Assignment 5: Data Visualization

Andrea Gonzalez Natera

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
getwd()
```

```
## [1] "C:/Users/andre/OneDrive/Documents/NSOE-MEM 2019-2021/Spring 2021/Data Analytics/Environmental_Da
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

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

```
LakeChemistry.PeterPaul <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Proces

LakeNutrients.PeterPaul <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processe

NiwotRidge <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = 

#2
LakeChemistry.PeterPaul$sampledate <-as.Date(LakeChemistry.PeterPaul$sampledate, format = "%Y-%m-%d")

LakeNutrients.PeterPaul$sampledate <-as.Date(LakeNutrients.PeterPaul$sampledate, format = "%Y-%m-%d")

LakeChemistry.PeterPaul$month <- month(LakeChemistry.PeterPaul$month, label = TRUE)

NiwotRidge$collectDate <- as.Date(NiwotRidge$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

```
mytheme <- theme_light(base_size = 15) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom",
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12),
        plot.title = element_text(hjust = 0.5))
```

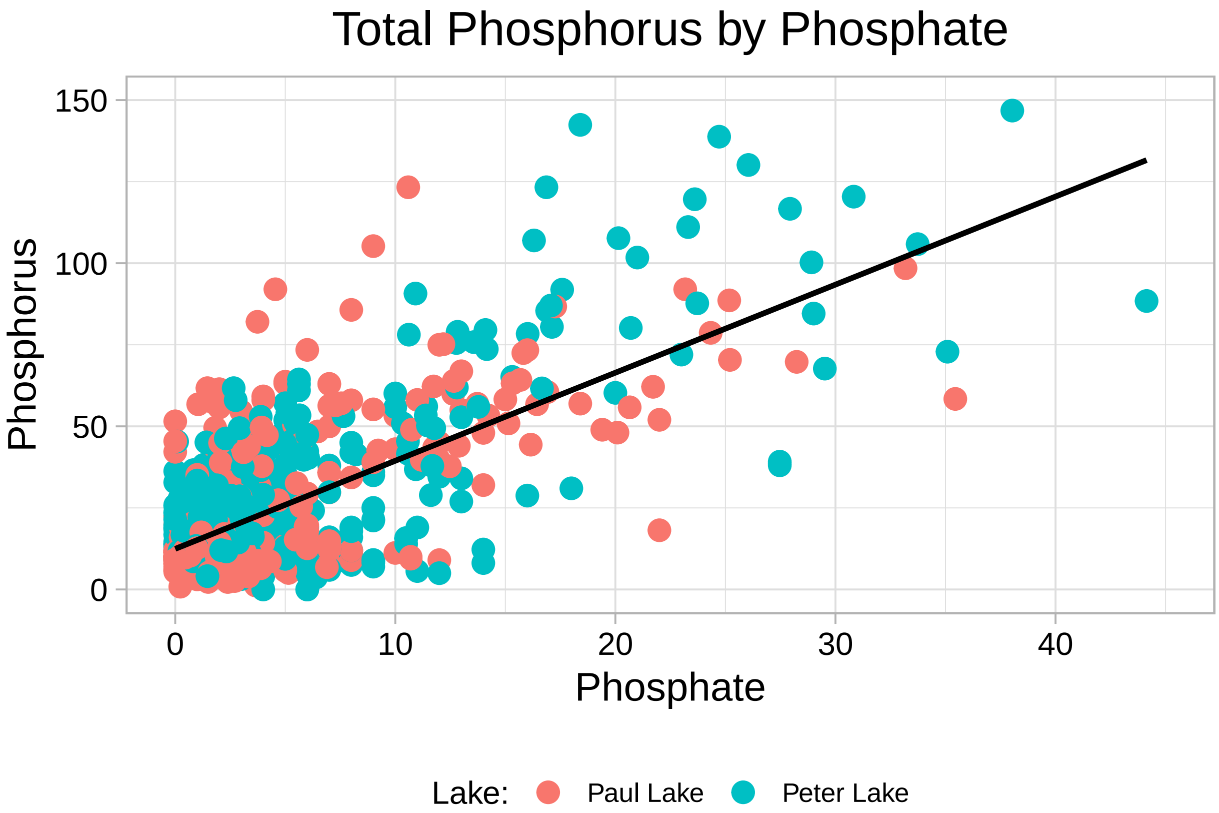
Create graphs

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.

```
plot1 <- ggplot(LakeChemistry.PeterPaul,aes(x = po4, y= tp_ug,
  color = lakename))+
  geom_point(size = 3.5) +
  xlim(0,45) +
  ylim(0,150) +
  labs(y = "Phosphorus", x= "Phosphate", color = "Lake:") +
  geom_smooth(method = lm, se=FALSE, color="black", aes(group = 1)) +
  ggtitle("Total Phosphorus by Phosphate")+
  mytheme
print(plot1)
```

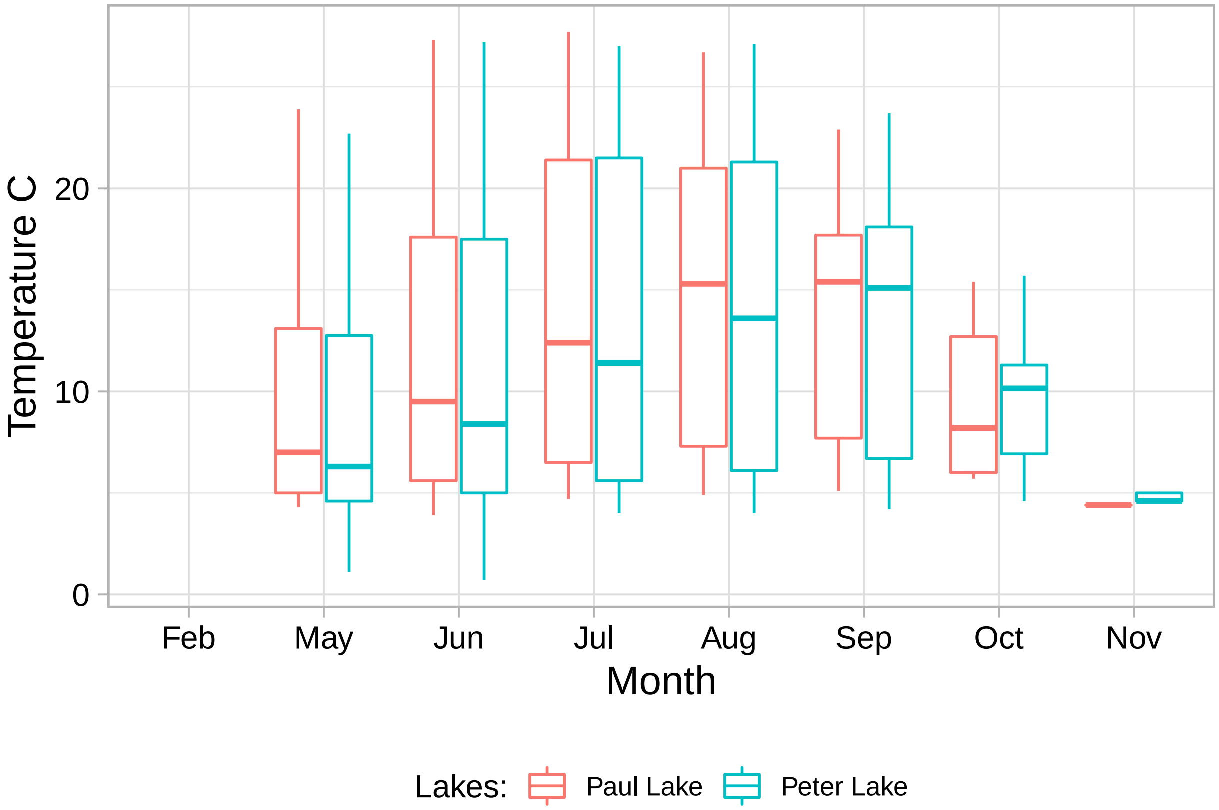
```
## `geom_smooth()` using formula 'y ~ x'

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

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



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.

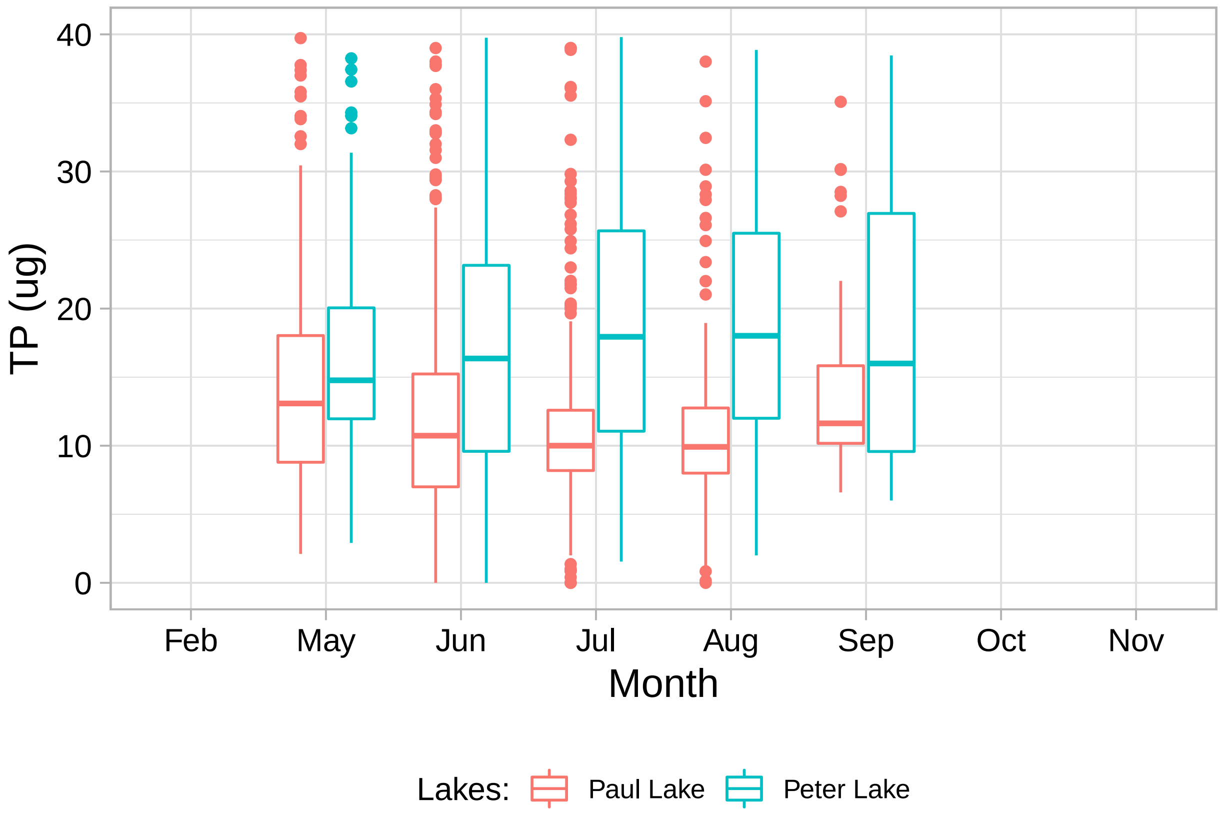
```
boxplot1 <- ggplot(LakeChemistry.PeterPaul,aes(x = month, y =temperature_C)) +
 geom_boxplot (aes(color = lakename))+
  labs(y = "Temperature C", x= "Month", color = "Lakes:")+
  mytheme
print(boxplot1)
```

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



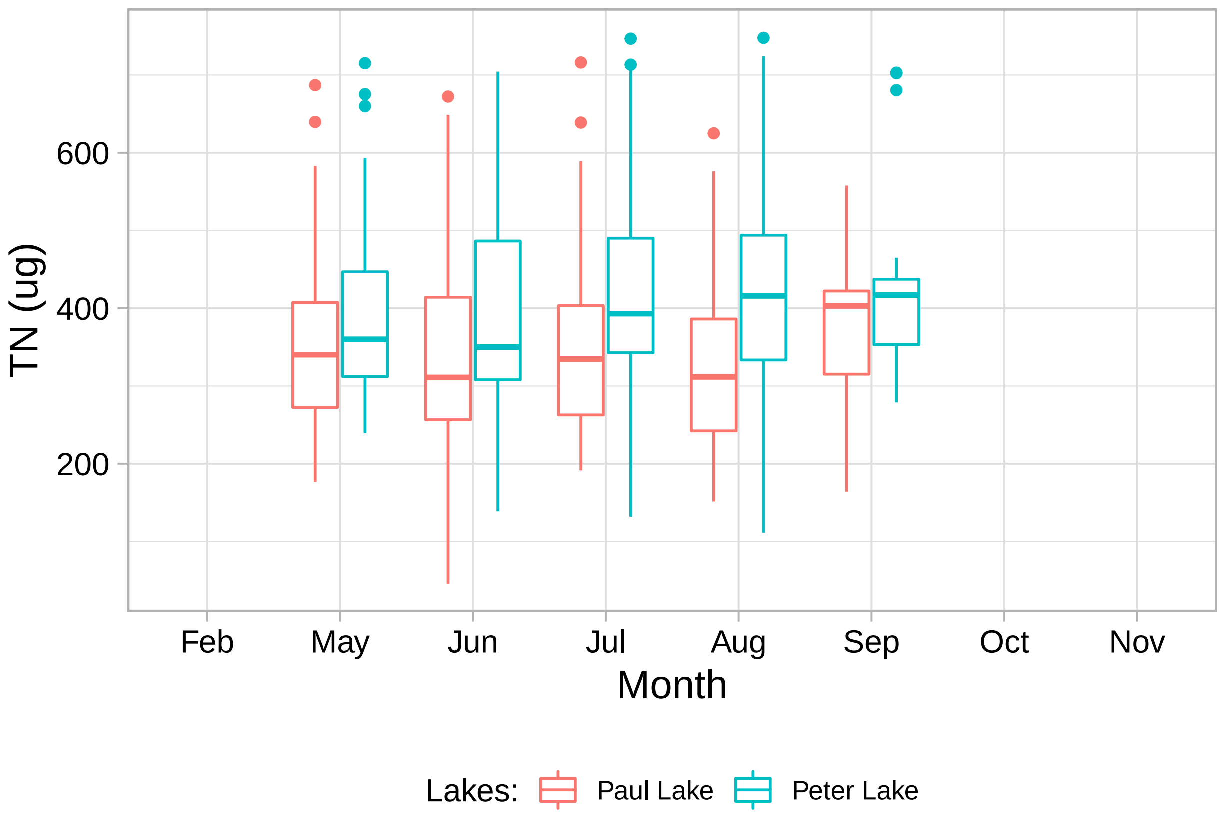
```
boxplot2 <- ggplot(LakeChemistry.PeterPaul,aes(x = month, y =tp_ug)) +
  geom_boxplot(aes(color = lakename))+
  labs(y = "TP (ug)", x= "Month", color = "Lakes:")+
   ylim(0,40) +
  mytheme
print(boxplot2)
```

```
## Warning: Removed 21110 rows containing non-finite values (stat_boxplot).
```



```
boxplot3 <- ggplot(LakeChemistry.PeterPaul,aes(x = month, y =tn_ug)) +
  geom_boxplot(aes(color = lakename))+
  labs(y = "TN (ug)", x= "Month", color = "Lakes:")+
  ylim(45,750) +
  mytheme
print(boxplot3)
```

```
## Warning: Removed 21785 rows containing non-finite values (stat_boxplot).
```

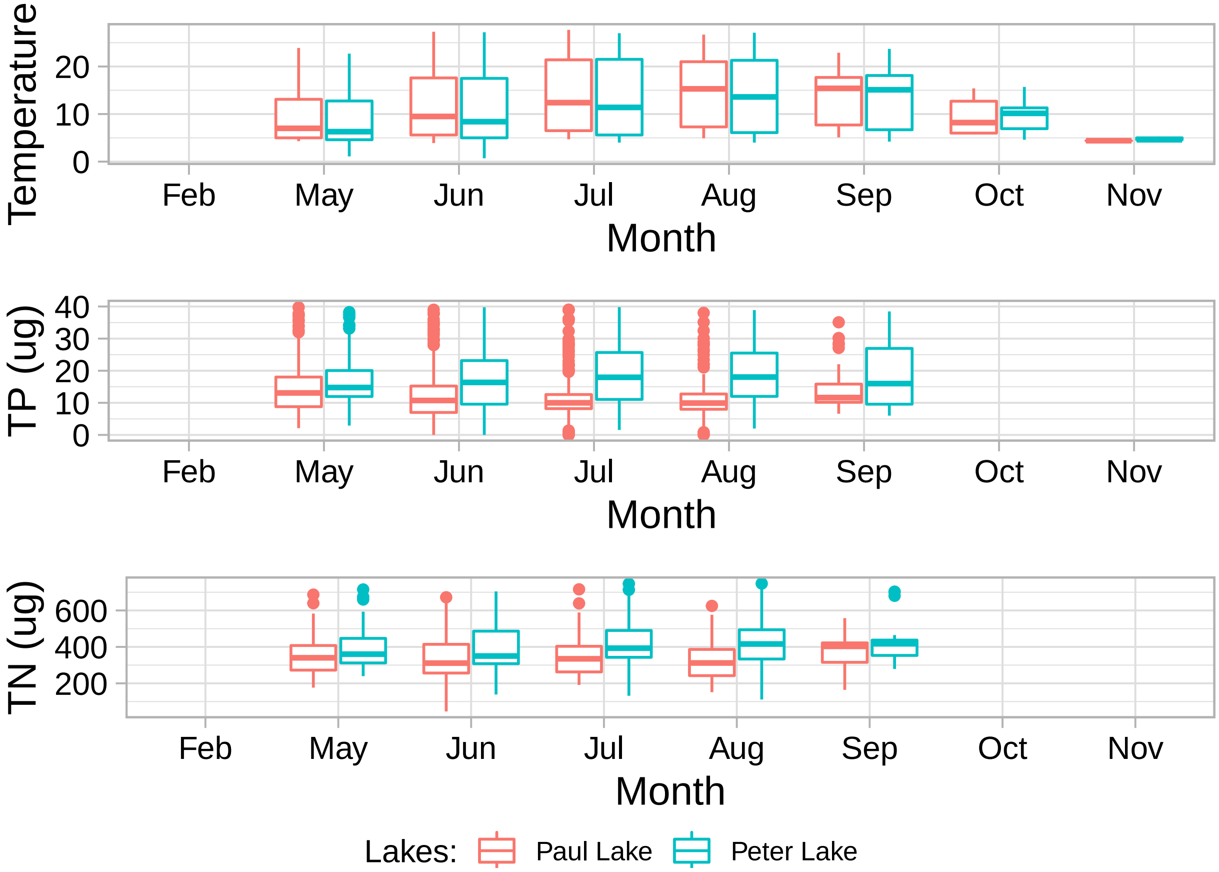


```
legend_bp1 <- get_legend(boxplot1 +
                           guides(color = guide_legend(nrow = 1))
                         + theme(legend.position = "bottom")+
                           labs(color = "Lakes:"))
```

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

```
plot_grid(boxplot1 + theme(legend.position = "none"),
          boxplot2 + theme(legend.position = "none"),
          boxplot3 + theme(legend.position = "none"),
          legend_bp1,
          align = 'v',
          nrow = 4,
          ncol = 1,
          rel_heights = c(2,2,2,0.25),
          axis = "b")
```

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

## Warning: Removed 21110 rows containing non-finite values (stat_boxplot).

## Warning: Removed 21785 rows containing non-finite values (stat_boxplot).
```



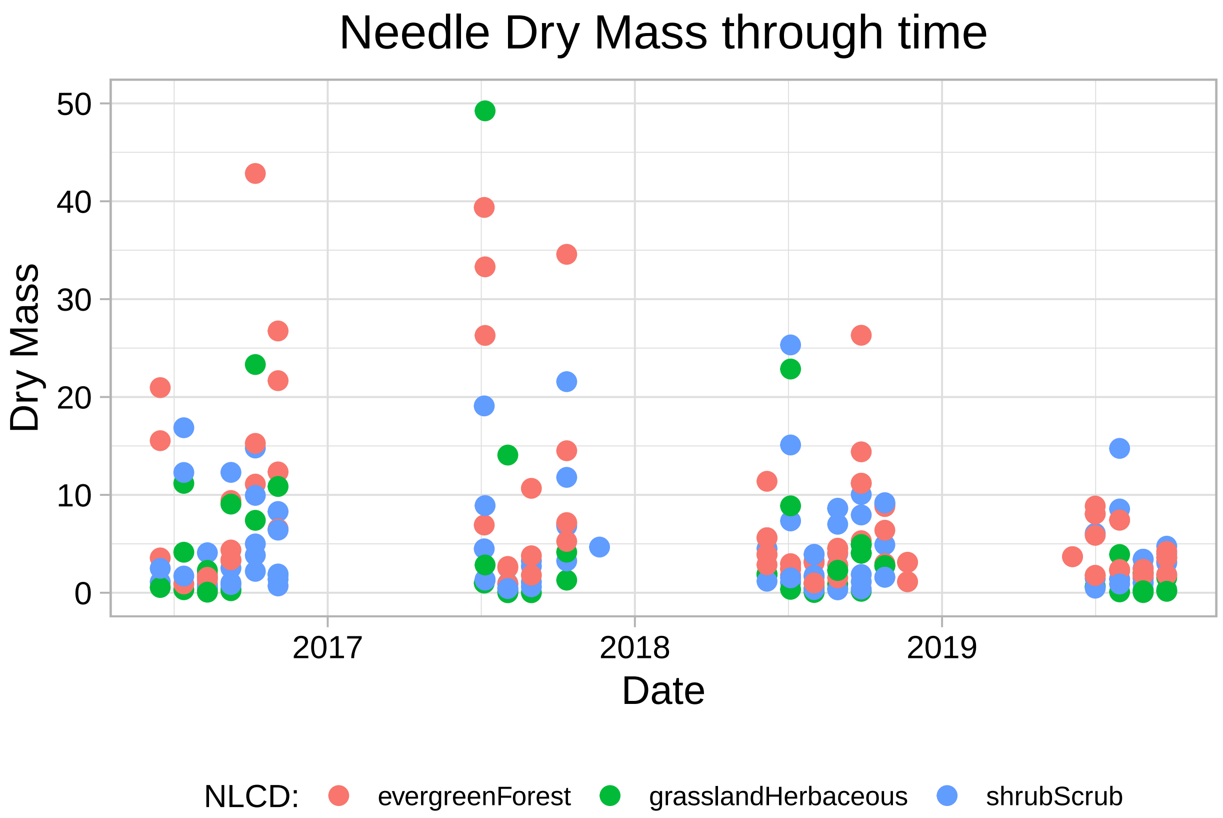
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature seems to be very similar between lakes and is higher on the summer months (Jun, July and August). TP seemes to have higher values in Peter Lake. Tp is very stable in Paul Lake throughout the year but in Peter lake we can see increase of TP over the summer. Fianlly TN seems to be fairly stable through time and similar in Paul and Peter 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)

```
plot2<- ggplot(subset(NiwotRidge, functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point(size = 3) +
   ylim(0,50) +
  labs(y = "Dry Mass", x= "Date", color = "NLCD:") +
  ggtitle("Needle Dry Mass through time")+
  mytheme

print(plot2)
```

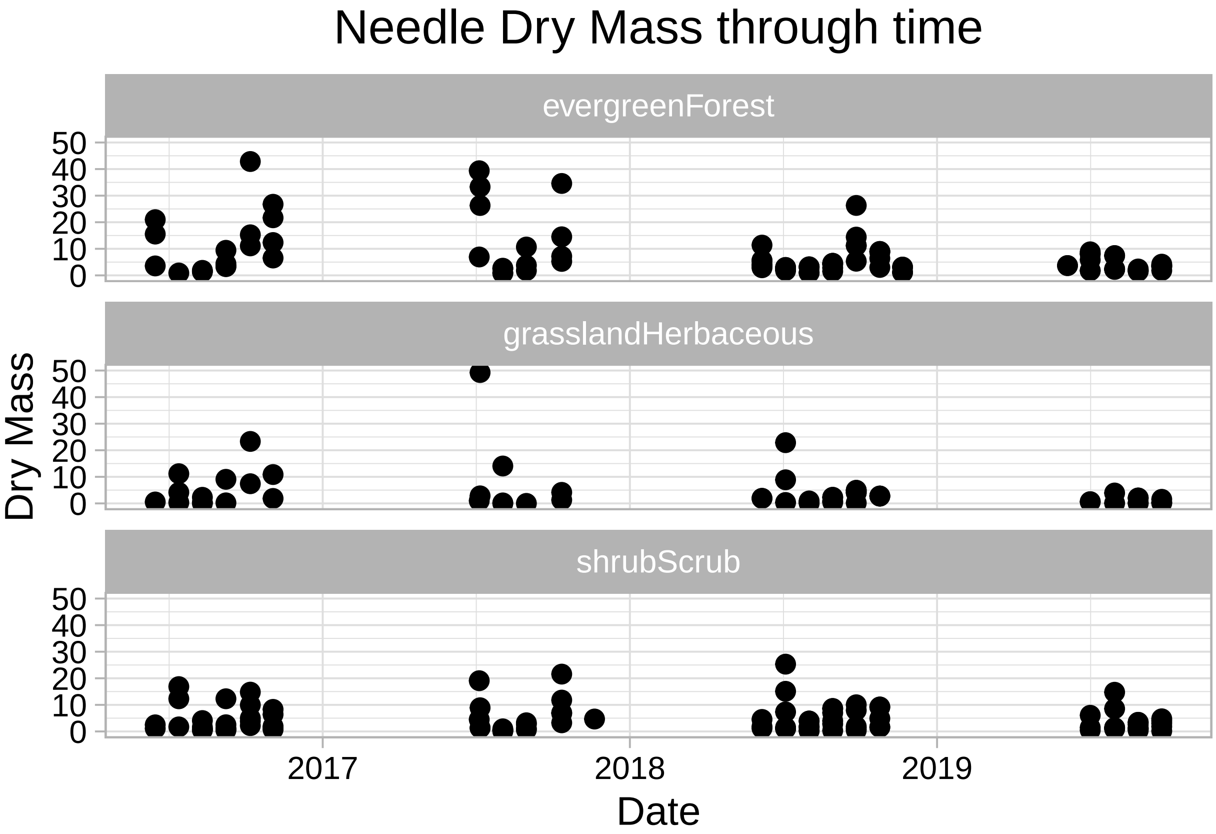
```
## Warning: Removed 1 rows containing missing values (geom_point).
```



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

```
plot3<- ggplot(subset(NiwotRidge, functionalGroup == "Needles"),
  aes(x= collectDate, y= dryMass))+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  geom_point(size = 3)+
  ylim(0,50) +
  labs(y = "Dry Mass", x= "Date") +
  ggtitle("Needle Dry Mass through time")+
  mytheme
print(plot3)
```

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



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

Answer: I think plot 7 is more effective because it is easier to distinguish the different needle dry mass patters for each NLCD class.