Assignment 5: Data Visualization

Chenjia Liu

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 1.9.2     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 C:/Users/15638/Desktop/DUKE/FALL2023/ENV872/EDE_Fall2023
```

```r
library(ggridges)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(RColorBrewer)
library(colormap)
library(ggthemes)
library(cowplot)
```

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

```r
here()
```

```
## [1] "C:/Users/15638/Desktop/DUKE/FALL2023/ENV872/EDE_Fall2023"
```

```r
processed_data = "Data/Processed_KEY"

NTL_LTER <- read.csv(
  here(processed_data,
       "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

NEON_NIWO_Litter_mass_trap <- read.csv(
  here(processed_data,
       "NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)
#2
NTL_LTER$sampledate <- ymd(NTL_LTER$sampledate)
NEON_NIWO_Litter_mass_trap$collectDate <- ymd(NEON_NIWO_Litter_mass_trap$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_base() +
  theme(
    line = element_line(
      color='white',
      linewidth =0.5
    ),
    legend.background = element_rect(
      color='grey',
      fill = 'white'
    ),
    legend.title = element_text(
      color='black'
    ),
    legend.position = "top",
    plot.background = element_rect(
      color='black'
    ),
    panel.grid.major =element_line(
      color = 'white',
    ),
    panel.background = element_rect(
      fill = "lightgray")



  )
```

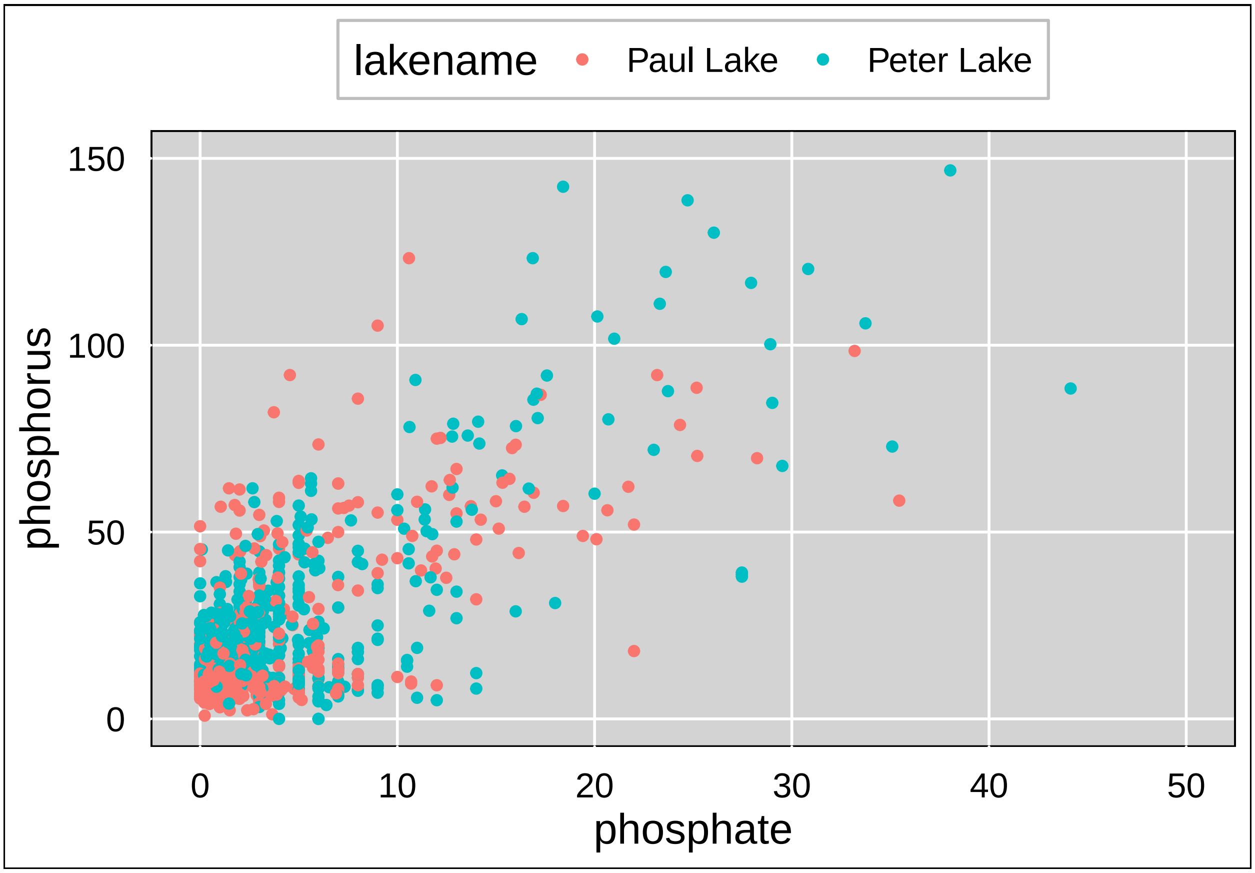
Create graphs

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(NTL_LTER, aes(x=po4,y=tp_ug,color=lakename))+
  geom_point()+
  xlab("phosphate")+
  ylab("phosphorus")+
  xlim(0,50)+
  ylim(0,150)+
  mytheme
```

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



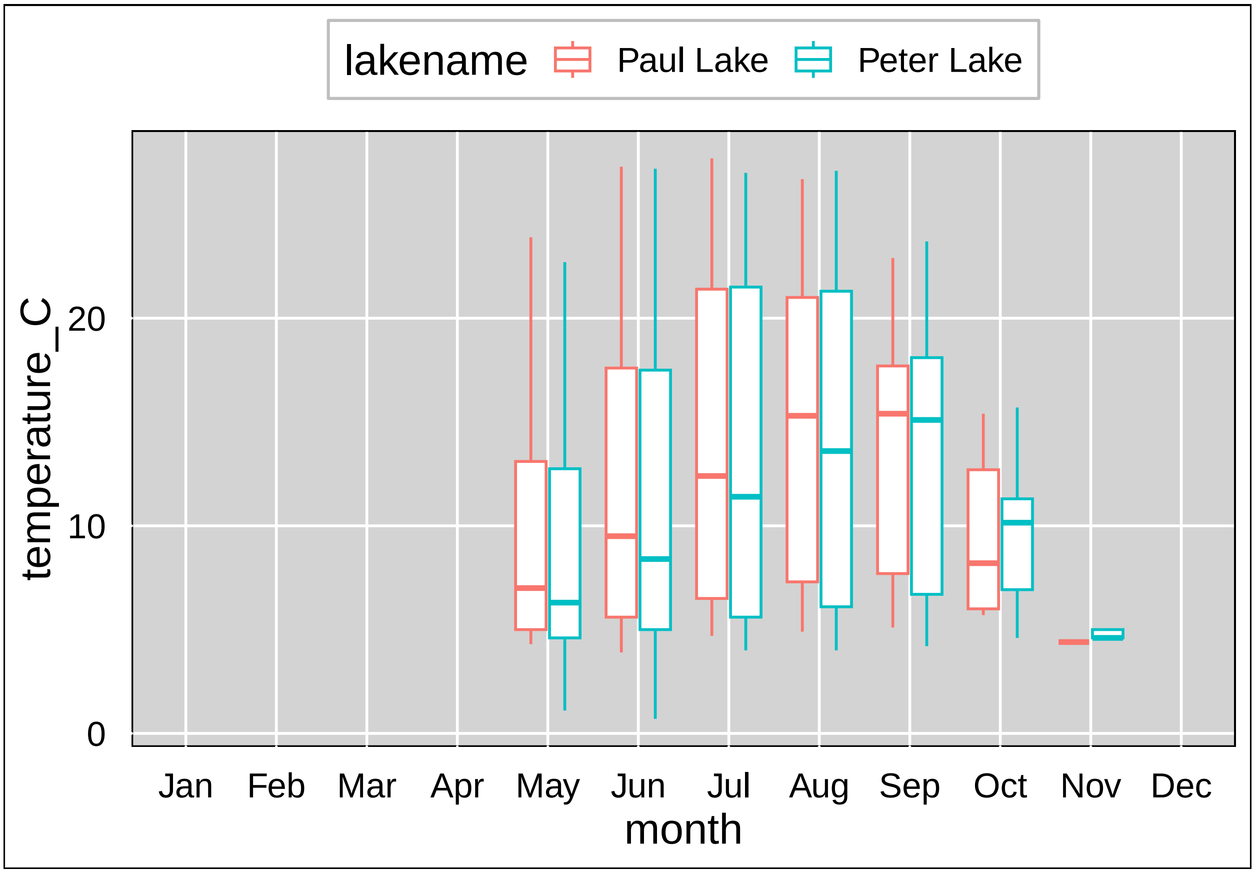
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * 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
Temperature_plot<- ggplot(NTL_LTER, aes(x = factor(NTL_LTER$month,levels = 1:12, labels = month.abb), y=
  scale_x_discrete(name="month",drop=FALSE)+
  geom_boxplot()+
  mytheme
print(Temperature_plot)
```

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

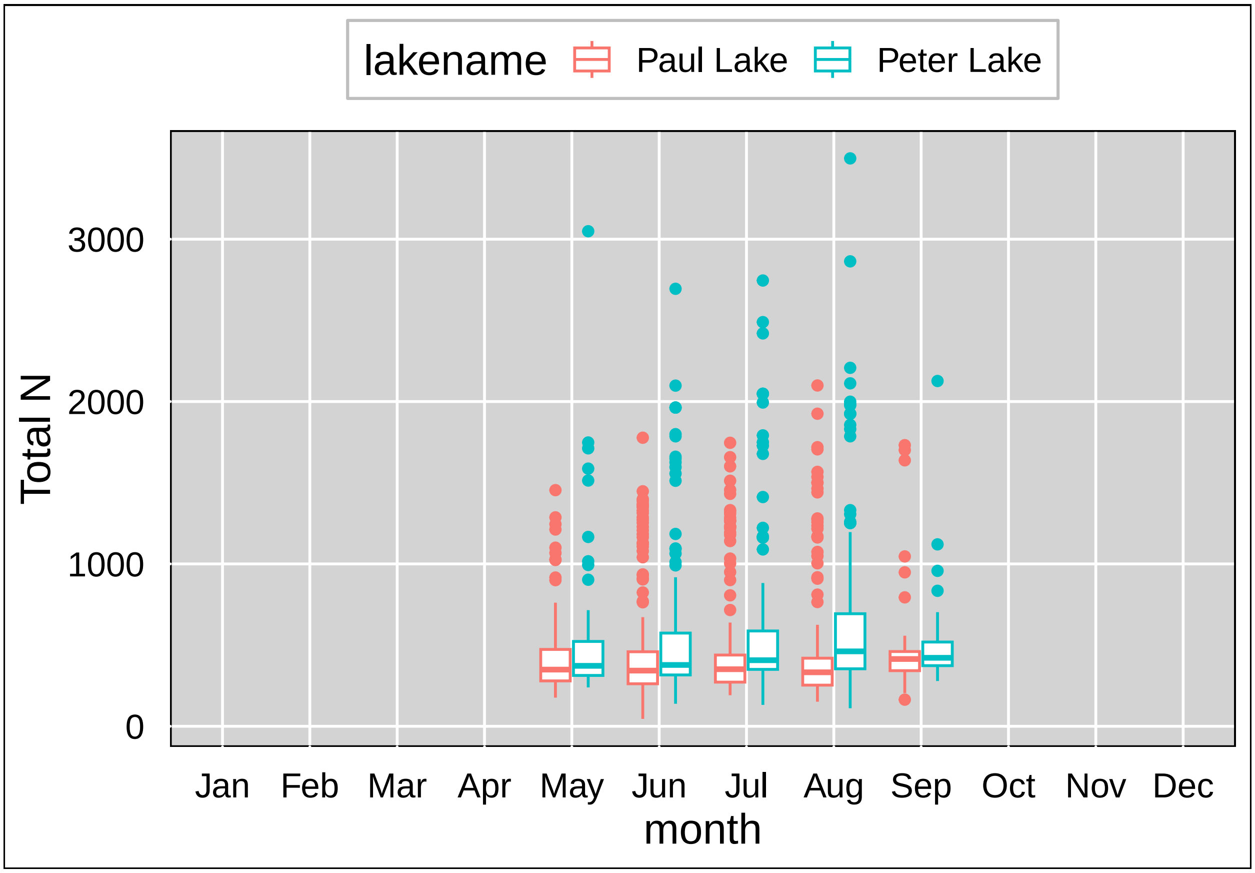
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



```
TN_plot<-ggplot(NTL_LTER, aes(x = factor(NTL_LTER$month,levels = 1:12, labels = month.abb), y=tn_ug, col
  scale_x_discrete(name="month",drop=FALSE)+
  geom_boxplot()+
  labs(y = "Total N")+
  mytheme
print(TN_plot)
```

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

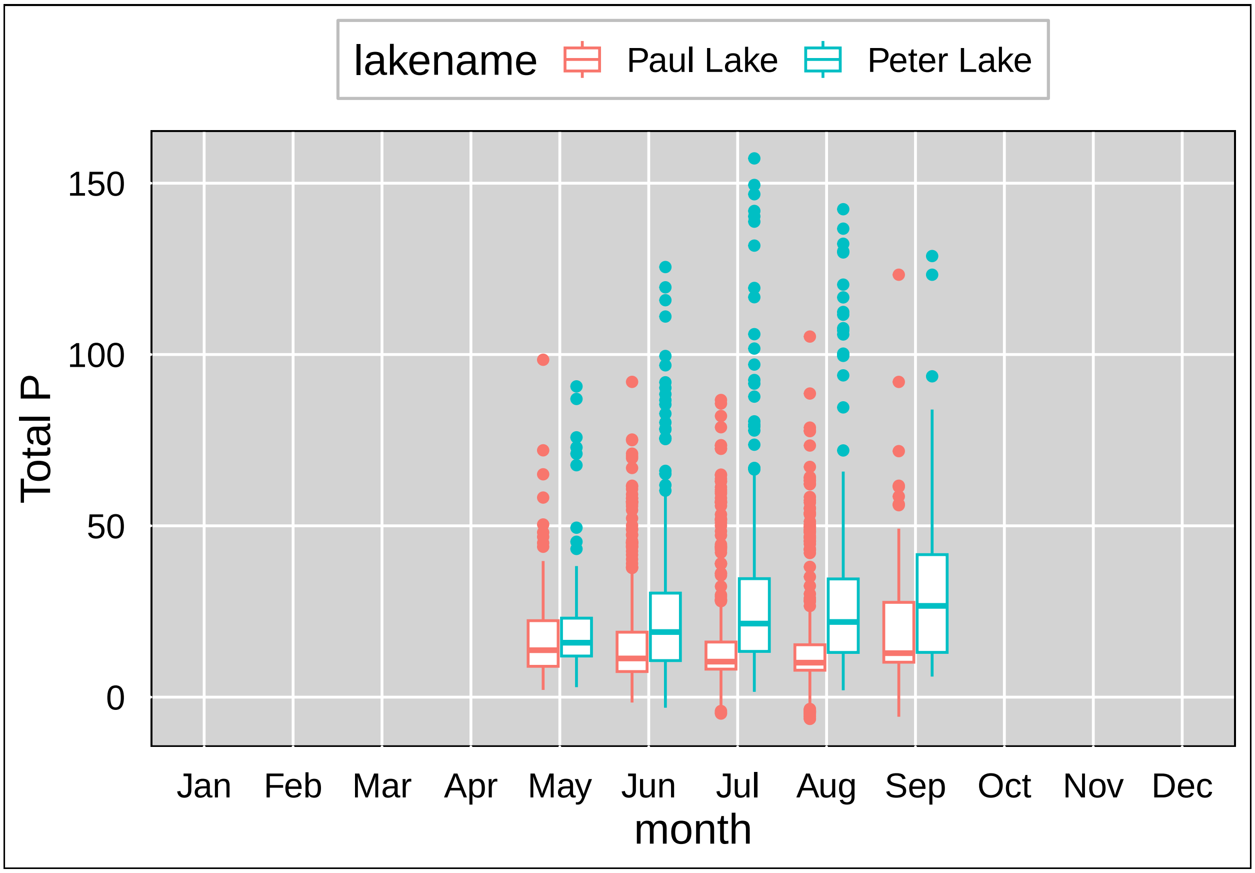
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```



```
TP_plot<-ggplot(NTL_LTER, aes(x = factor(NTL_LTER$month,levels = 1:12, labels = month.abb), y=tp_ug, col
  scale_x_discrete(name="month",drop=FALSE)+
  geom_boxplot()+
  labs(y = "Total P")+
  mytheme
print(TP_plot)
```

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

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



```
Temperature_plot2 <- Temperature_plot + theme(axis.title.x = element_blank())
TN_plot2 <- TN_plot + theme(legend.position = "none")+theme(axis.title.x = element_blank())
TP_plot2 <- TN_plot + theme(legend.position = "none")
plot_grid(Temperature_plot2, TN_plot2,TP_plot2,align = 'v',nrow=3)
```

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

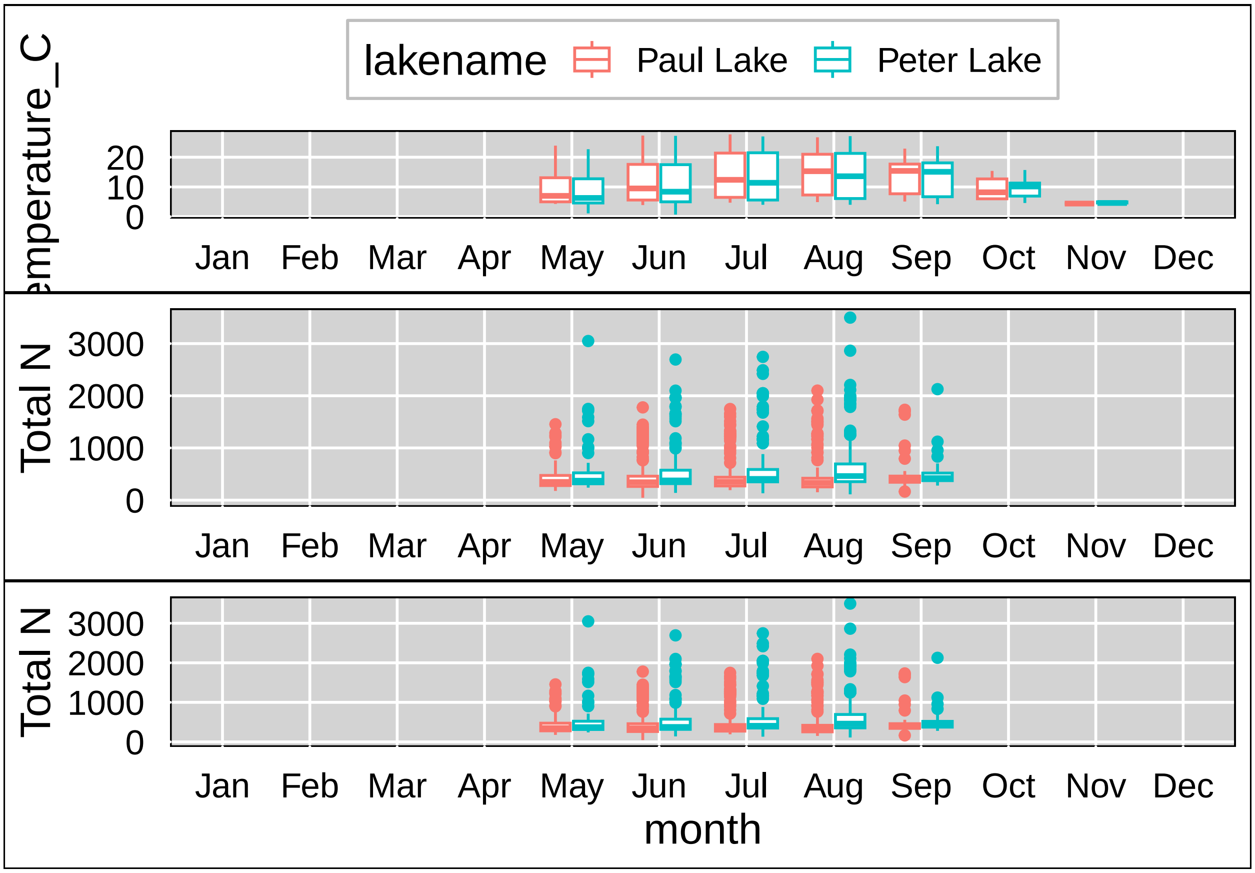
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

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

```
## Warning: Use of ‘NTL_LTER$month‘ is discouraged.
## i Use ‘month‘ instead.
```

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

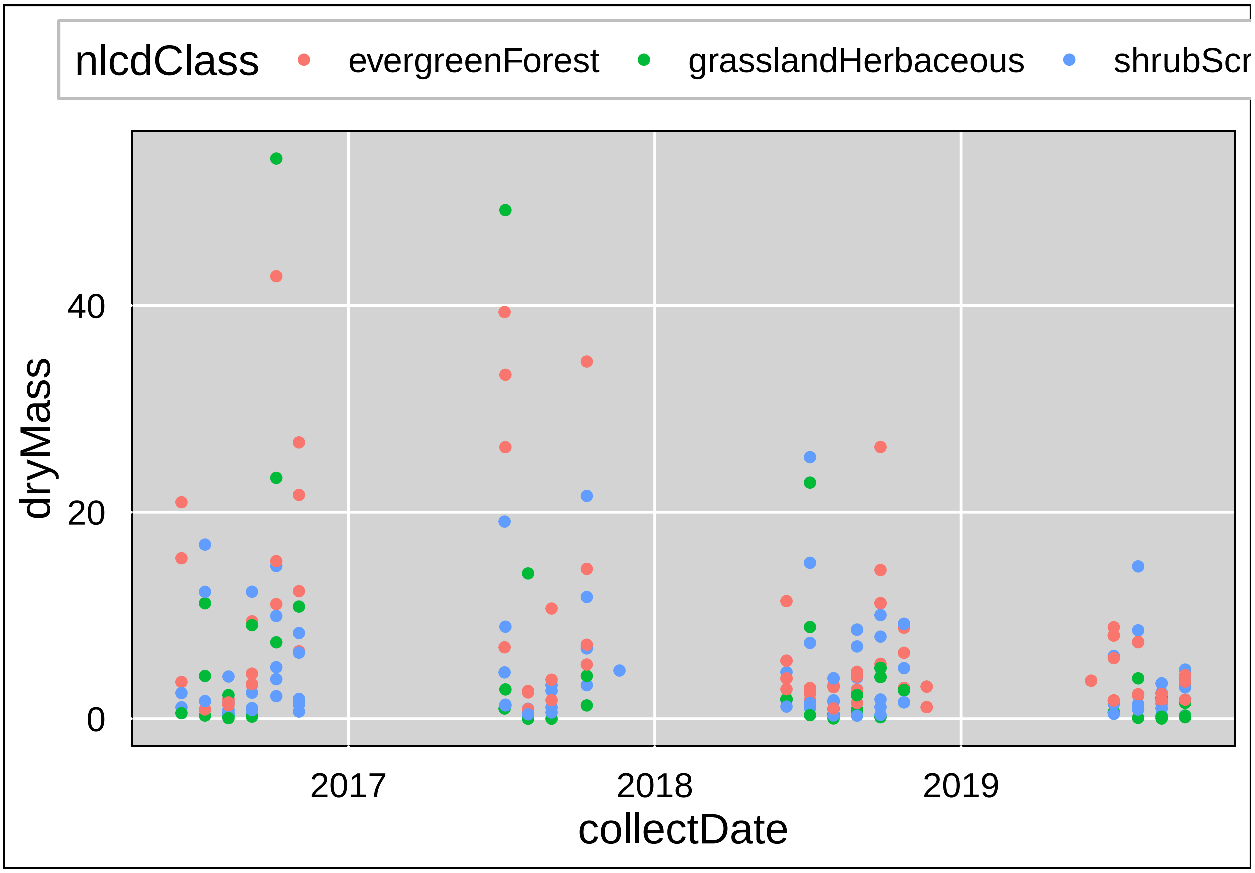


Question: What do you observe about the variables of interest over seasons and between lakes?

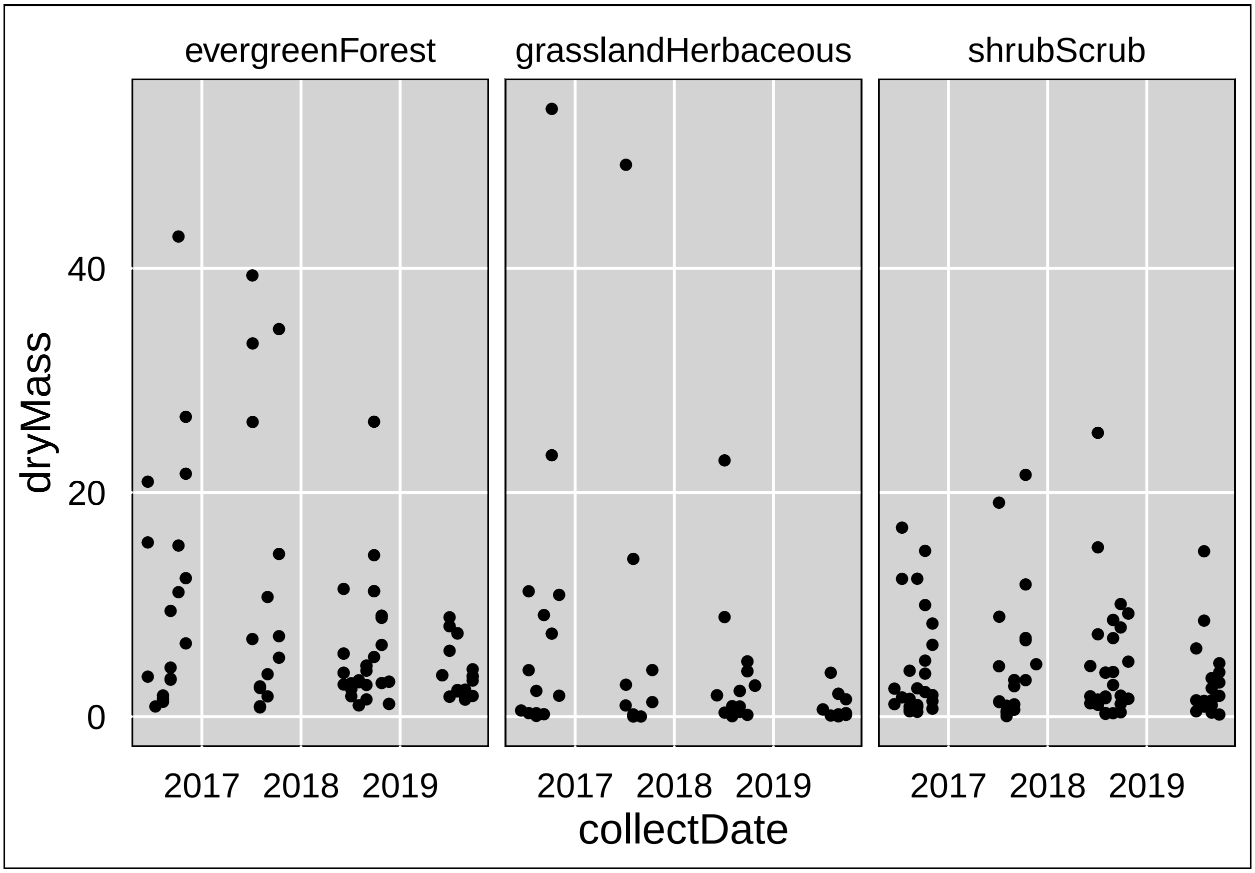
> Answer: The mean temperature is high in August and September(summer), and paul lake has higher temperature most of time. The total N doesn't fluctuate that much in different season, but Peter lake generally has higher total N.The total P is relatively low in summer, and Peter lake has a larger total P in every month.

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
Needles_plot <- NEON_NIWO_Litter_mass_trap %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
    ) +
  geom_point()+
  mytheme
print(Needles_plot)
```



```
#7
Needles_plot_2 <- NEON_NIWO_Litter_mass_trap %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      )) +
  geom_point()+
  mytheme +
  facet_wrap(vars(nlcdClass))
print(Needles_plot_2)
```



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

Answer: I think plot 7 is more effective since in the plot 6, when the point density is high, it becomes hard to diferentiate the color. When all data points are mixed, it's also hard to make comparison between years. In plot 7, we can easily make comparision within the same year to see which class has higher drymass, or we can also compare the drymass of the same class in different year