Assignment 5: Data Visualization

Zhiteng Ma

{r setup, include=FALSE, echo=FALSE} require("knitr") opts_knit$set(root.dir = "~/path/to/folder/")

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_PeterPa`
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   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## 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)
```

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

```
library(dplyr)
library(cowplot)
```

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

```
library(ggplot2)
getwd()
```

```
## [1] "C:/Users/Zhiteng Ma/Desktop"
```

```
setwd("c:/Users/Zhiteng Ma/Desktop/EDA-Fall2022-main/Data/Processed/")
PeterPaul <- read.csv("NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
    stringsAsFactors = TRUE)
NEON <- read.csv("NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = TRUE)
# 2
PeterPaul$sampledate <- as.Date(PeterPaul$sampledate, format = "%Y-%m-%d")
NEON$collectDate <- as.Date(NEON$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.

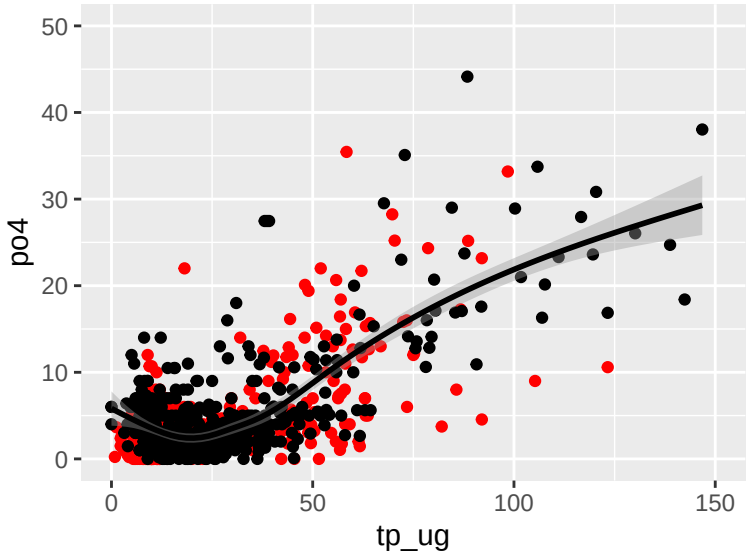
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
# PeterPaul <-

summary(PeterPaul$lakename)
```

```
##  Paul Lake Peter Lake
##      11060      11948
```

```
PeterPaul1 <- filter(PeterPaul, lakename == "Paul Lake")

PeterPaul2 <- filter(PeterPaul, lakename == "Peter Lake")

PeterPaul3 <- ggplot(NULL, aes(x = tp_ug, y = po4)) + geom_point(data = PeterPaul1,
    color = "red") + geom_point(data = PeterPaul2) + geom_smooth(data = PeterPaul2,
    color = "black") + xlim(0, 150) + ylim(0, 50)
print(PeterPaul3)
```

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




```
# 4
```

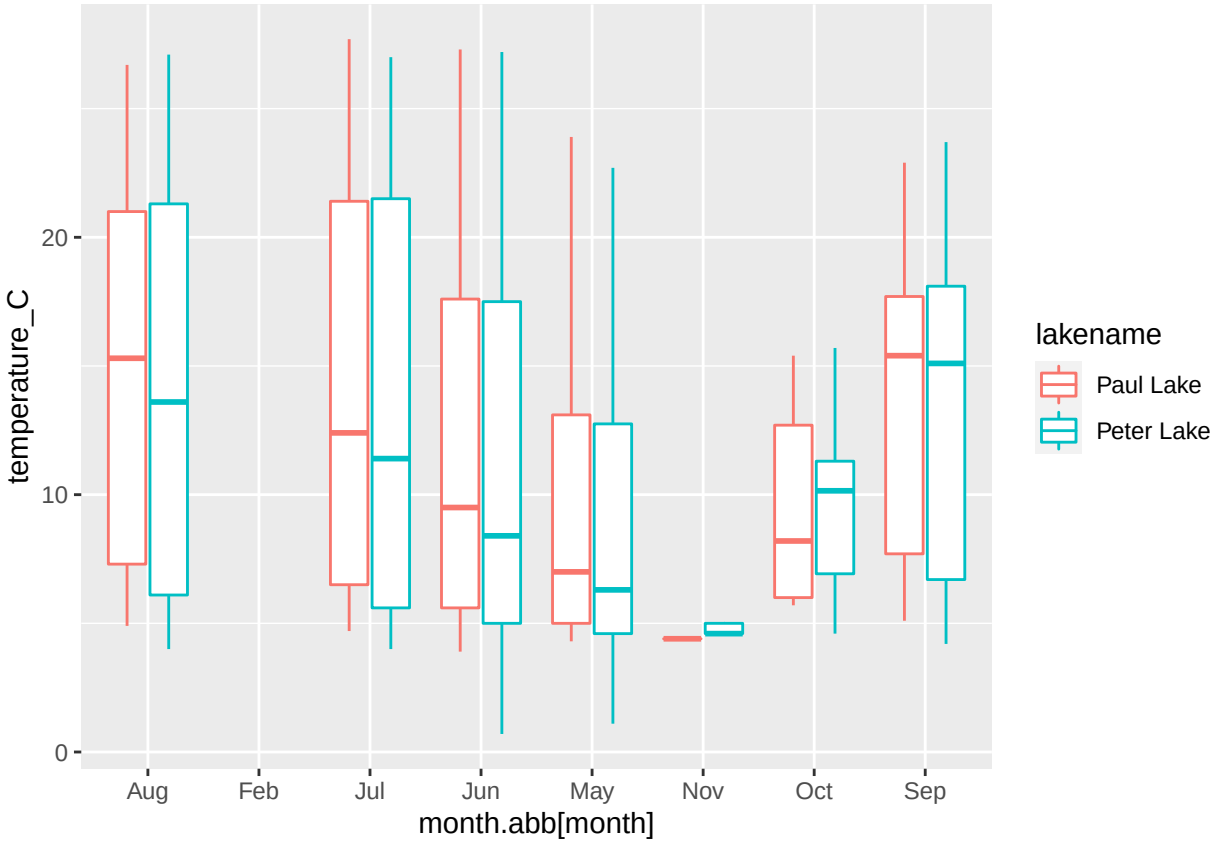
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

(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

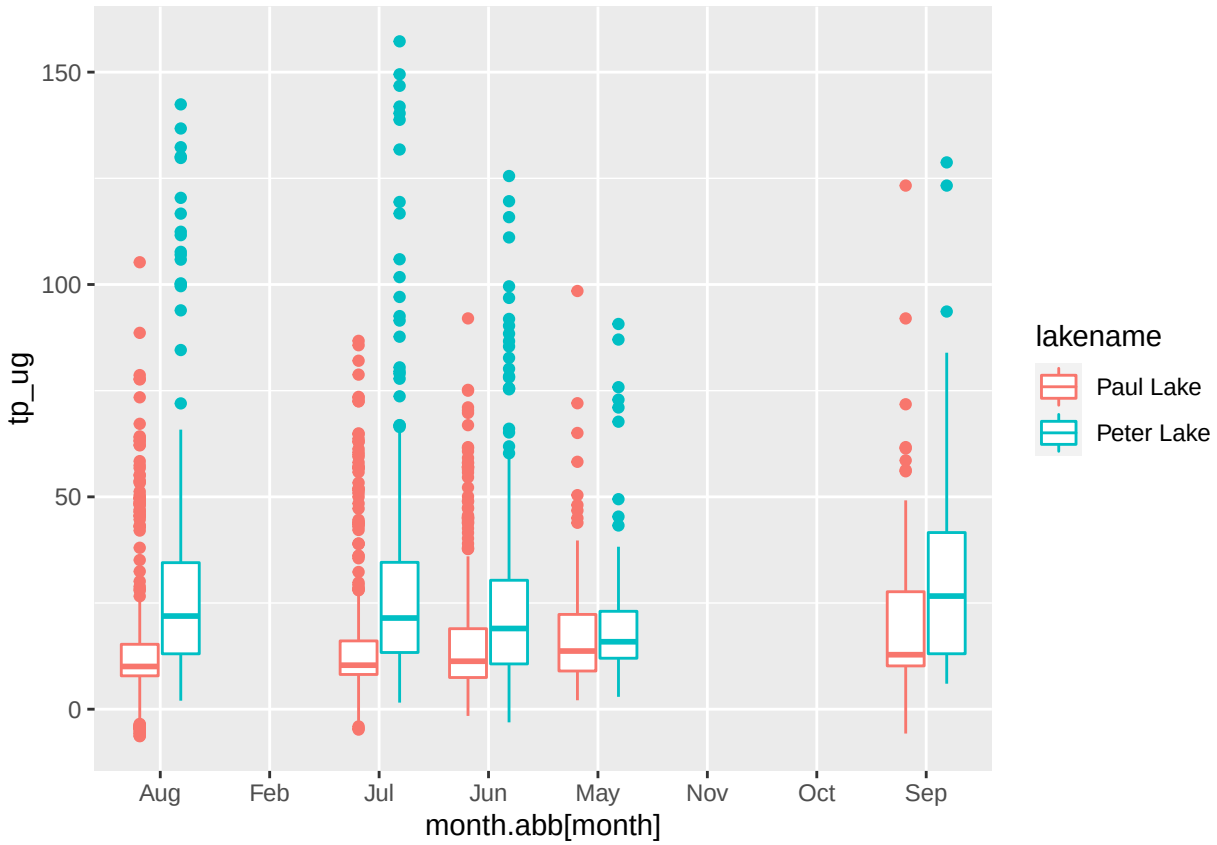
```
# 5

temperature <- ggplot(PeterPaul, aes(x = month.abb[month], y = temperature_C)) +
    geom_boxplot(aes(color = lakename))  # Why didn't we use 'fill'?
print(temperature)
```

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



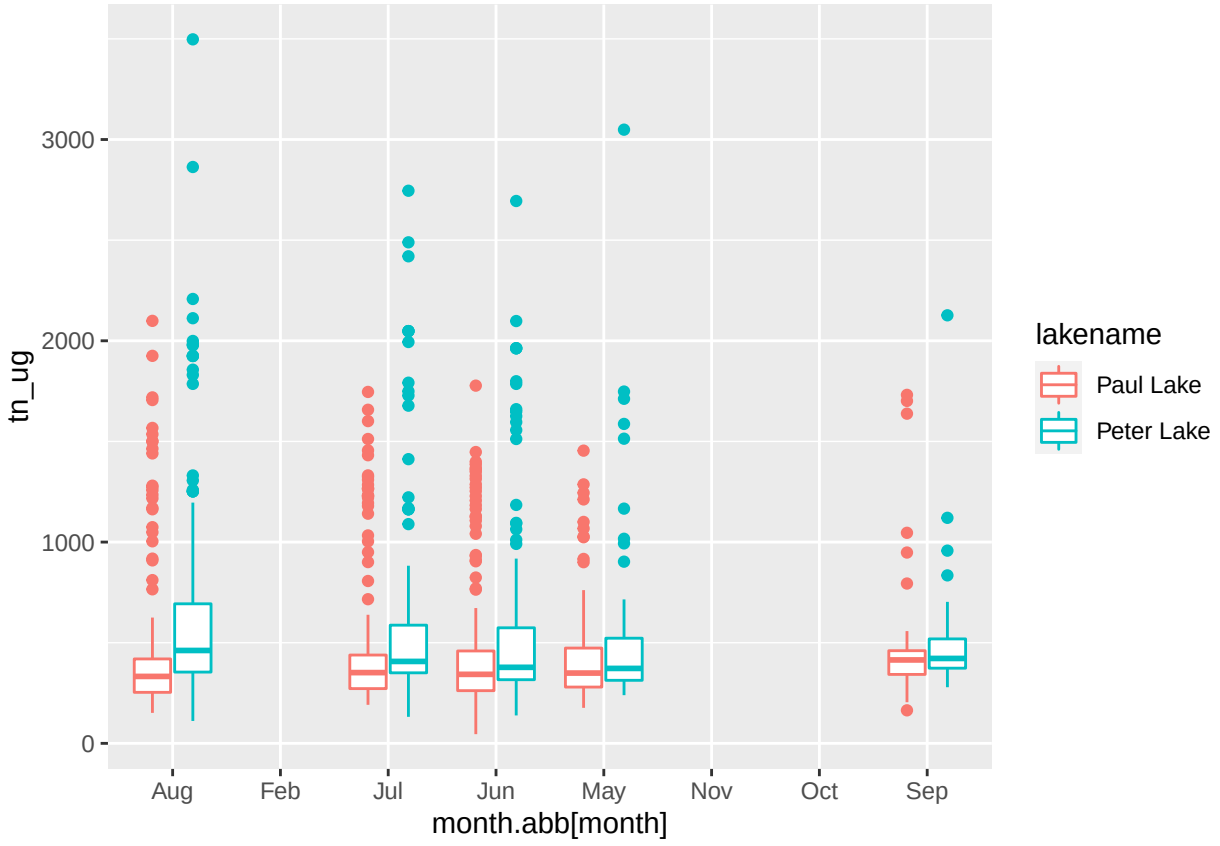

```
TP <- ggplot(PeterPaul, aes(x = month.abb[month], y = tp_ug)) + geom_boxplot(aes(color = lakename))  #
print(TP)
```

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




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

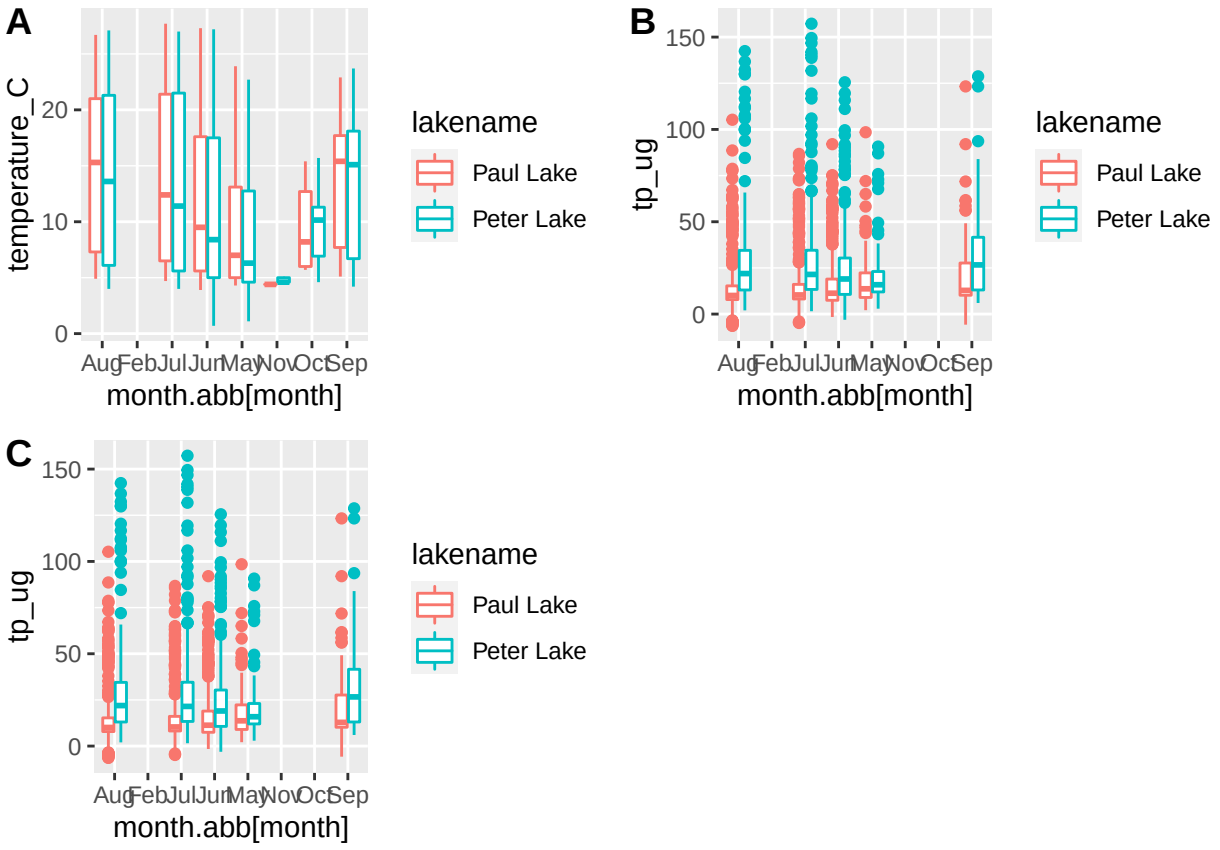
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```




```
plot_grid(temperature, TP, TP, 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).
## Removed 20729 rows containing non-finite values (stat_boxplot).
```



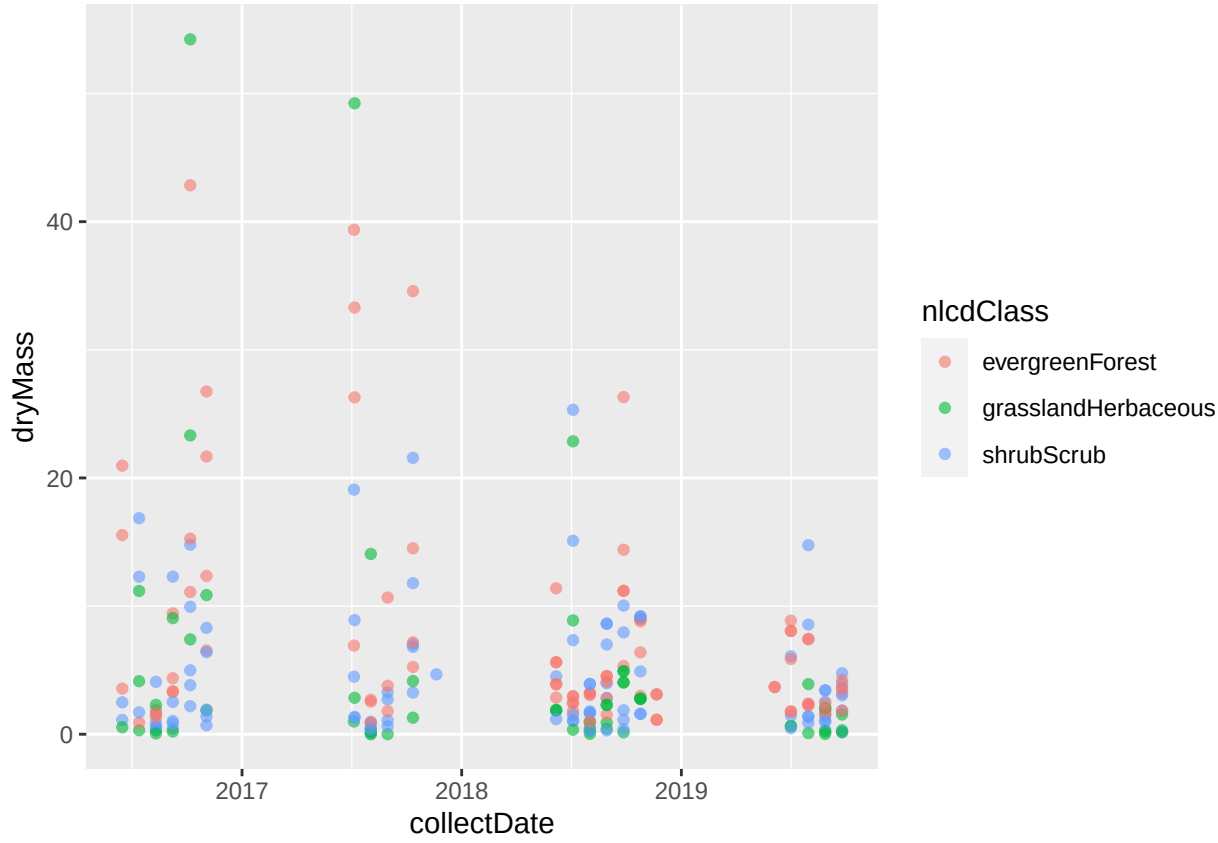
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: In my observation, under the temperature figure, the temperature in November is not low, and in most months, the temperature span of Peter Lake is larger than that of Paul Lake.

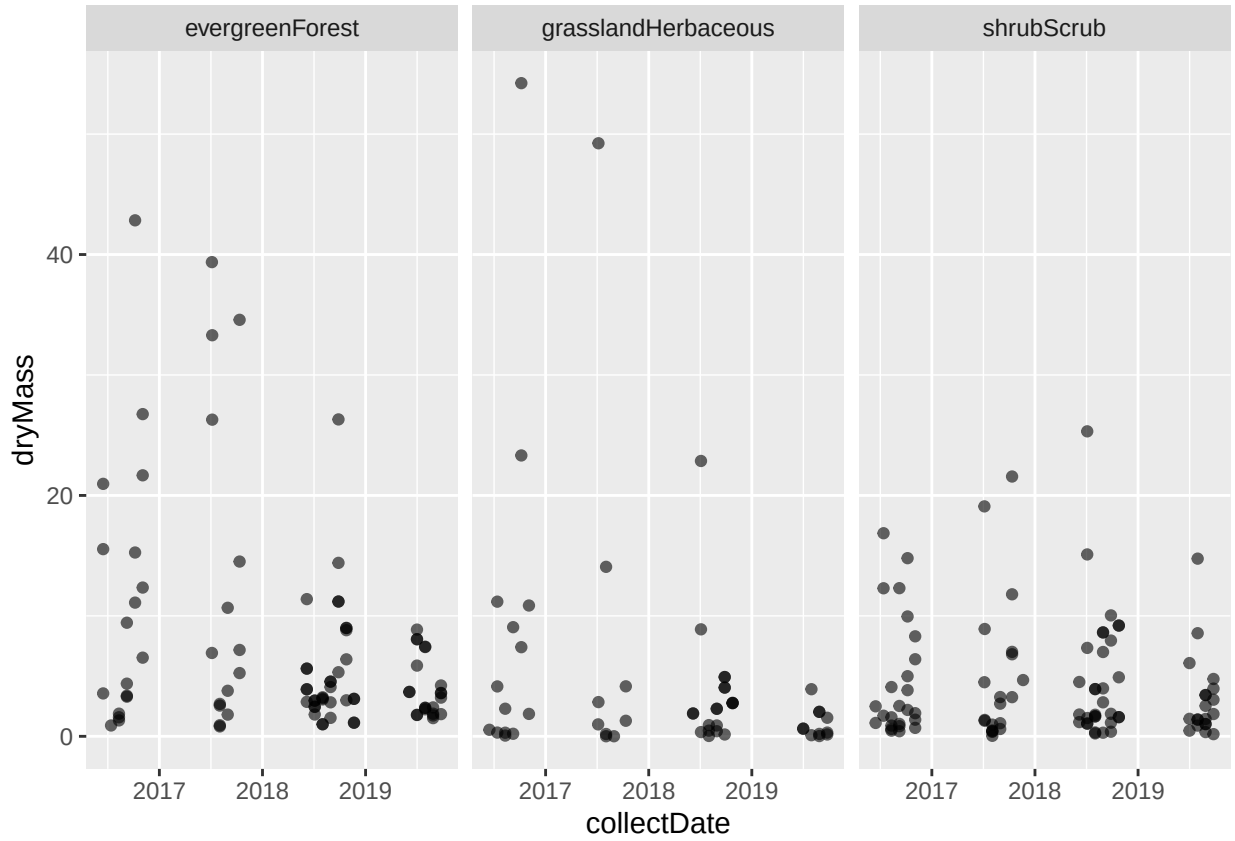
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
# 6

needles = NEON[NEON$functionalGroup == "Needles", ]
picture <- ggplot(needles, aes(x = collectDate, y = dryMass)) + geom_point(aes(color = nlcdClass),
    alpha = 0.6)
print(picture)
```




```
# 7

p2 <- ggplot(needles, aes(x = collectDate, y = dryMass)) + geom_point(alpha = 0.6) +
    facet_wrap(~nlcdClass)
print(p2)
```



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

Answer: I think Figure 7 is more efficient than Figure 6. In Figure 6, many points overlap each other, while in Figure 7, many points do not overlap. So in this case, Figure 7 is more clear.