Lake do not have big different. But for TP_ug and TN_ug, Peter Lake are bigger than Paul Lake.

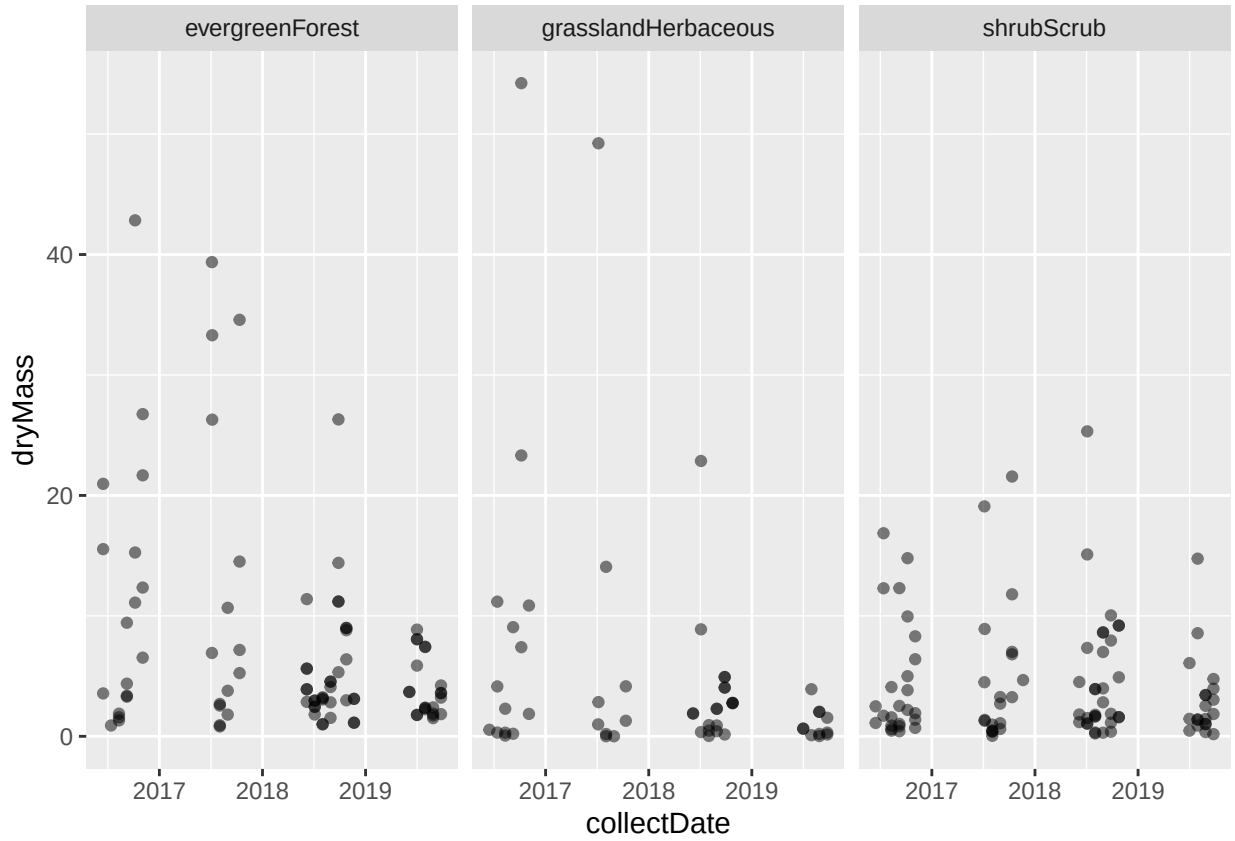
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
library(ggplot2)
needles = NEON_NIWO[NEON_NIWO$functionalGroup=="Needles",]
image1 <- ggplot(needles, aes(x = collectDate, y = dryMass))+
  geom_point(aes(color=nlcdClass),alpha=0.5)
print(image1)
```




```
#7
image2 <- ggplot(needles, aes(x = collectDate, y = dryMass))+
  geom_point(alpha=0.5)+
  facet_wrap(~nlcdClass)
print(image2)
```



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

Answer: I think picture 7 is more effective. Because in picture 6, lots of points are overlapping each other. So they could not recognize.