Assignment 5: Data Visualization

Ying Liu

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
getwd()
```

```
## [1] "C:/Users/Alina/Desktop/DUKE_22FALL/872/EDA-Fall2022/Assignments"
```

```
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   1.2.0      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()
```

```r
library(lubridate)
```

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

```r
library(cowplot)
```

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

```r
PeterPaul.chem.nutrients <- read.csv("C:/Users/Alina/Desktop/DUKE_22FALL/872/EDA-Fall2022/Data/Processed
    stringsAsFactors = T)
NiwotRidgelitter <- read.csv("C:/Users/Alina/Desktop/DUKE_22FALL/872/EDA-Fall2022/Data/Processed/NEON_NI
    stringsAsFactors = T)

# 2
class(PeterPaul.chem.nutrients$sampledate)
```

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

```r
class(NiwotRidgelitter$collectDate)
```

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

```r
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate,
    format = "%Y-%m-%d")
NiwotRidgelitter$collectDate <- as.Date(NiwotRidgelitter$collectDate, format = "%Y-%m-%d")
class(PeterPaul.chem.nutrients$sampledate)
```

```
## [1] "Date"
```

```r
class(NiwotRidgelitter$collectDate)
```

```
## [1] "Date"
```

Define your theme

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

```r
# 3
mytheme <- theme_half_open(font_size = 10) + theme(axis.text = element_text(color = "black"),
    legend.position = "right")
```

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.

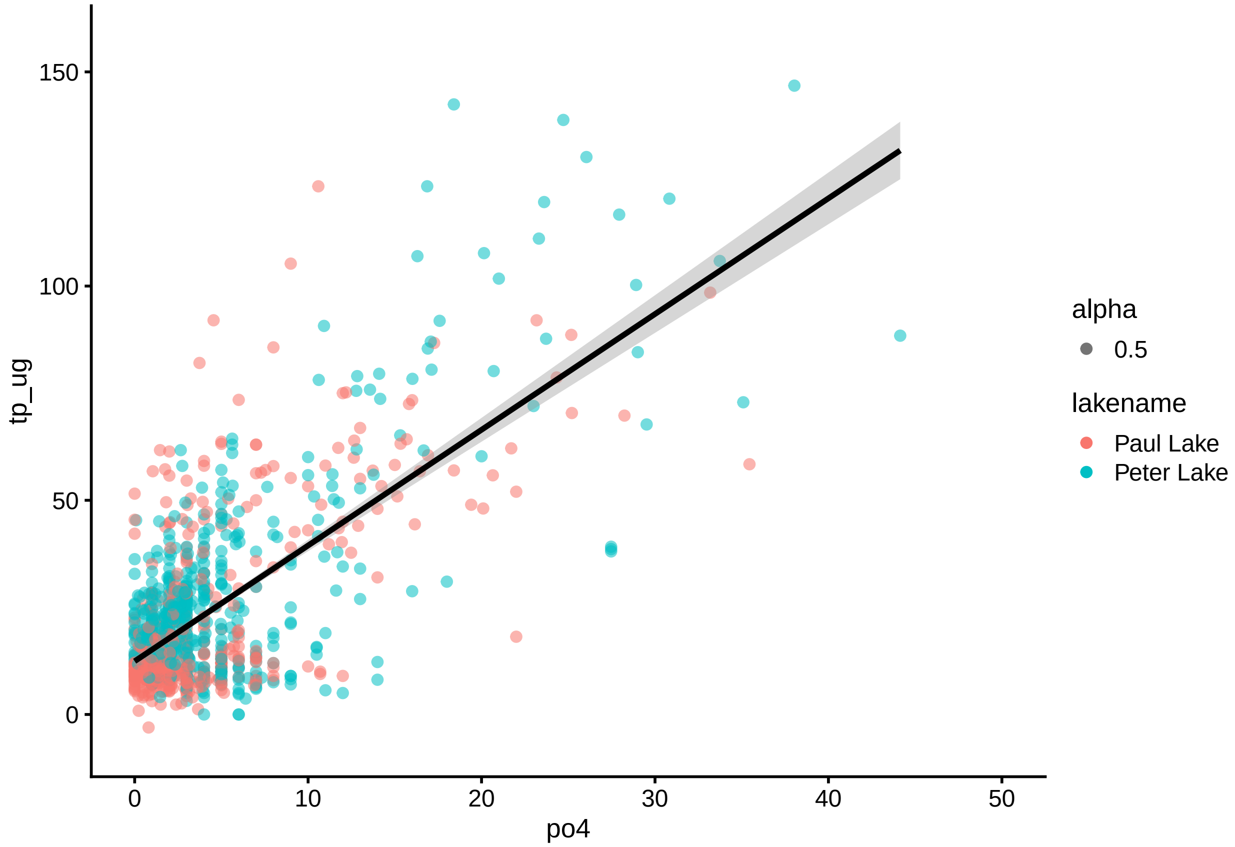
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
# 4
totalphosphorus <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug)) + geom_point(aes(color = la
    alpha = 0.5)) + geom_smooth(method = lm, color = "black") + xlim(0, 50) + mytheme
print(totalphosphorus)
```

```
## `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.

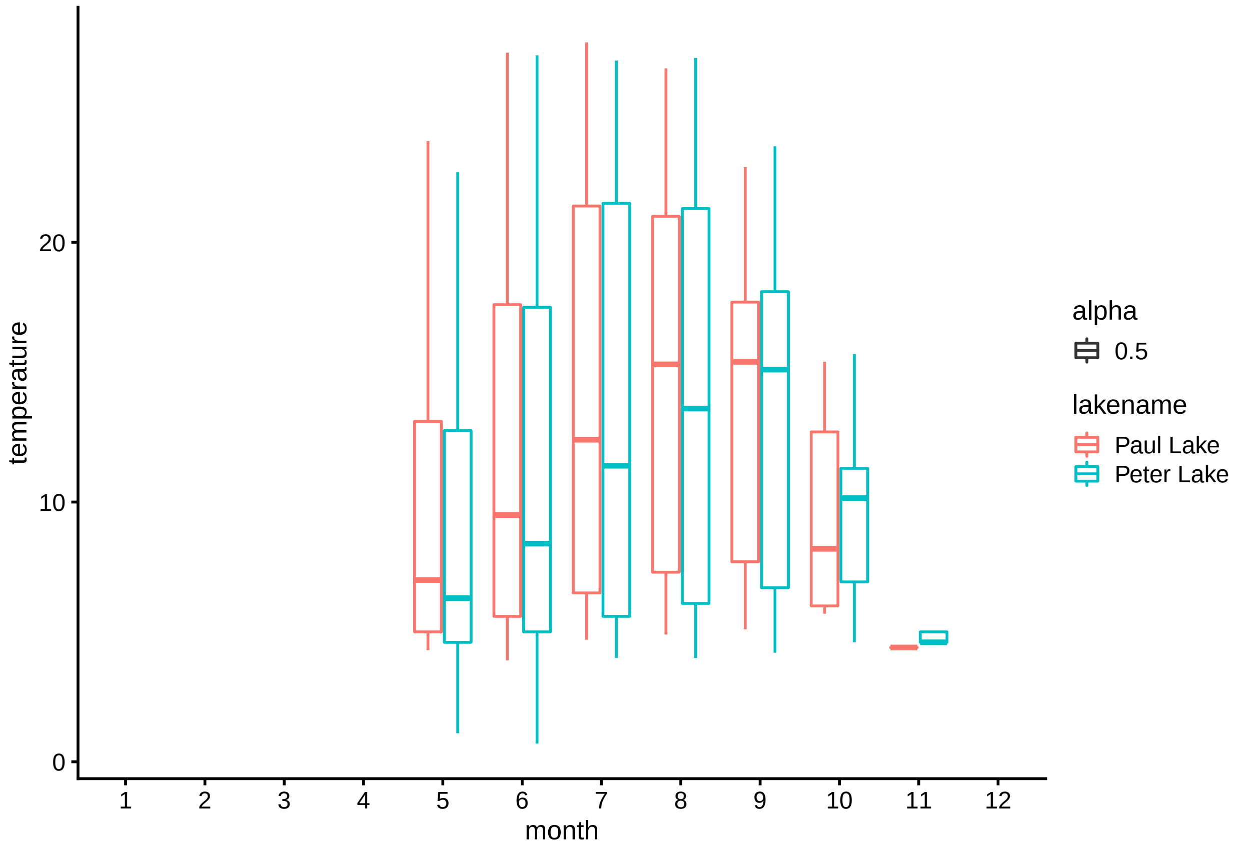
Tip: R has a build in variable called `month.abb` that returns a list of months; see https://r-lang.com/month-abb-in-r-with-example

```
# 5 make sure month is factor
PeterPaul.chem.nutrients$month <- factor(PeterPaul.chem.nutrients$month, levels = c(1:12))
class(PeterPaul.chem.nutrients$month)
```

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

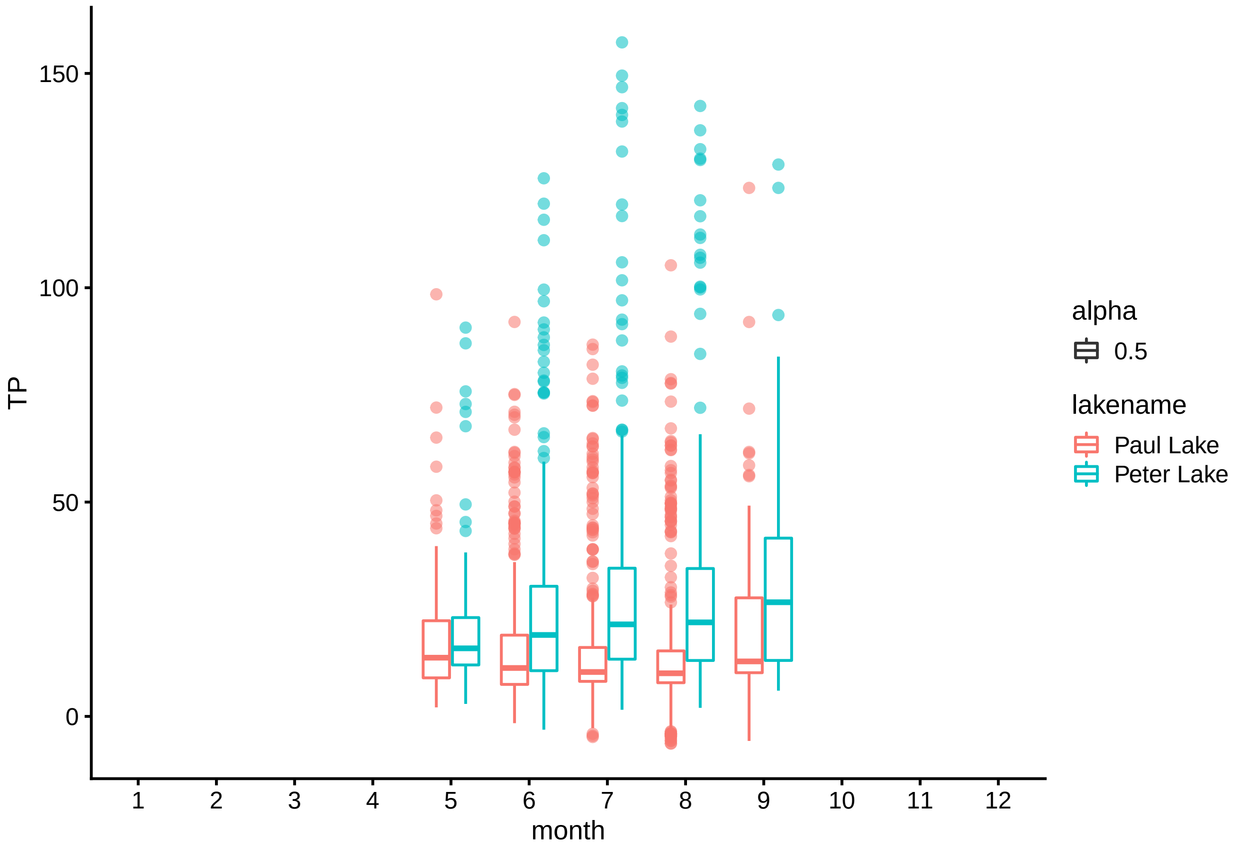
```
# month.abb[PeterPaul.chem.nutrients$month] draw three plots
temp.box <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C)) +
    geom_boxplot(aes(color = lakename, alpha = 0.5)) + labs(x = "month", y = "temperature") +
    scale_x_discrete(drop = FALSE) + mytheme
TP.box <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug)) + geom_boxplot(aes(color = lakenar
    alpha = 0.5)) + labs(x = "month", y = "TP") + scale_x_discrete(drop = FALSE) +
    mytheme
TN.box <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug)) + geom_boxplot(aes(color = lakenar
    alpha = 0.5)) + labs(x = "month", y = "TN") + scale_x_discrete(drop = FALSE) +
    mytheme
print(temp.box)
```

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



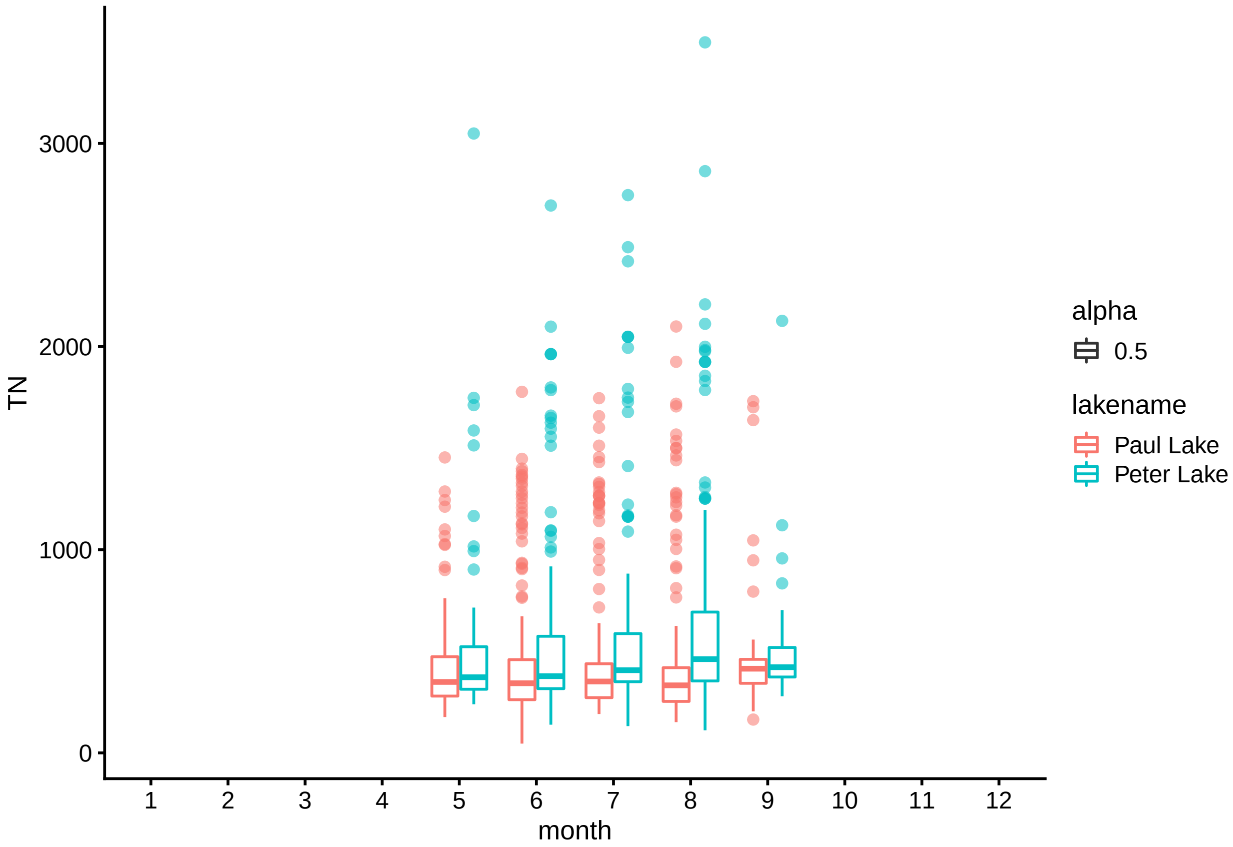
```
print(TP.box)
```

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



```
print(TN.box)
```

```
## 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?

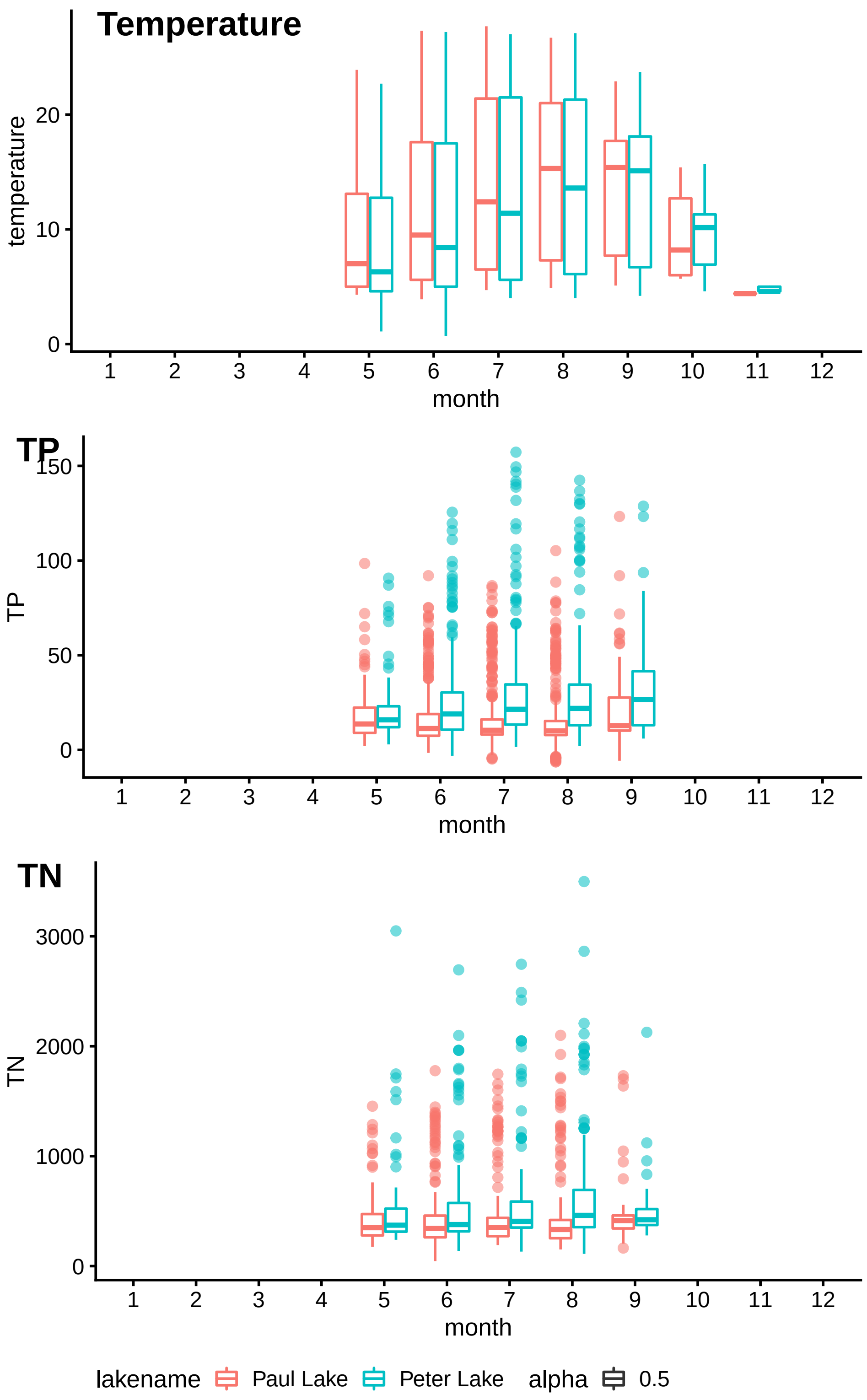
```
# put three plots in one
allthree <- plot_grid(temp.box + theme(legend.position = "none"), TP.box + theme(legend.position = "none
    TN.box + theme(legend.position = "bottom"), axis = "bt", labels = c("Temperature",
        "TP", "TN"), rel_heights = c(1, 1, 1.3), scale = c(1, 1, 1), ncol = 1)
```

```
## 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).
```

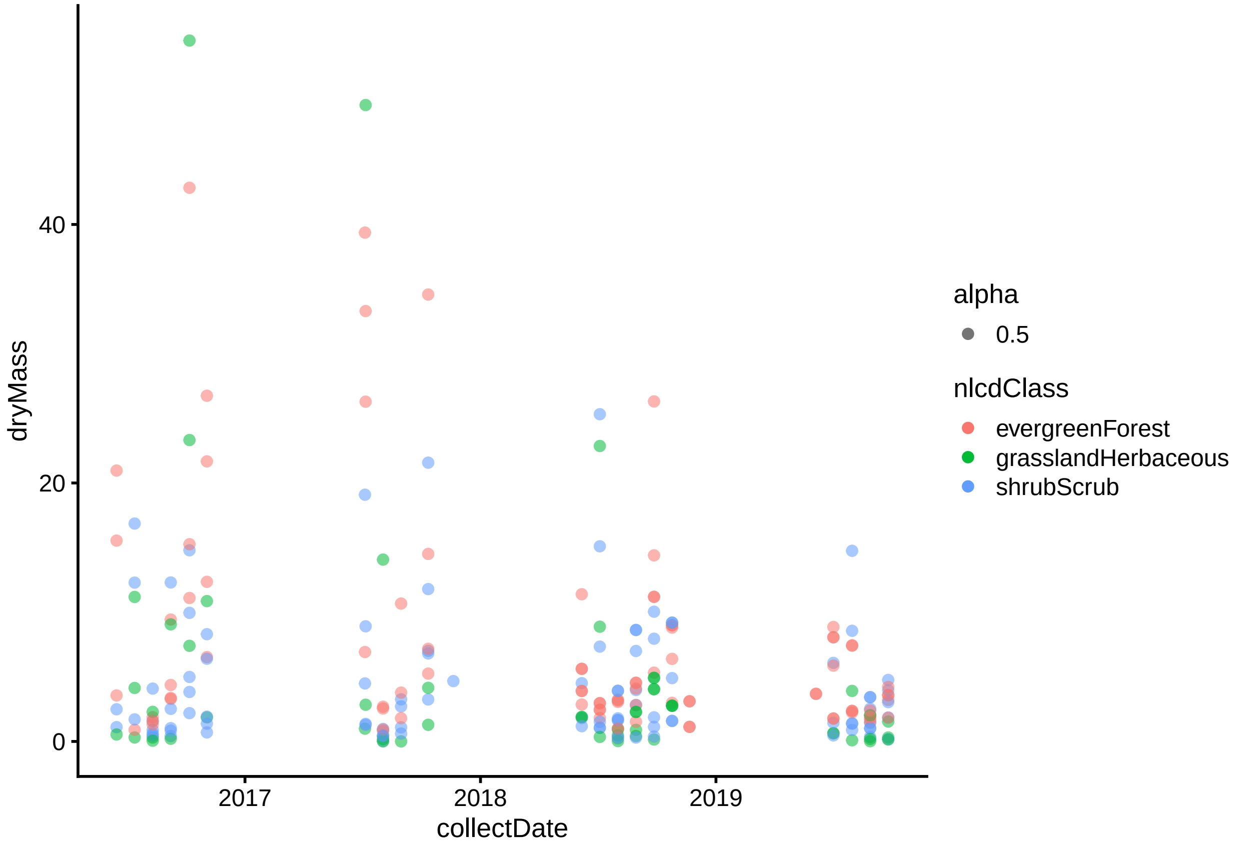
```
allthree
```



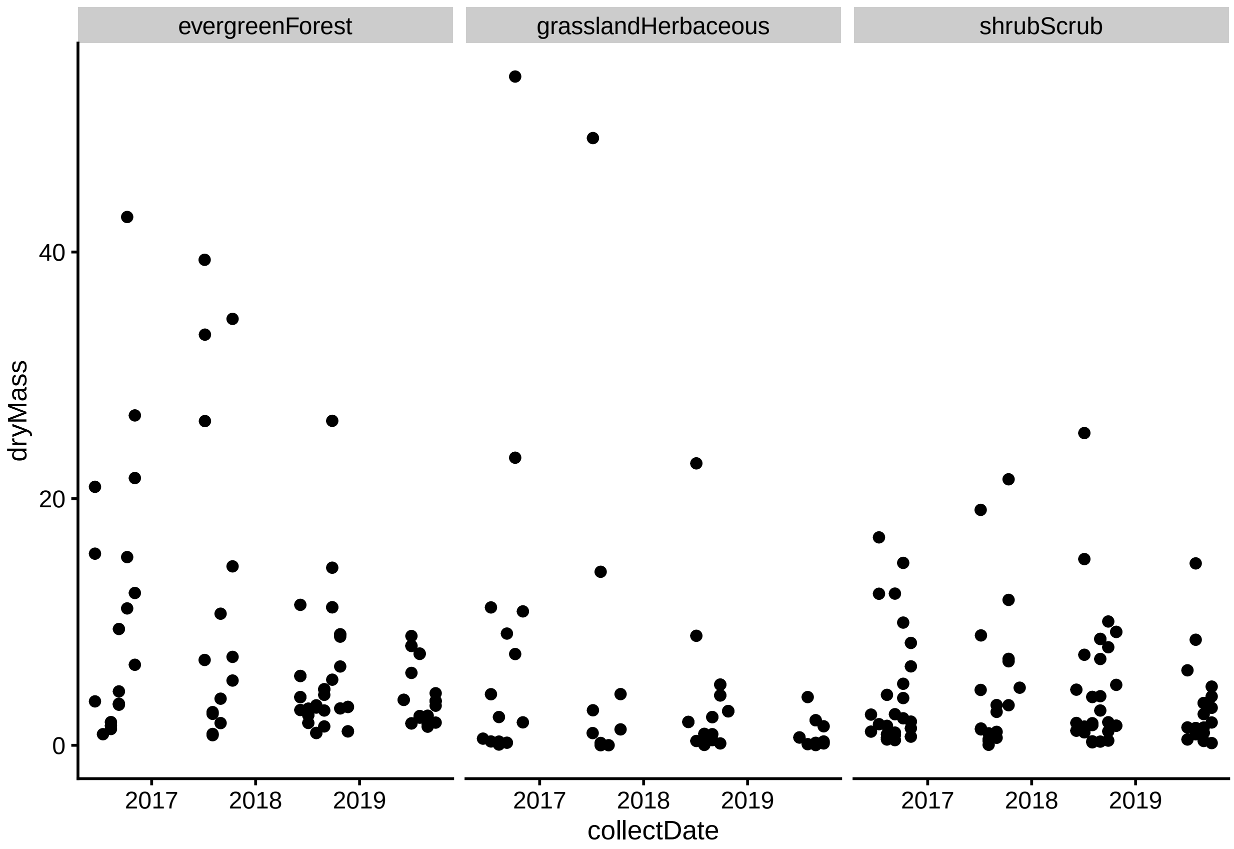
Answer: 1. Temperature:Both lake has similar median temperature but Peter lake's temperature varies a bit more.Temperature are highest and varies the most in summer(June, July and August). 2.TN:Peter lake has much greater range in TN and TP than Paul lake. Both lake have higher extremes in summer(June, July and August).

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
sublitter <- filter(NiwotRidgelitter, functionalGroup == "Needles")
plotsub <- ggplot(sublitter, aes(x = collectDate, y = dryMass)) + geom_point(aes(color = nlcdClass,
    alpha = 0.5)) + mytheme
print(plotsub)
```



```
# 7
facetsub <- ggplot(sublitter, aes(x = collectDate, y = dryMass)) + geom_point() +
    facet_wrap(vars(nlcdClass), ncol = 3) + mytheme
print(facetsub)
```



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

Answer: the second one is more effective.Beacuse though the first one has more colors, its cluster together so its hard to distinguish the trend of one specfic land use's changes with time. The second one has only black and white, but its more clear to see each land use's changes with time, and can easily compare each years's land use portion.