Assignment 5: Data Visualization

Miaojun Pang

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Monday, February 14 at 7:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and 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_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.

```
#1
getwd()
```

```
## [1] "C:/Users/mpang/Downloads"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- 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)
```

```
## Warning: package ’here’ was built under R version 4.2.3
```

```
## here() starts at C:/Users/mpang/Downloads
```

```
library(cowplot)
```

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

```
library(ggplot2)

NTL_LTER<- read.csv("/Users/mpang/OneDrive - Duke University/Desktop/Assignments/NTL-LTER_Lake_Chemistry
NEON_LITTER<- read.csv("/Users/mpang/OneDrive - Duke University/Desktop/Assignments/NEON_NIWO_Litter_mas

#2
NTL_LTER$sampledate<- as.Date(NTL_LTER$sampledate, "%Y-%m-%d")
NEON_LITTER$collectDate<- as.Date(NEON_LITTER$collectDate, "%Y-%m-%d")
```

Define your theme

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

```
#3
EDA_Theme <- theme_classic(base_size = 10) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
theme_set(EDA_Theme)
```

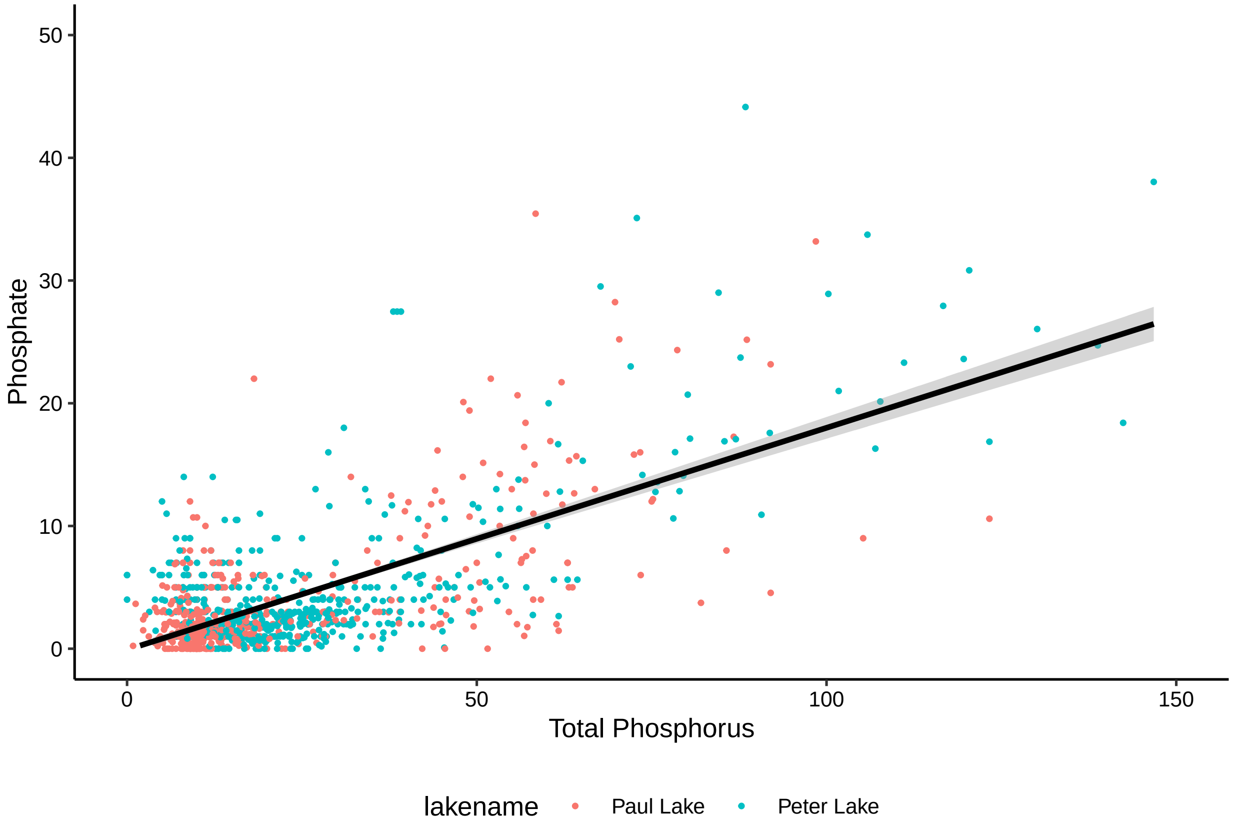
Create graphs

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 ylim()).

```
#4
PeterPaulPlot1<-
  ggplot(NTL_LTER, aes(x=tp_ug, y=po4, color= lakename))+
  geom_point(size=0.5)+
  geom_smooth(method = lm, color='black')+
   xlim(0, 150) +
  ylim(0, 50)+
  xlab("Total Phosphorus")+
  ylab("Phosphate")
print(PeterPaulPlot1)
```

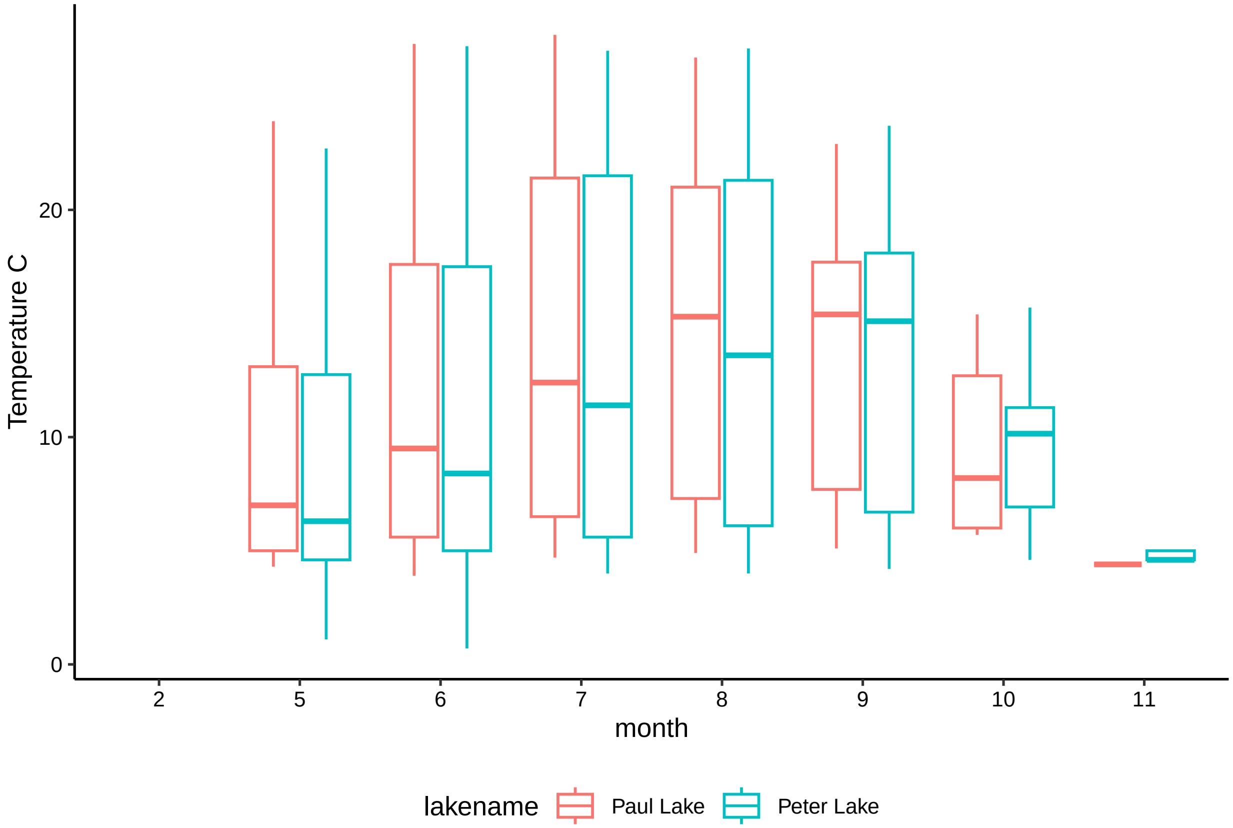
```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

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

## Warning: Removed 1 rows containing missing values (`geom_smooth()`).
```



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.

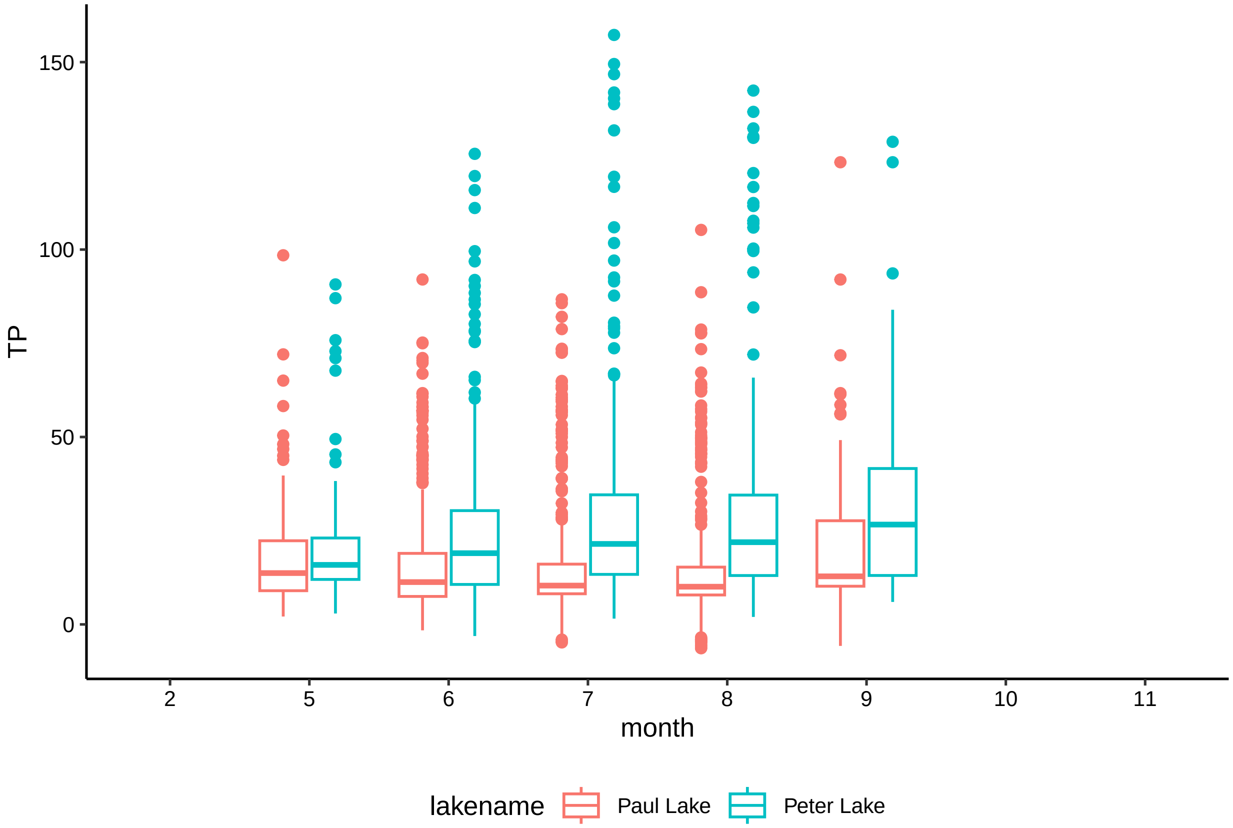
```
#5

NTL_LTER$month<- as.factor(NTL_LTER$month)

peterpaulplot2<-
  ggplot(NTL_LTER, aes(x=month, y=temperature_C))+
  geom_boxplot(aes(color= lakename))+
  ylab("Temperature C")
print(peterpaulplot2)
```

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



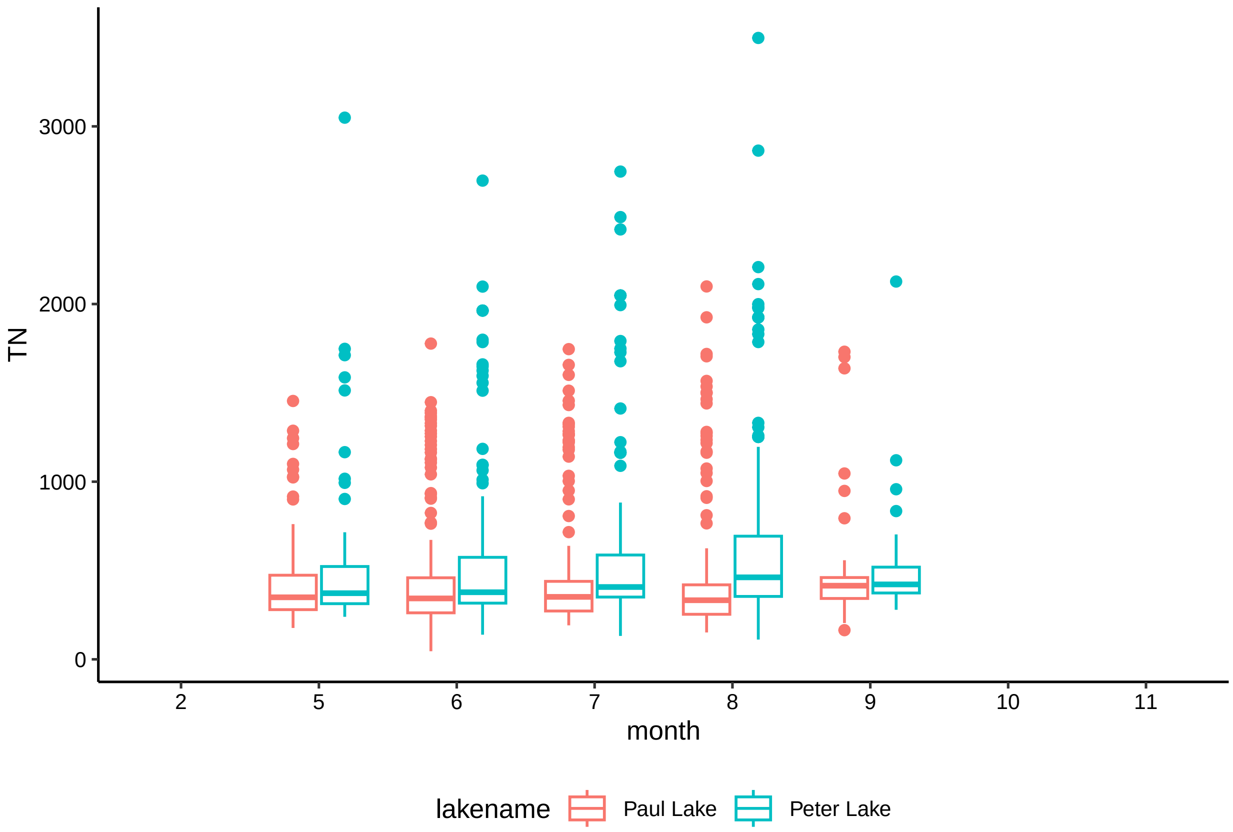
```
peterpaulplot3<-
  ggplot(NTL_LTER, aes(x=month, y=tp_ug))+
  geom_boxplot(aes(color=lakename))+
  ylab("TP")
  print(peterpaulplot3)
```

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



```
peterpaulplot4<-
  ggplot(NTL_LTER, aes(x=month, y=tn_ug))+
  geom_boxplot(aes(color=lakename))+
  ylab("TN")
print(peterpaulplot4)
```

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



```
legendpeterpaul<- get_legend(peterpaulplot4)
```

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

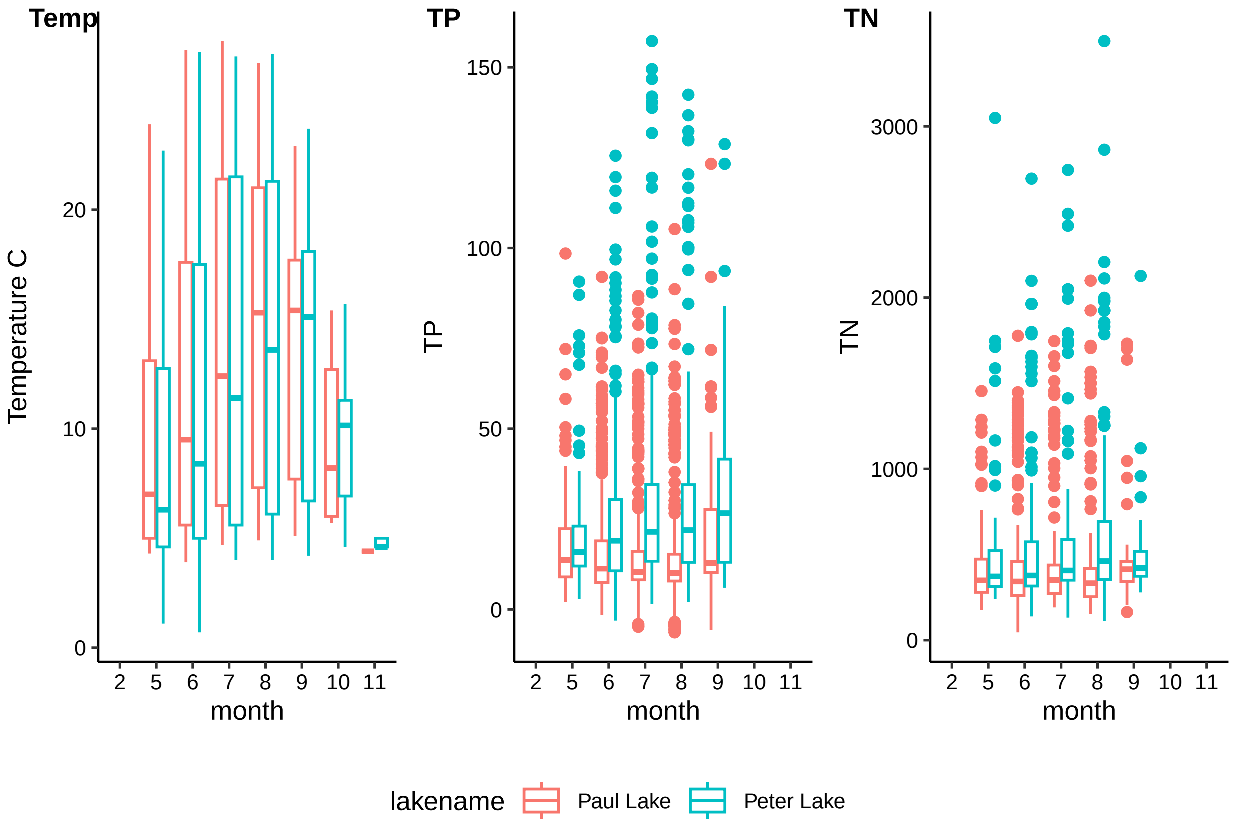
```
plotgrid1<-plot_grid(peterpaulplot2+ theme(legend.position = "none"),
                     peterpaulplot3+ theme(legend.position = "none"),
                     peterpaulplot4+ theme(legend.position = "none"),
                     rel_heights = c(3, 3,3),
                     nrow = 1,
          align = "v", labels= c("Temp", "TP", "TN"),
          label_size = 10)
```

```
## 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()‘).
```

```
plotgridfinal<-plot_grid(plotgrid1,nrow=2,legendpeterpaul,
                         rel_heights= c(3,0.5)
                        )
print(plotgridfinal)
```

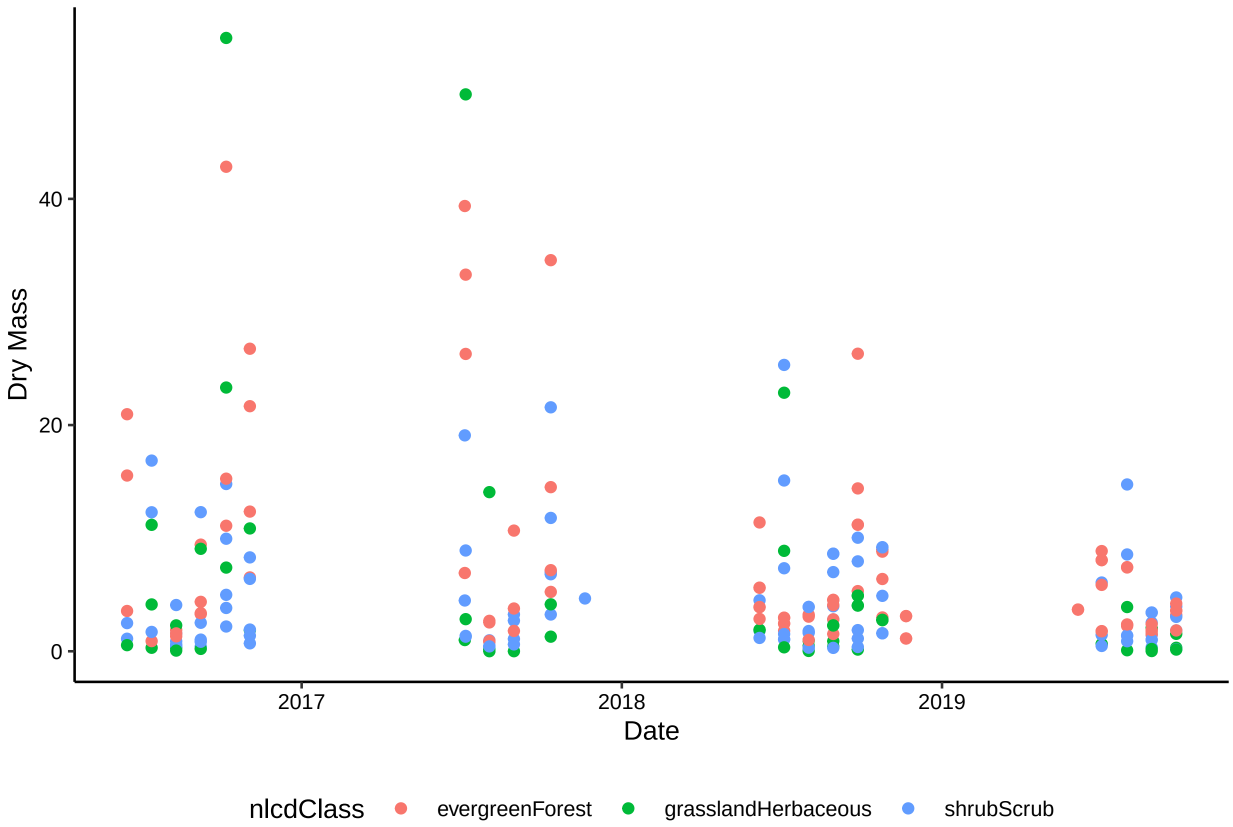


Question: What do you observe about the variables of interest over seasons and between lakes?

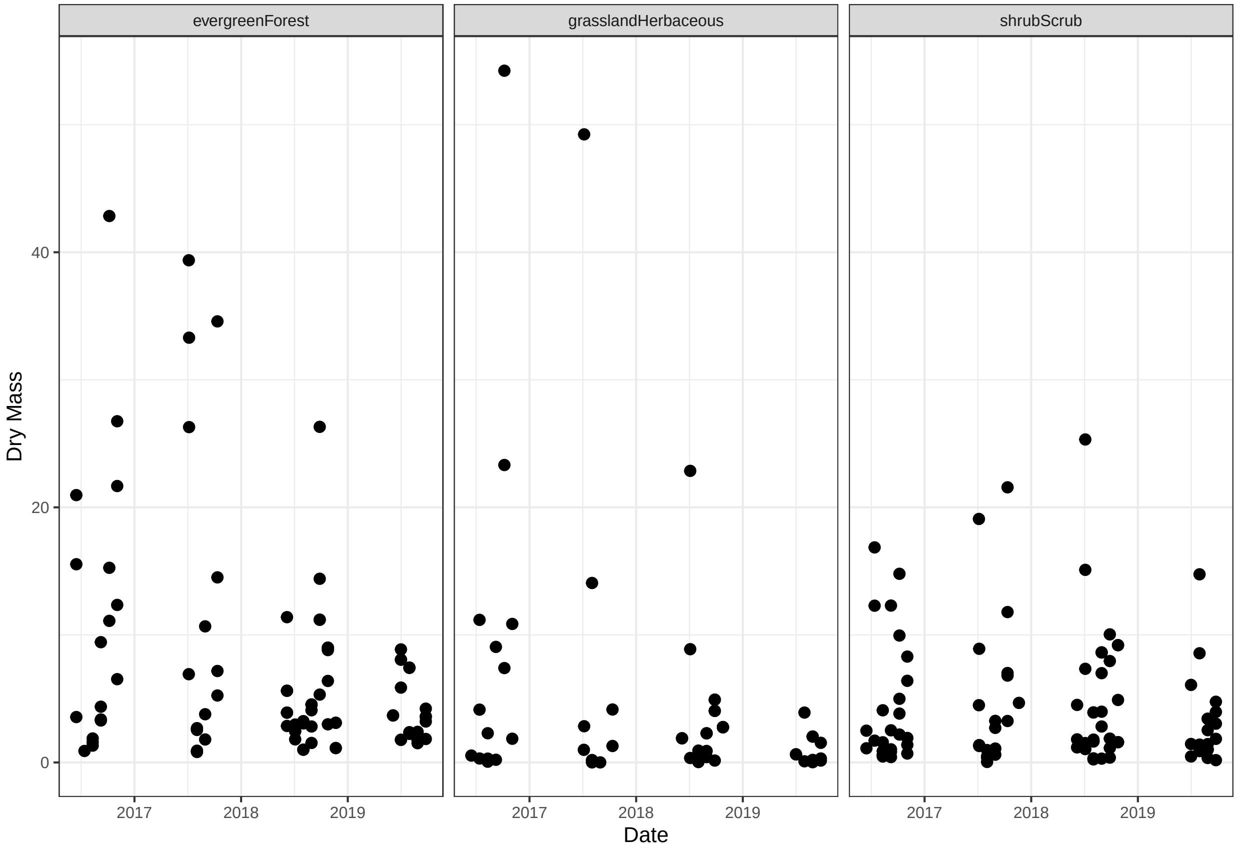
> Answer: The temperature across both lakes remains relatively consistent throughout the various seasons, experiencing its lowest average in November and peaking in August or September. Peter Lake consistently exhibits higher average total phosphorus (TP) levels compared to Paul Lake, along with more pronounced TP outliers. The TP levels reach their peak during the summer months (July, August, September). Similarly, Peter Lake demonstrates higher average total nitrogen (TN) levels than Paul Lake. While TN levels are generally stable throughout the year, they show a notable increase in August or September.

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
Niwot_Ridge_Plot1<-
  ggplot(subset(NEON_LITTER, functionalGroup=="Needles"),
         aes(x=collectDate, y=dryMass))+
  geom_point(aes(color=nlcdClass))+
  xlab("Date")+
  ylab("Dry Mass")
print(Niwot_Ridge_Plot1)
```



```
#7
Niwot_Ridge_Plot2<-
  ggplot(subset(NEON_LITTER, functionalGroup=="Needles"),
         aes(x=collectDate, y=dryMass))+
  geom_point()+
  facet_wrap(vars(nlcdClass))+
  xlab("Date")+
  ylab("Dry Mass")+
  theme_bw(base_size = 8)
print(Niwot_Ridge_Plot2)
```



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

> Answer: I find the faceted plot to be more effective due to its separation of each class into distinct sections. This organization eliminates the need to color-code each data point, simplifying the visual presentation. The absence of colors allows for an easier comparison and contrast between different classes, enhancing focus and clarity in the analysis.