Assignment 5: Data Visualization

Xiaoge Zhang

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.

```r
#1
#load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## 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)
```

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

```
library(here)
```

```
## here() starts at /home/guest/R/EDA-Spring2023
```

```
library(cowplot)
```

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

```
# check working directory
getwd()
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

```
#upload datasets
lakes <- read.csv('./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv', strings
litter <- read.csv('./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv', stringsAsFactors = TRUE

#2
# change the date columns to Date format
lakes$sampledate <- as.Date(lakes$sampledate, format = '%Y-%m-%d')
litter$collectDate <- as.Date(litter$collectDate, format = '%Y-%m-%d')
```

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

```r
#3
hw_theme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "grey"),
        axis.title = element_text(color='grey'),
        legend.position = "top",
        plot.background = element_rect(fill = 'lightyellow'),
        plot.title = element_text(size = rel(2)))
theme_set(hw_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.

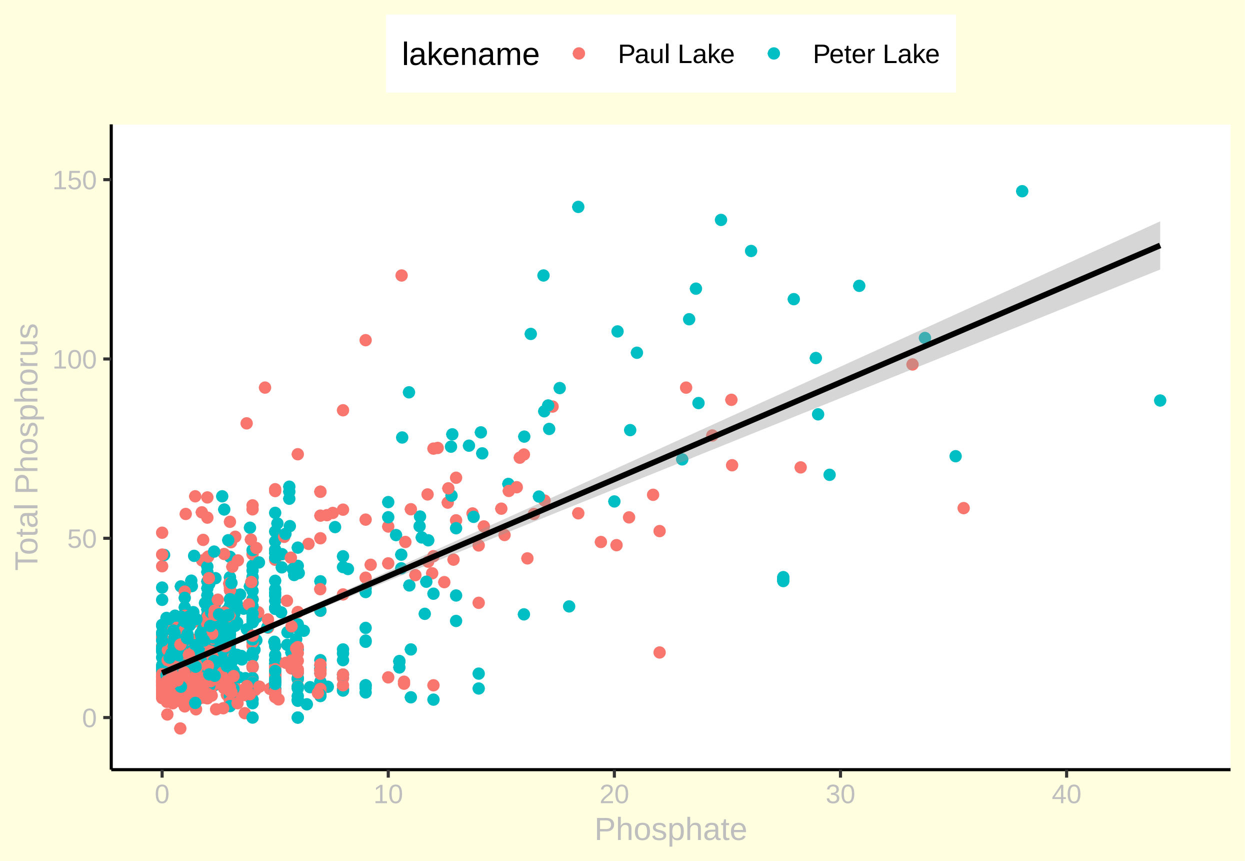
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4
plot4 <-
  ggplot(lakes, aes(y = tp_ug, x = po4)) +
  geom_point(aes(color=lakename)) +
  geom_smooth(method = lm, color='black') +
  xlim(0,45) +
  ylab('Total Phosphorus') +
  xlab('Phosphate')
print(plot4)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21947 rows containing missing values (`geom_point()`).
```

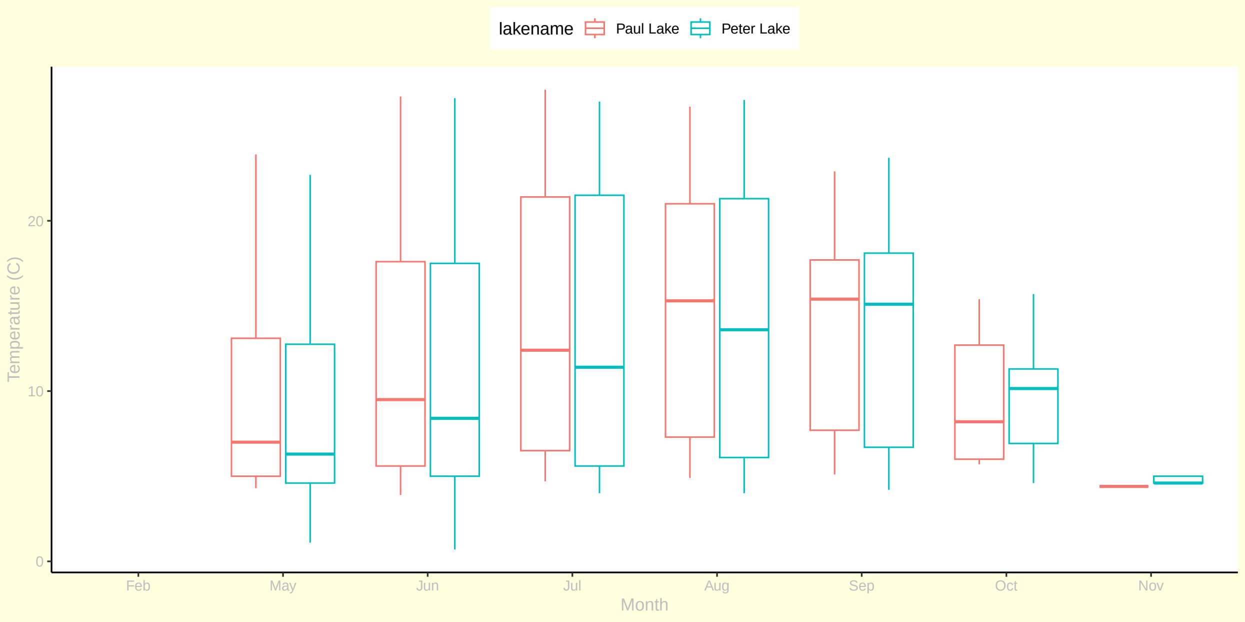


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

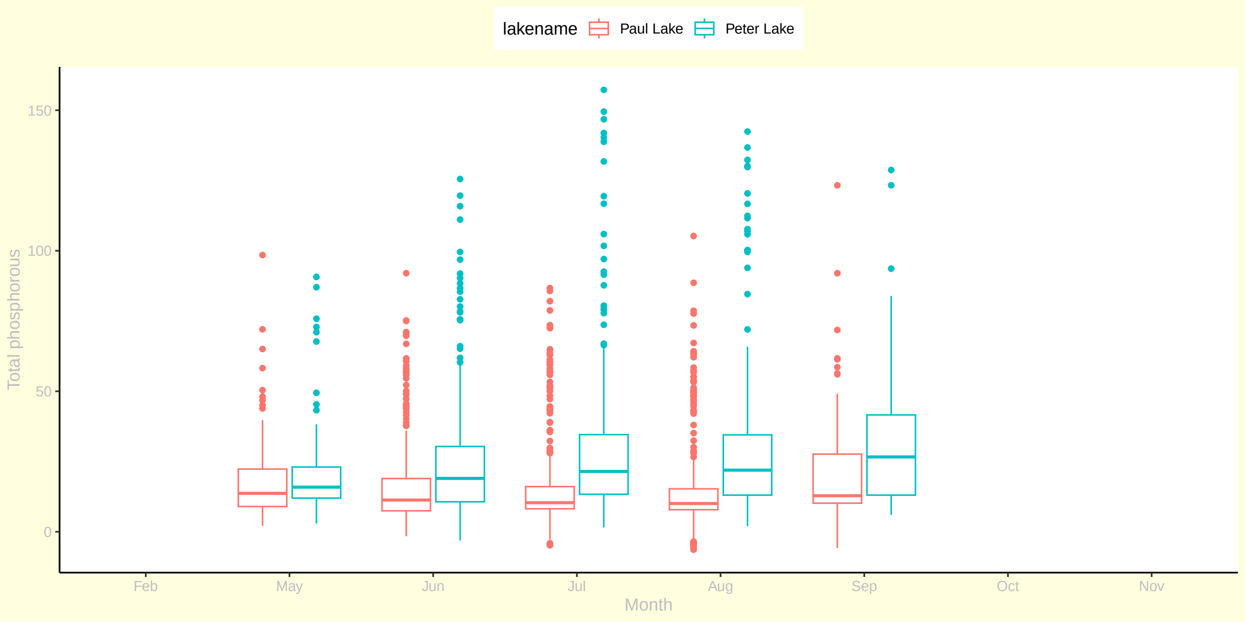
```
#5
# temperature
plot5a <-
  ggplot(lakes, aes(x = factor(month, levels=1:12, labels=month.abb), y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  xlab('Month') +
  ylab('Temperature (C)')
print(plot5a)
```

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



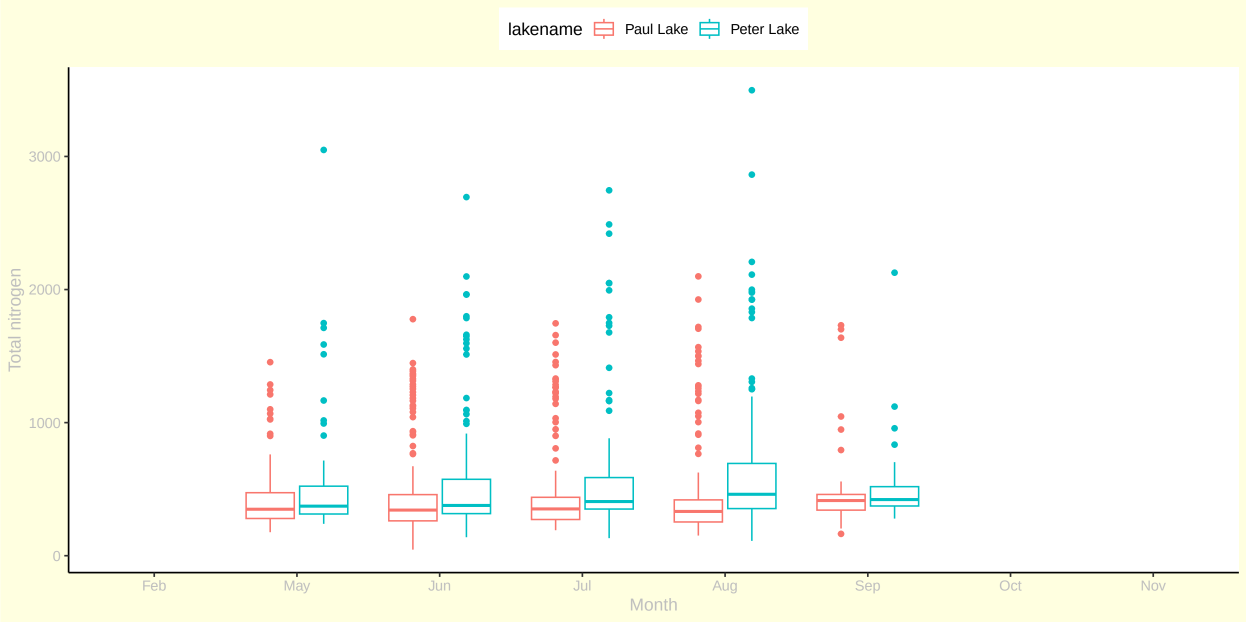
```
# TP
plot5b <-
  ggplot(lakes, aes(x = factor(month, levels=1:12, labels=month.abb), y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  xlab('Month') +
  ylab('Total phosphorous')
print(plot5b)
```

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



```
# TN
plot5c <-
  ggplot(lakes, aes(x = factor(month, levels=1:12, labels=month.abb), y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  xlab('Month') +
  ylab('Total nitrogen')
print(plot5c)
```

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

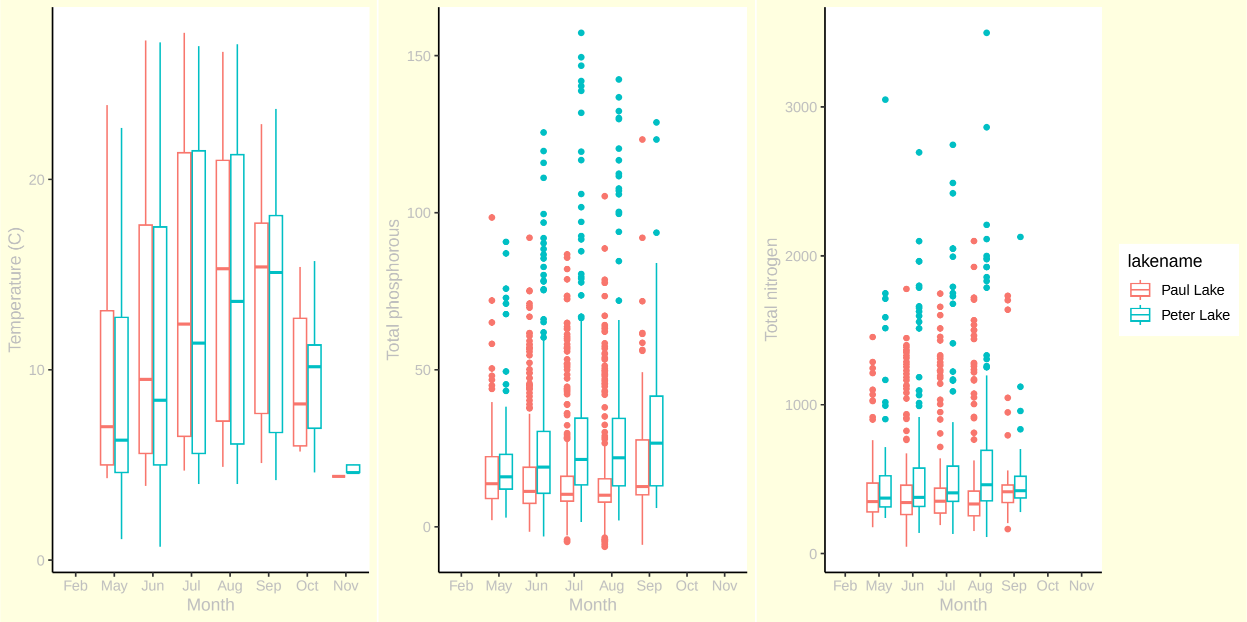


```
# a cowplot that combines the previous three graphs
plot_grid(plot5a + theme(legend.position="none"),
          plot5b + theme(legend.position="none"),
          plot5c + theme(legend.position = 'right'),
          rel_widths = c(1, 1, 1.3), nrow=1, align = 'h')
```

```
## 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()`).
```

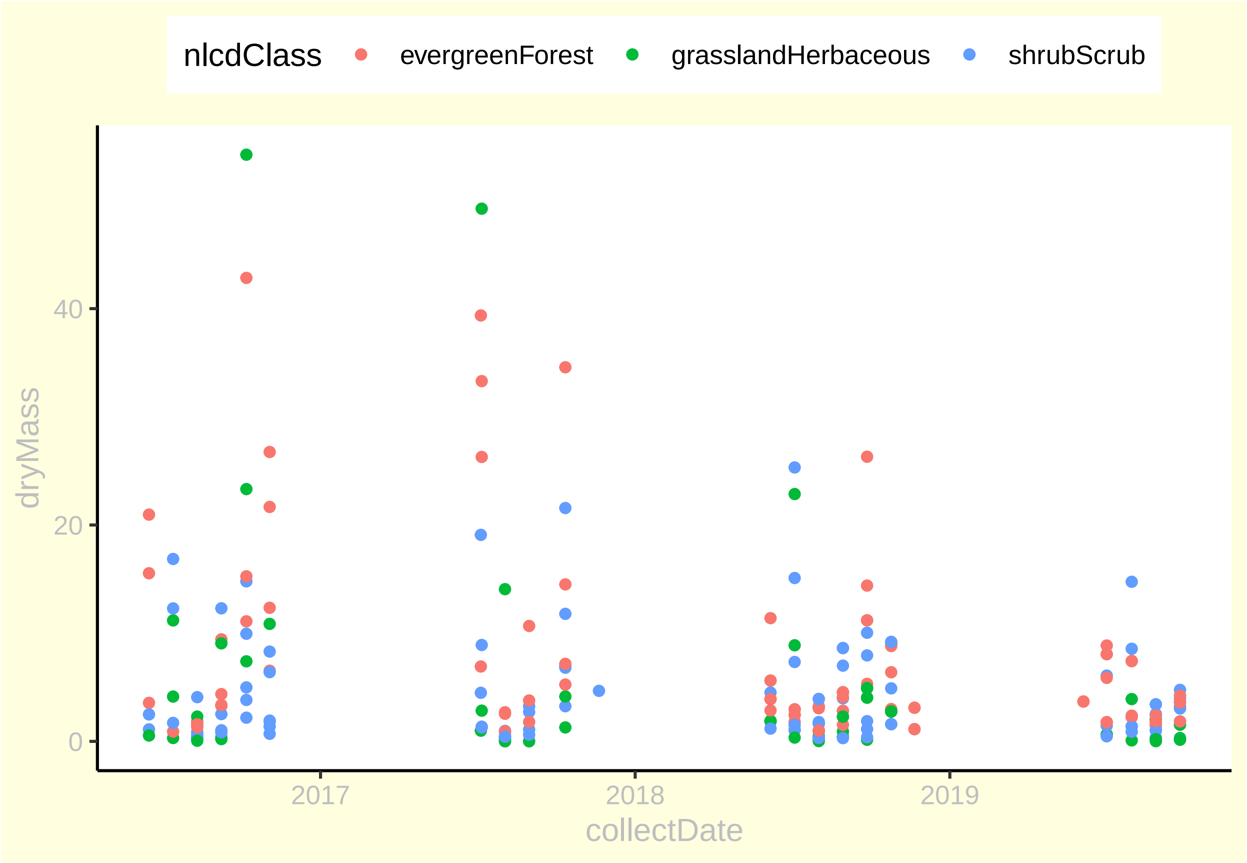


Question: What do you observe about the variables of interest over seasons and between lakes?

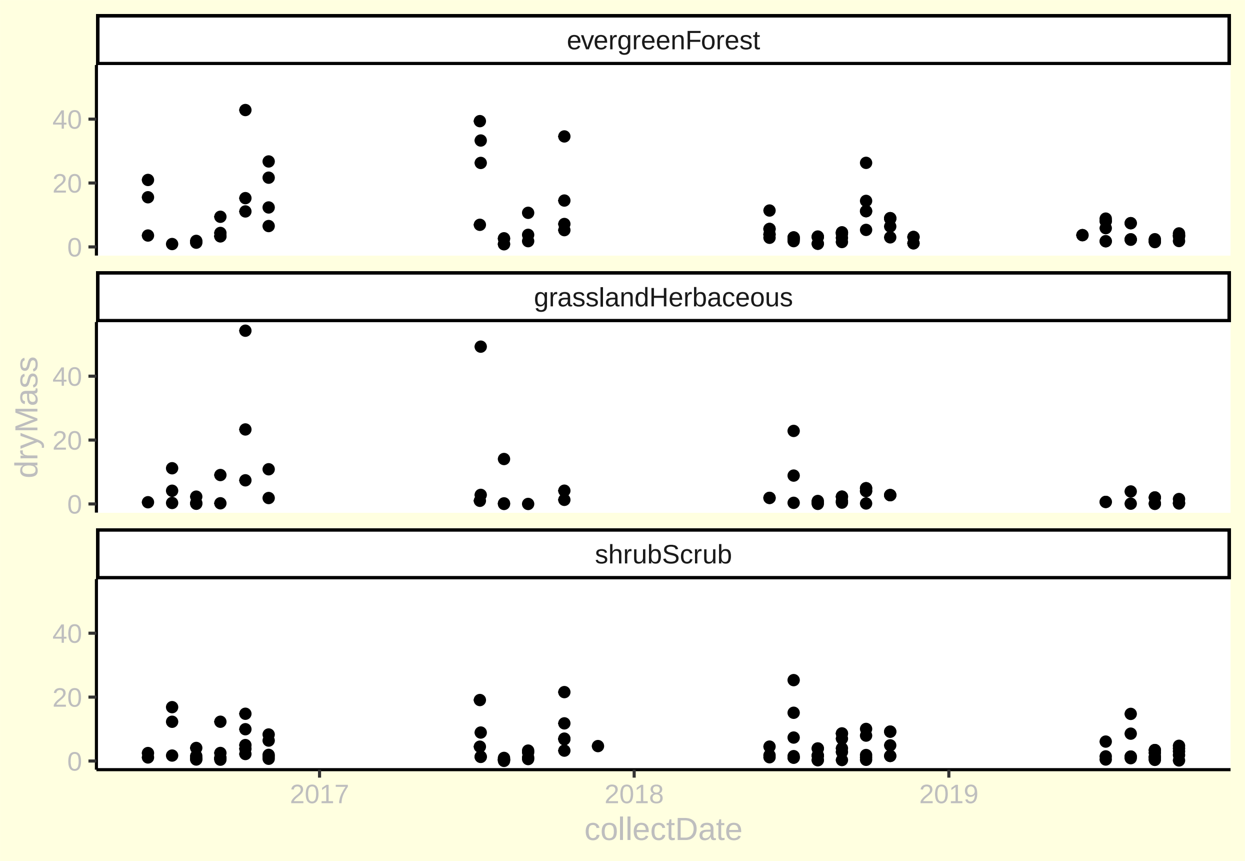
> Answer: For temperature, temperatures for summer and fall are higher than temperatures for winter and spring; Paul Lake usually have a slightly higher median temperature than Peter Lake in most months. For phosphorous, Peter Lake generally has higher amount of total phosphorous than Paul Lake; for Paul Lake, the level did not fluctuate too much over the seasons, while the level slightly increased from May to Sept for Peter Lake. Total nitrogen did not fluctate too much over the seasons for both lakes, while Peter Lake usually has a higher median nitrogen amount 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
plot6 <- ggplot(subset(litter, functionalGroup=='Needles'), aes(x=collectDate, y=dryMass)) +
  geom_point(aes(color=nlcdClass))
print(plot6)
```



```
#7
plot7 <- ggplot(subset(litter, functionalGroup=='Needles'), aes(x=collectDate, y=dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow=3)
print(plot7)
```



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

> Answer: For me, plot 7 is more effective than plot 6 because of it is hard to see trends in plot 6 with all classes mixed together. In plot 7, it is much more clearer to see how dry mass could change for each land cover type in different collection dates.