Assignment 5: Data Visualization

Aditi Jackson

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
# installing Packages
# install.packages("tidyverse") # helps with data manipulation and visualization
# install.packages("lubridate") # helps with manipulating date objects
# install.packages("here") # helps with controlling relative path
# install.packages("cowplot") # helps with data visualization
# install.packages("ggplot2") # helps with data visualization
# install.packages("ggthemes") # helps with themes

# loading Packages
library(tidyverse)
library(lubridate)
library(here)
library(cowplot)
library(ggplot2)
library(ggthemes)


# verifying working directory is "ENV872 Setup (Local)"
getwd()
```

```
## [1] "/Users/aditijackson/ENV872 Setup (local)"
```

```
# reading in data sets
PeterPaul_Lakes <- read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Process
NiwotRidge_Litter <- read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),string

#2

# converting sampledate column in Peter Paul data to date format
class(PeterPaul_Lakes$sampledate) # column imported as factor
```

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

```
PeterPaul_Lakes$sampledate <- ymd(PeterPaul_Lakes$sampledate) # changing to date

# converting collectDate column in Niwot Ridge data to date format
class(NiwotRidge_Litter$collectDate) # column imported as factor
```

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

```
NiwotRidge_Litter$collectDate <- ymd(NiwotRidge_Litter$collectDate) # changing to date
```

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
# creating a theme
my_theme <- theme_base()  +
 theme(
  axis.text = element_text(color = "black"),# setting axis text color
  plot.title = element_text(color = "black"),# setting title text color
  panel.background = element_rect(fill="lightyellow",color="blue"),
  legend.background = element_rect(color='grey',fill = 'white'),# setting legend color
  legend.position = 'bottom', # setting position of legend to bottom
  legend.justification = "center") # centering legend

# setting theme as default
theme_set(my_theme)
```

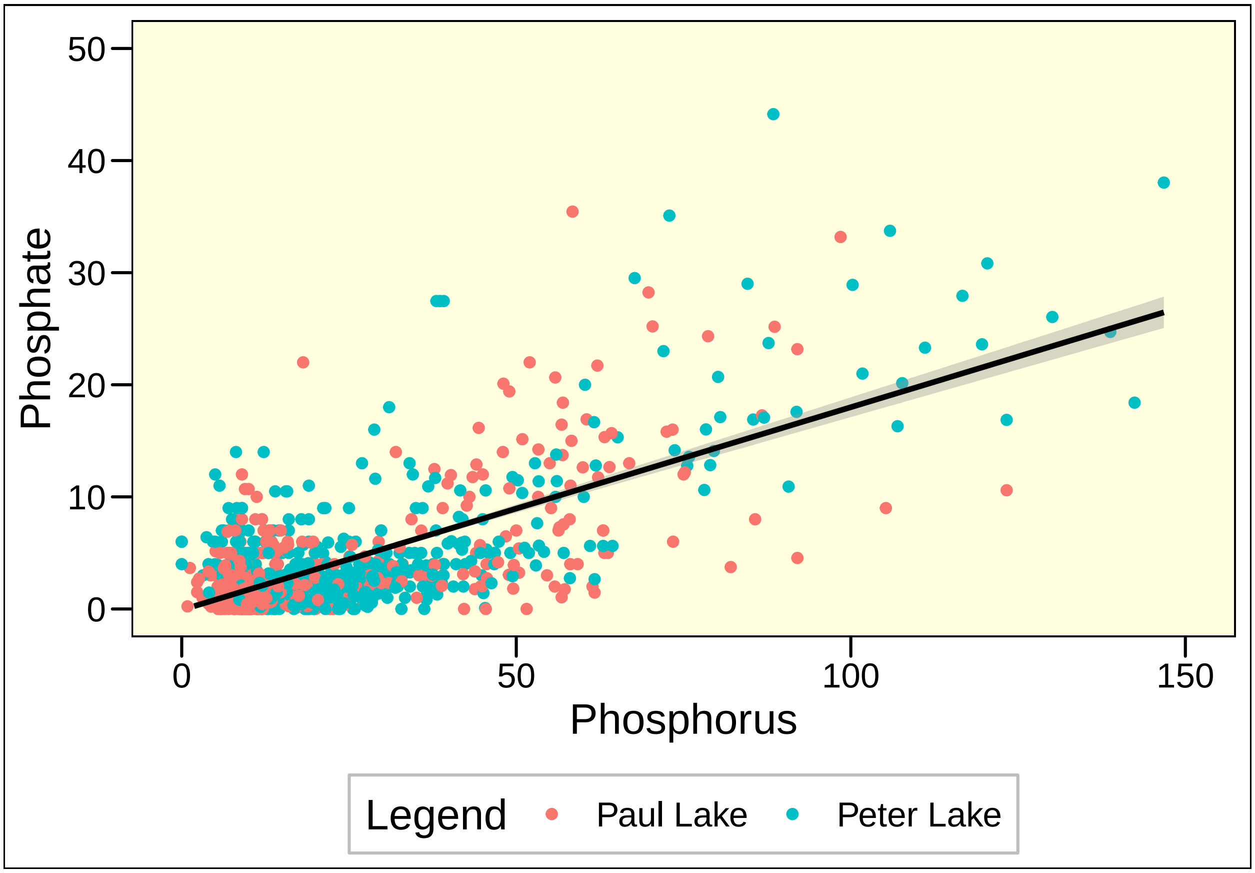
Create graphs

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()`).

```
#4
# plotting total phosphorus by phosphate

# setting separate aesthetics for tp_ug and po4, coloring points by lakename
Phos_Plot <- ggplot(PeterPaul_Lakes,aes(x = tp_ug, y = po4, color = lakename)) +
  # ploting points
  geom_point() +
  # adding line of best fit and coloring black
  geom_smooth(method = 'lm', color = "black") +
  # setting x-limit
  xlim(0,150) +
  # setting y-limit
  ylim(0,50) +
  # relabeling axes and renaming legend
  labs(x="Phosphorus",y="Phosphate",color="Legend")

# printing Phos_Plot
print(Phos_Plot)
```

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



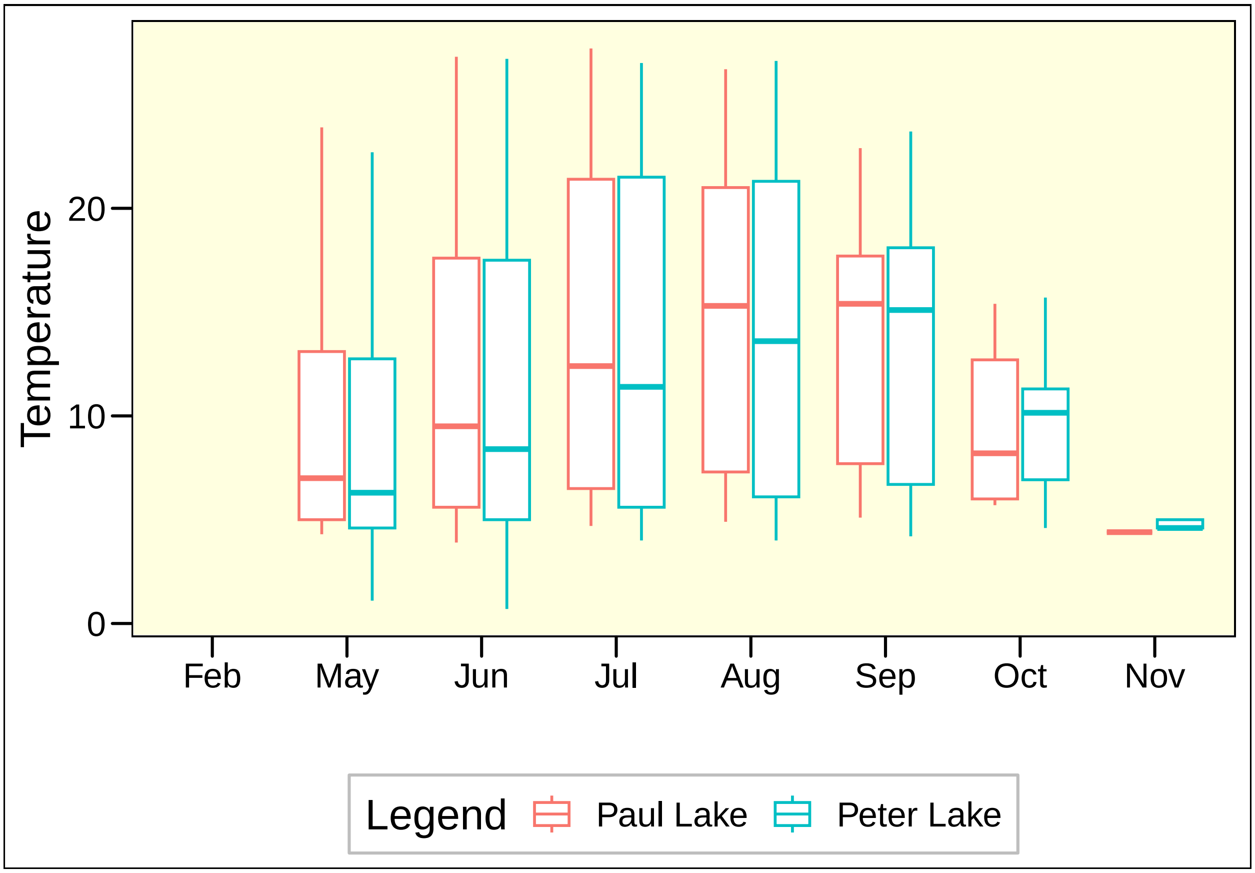
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

# (a)
# plotting temperature
Temp_BoxPlot <- ggplot(PeterPaul_Lakes,aes(
  # setting separate aesthetics, changing month numbers to month abbreviations
  x = factor(PeterPaul_Lakes$month,levels=1:12,labels=month.abb),
  y = temperature_C, color = lakename)) + # coloring points by lakename
  geom_boxplot() + # plotting as box plot
  labs(x=" ", y="Temperature",color="Legend") # labeling axes and renaming legend

# printing Temp_BoxPlot
print(Temp_BoxPlot)
```

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

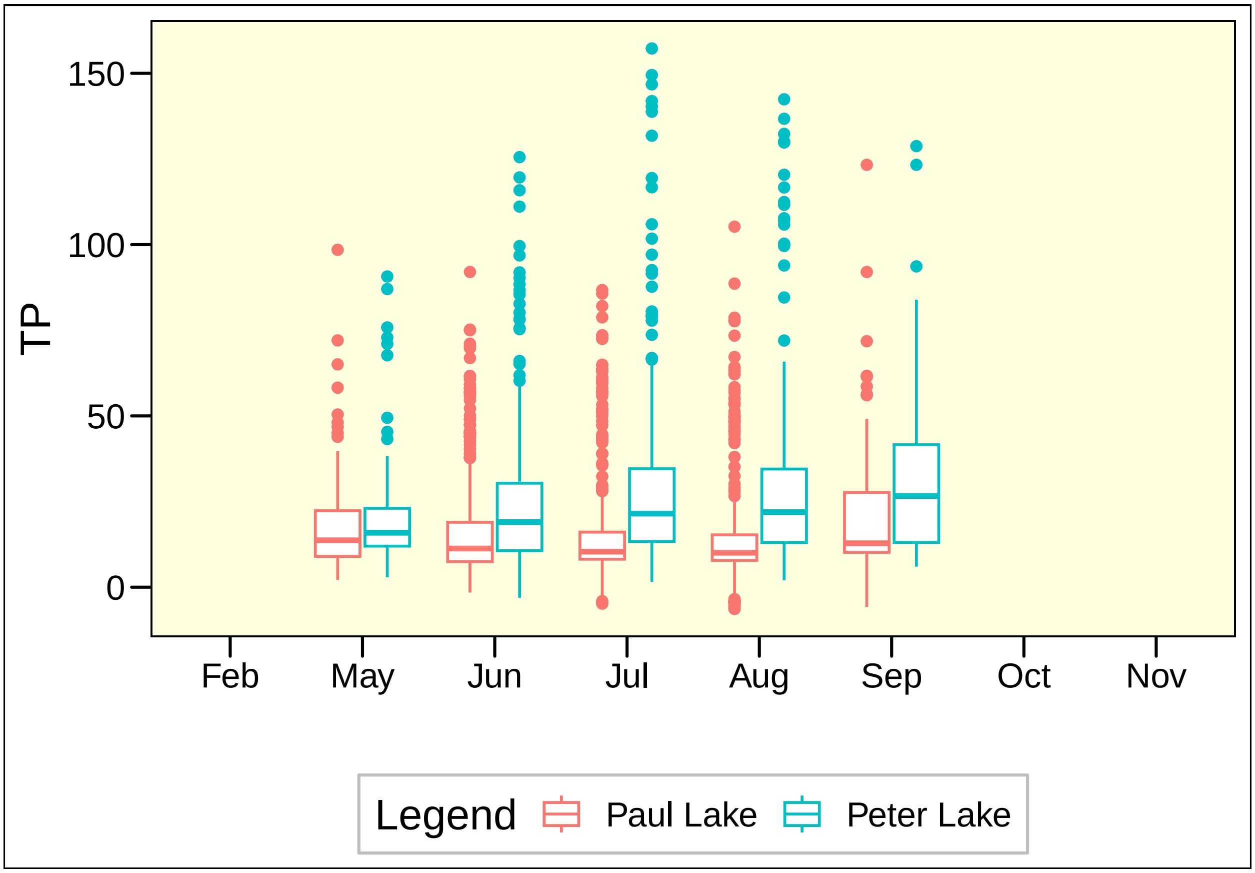
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



```r
# (b)
# plotting TP
TP_BoxPlot <- ggplot(PeterPaul_Lakes,aes(
  # setting separate aesthetics, changing month numbers to month abbreviations
  x = factor(PeterPaul_Lakes$month,levels=1:12,labels=month.abb),
  y = tp_ug, color = lakename)) + # coloring points by lakename
  geom_boxplot()+ # plotting as box plot
  labs(x= " ", y="TP",color="Legend") # labeling axes and renaming legend

# printing TP_BoxPlot
print(TP_BoxPlot)
```

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

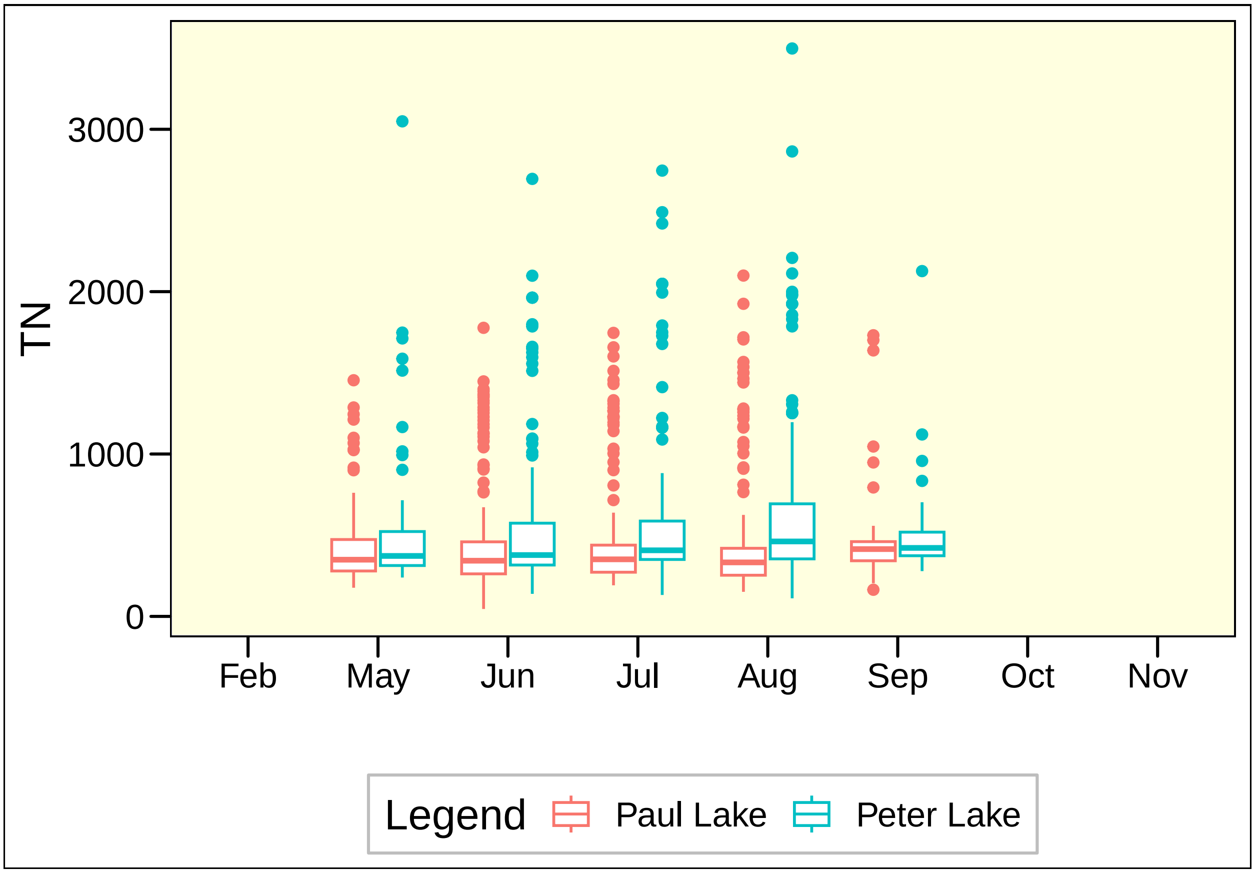
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```



```
# (c)
# plotting temperature
TN_BoxPlot <- ggplot(PeterPaul_Lakes,aes(
  # setting separate aesthetics, changing month numbers to month abbreviations,
  x = factor(PeterPaul_Lakes$month,levels=1:12,labels=month.abb),
  y = tn_ug, color = lakename)) + # coloring points by lakename
  geom_boxplot()+ # plotting as box plot
  labs(x=" ", y="TN",color="Legend") # labeling axes and renaming legend

# printing TN_BoxPlot
print(TN_BoxPlot)
```

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

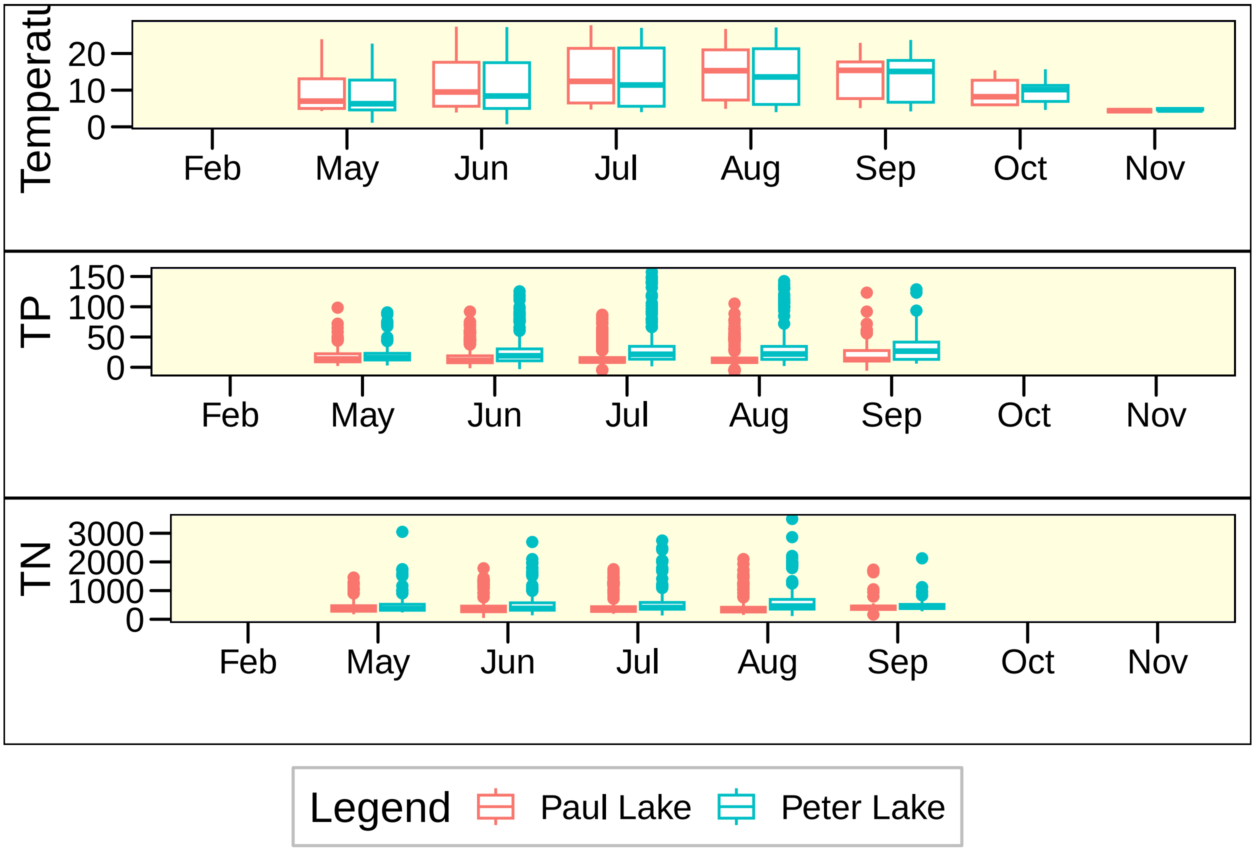
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```



```
# Combining all three graphs, removing legends
CombinedPlot <- plot_grid(Temp_BoxPlot + theme(legend.position="none"),
          TP_BoxPlot + theme(legend.position = "none"),
          TN_BoxPlot + theme(legend.position = "none"),
          get_legend(Temp_BoxPlot),
          ncol = 1,
          align = 'vh',
          rel_heights = c(1,1,1,.5)
          )
```

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

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

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

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

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

## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
## Warning: Use of `PeterPaul_Lakes$month` is discouraged.
## i Use `month` instead.

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

## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
## Placing graphs unaligned.

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
print(CombinedPlot)
```



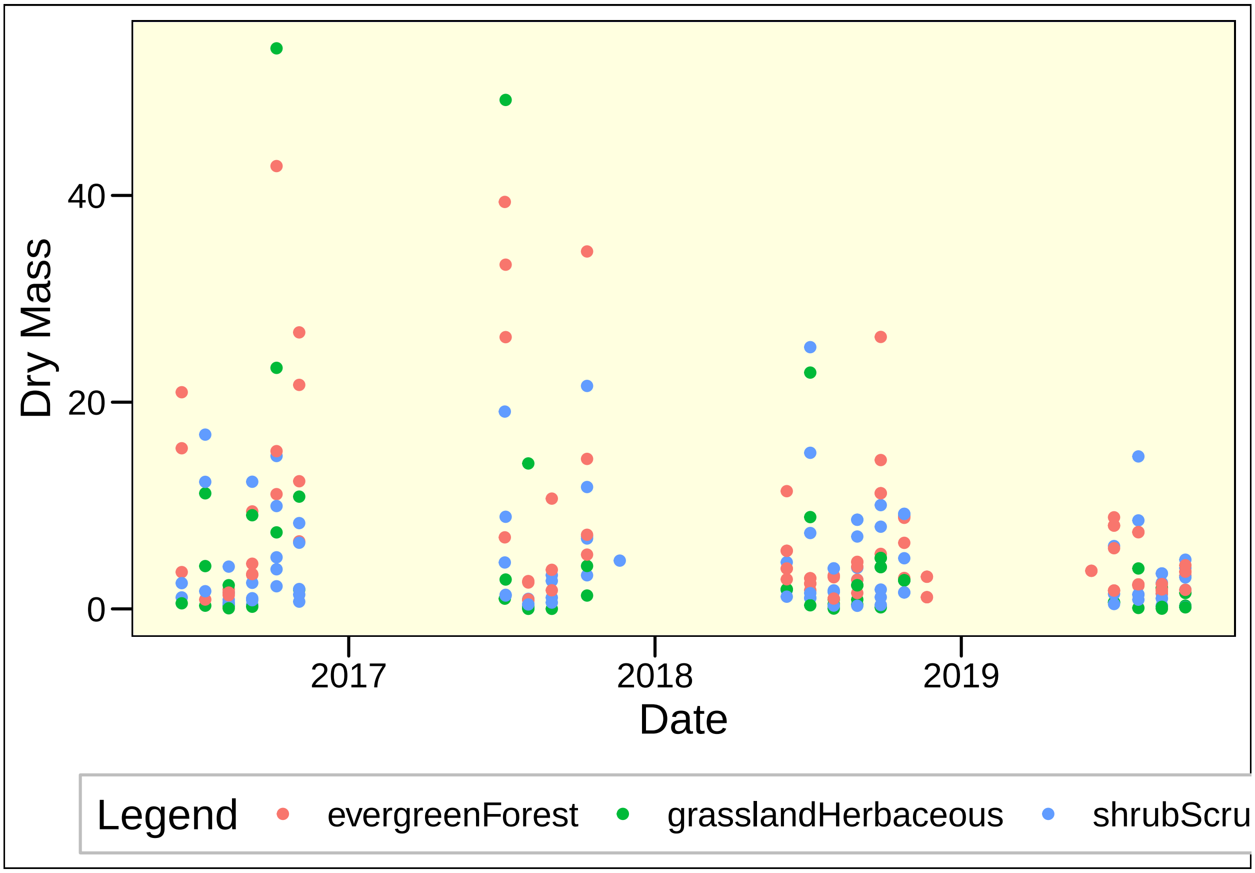
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The data is concentrated between May and September, spanning end of Spring to early Fall. The variables of interest appear to exhibit similar patterns between lakes over these seasons. For instance, mean temperature (C) appears to be very almost the same between the two lakes with similar IQRs. From the graph it appears that the TP and TN variables between the two lakes also exhibit means and spreads.

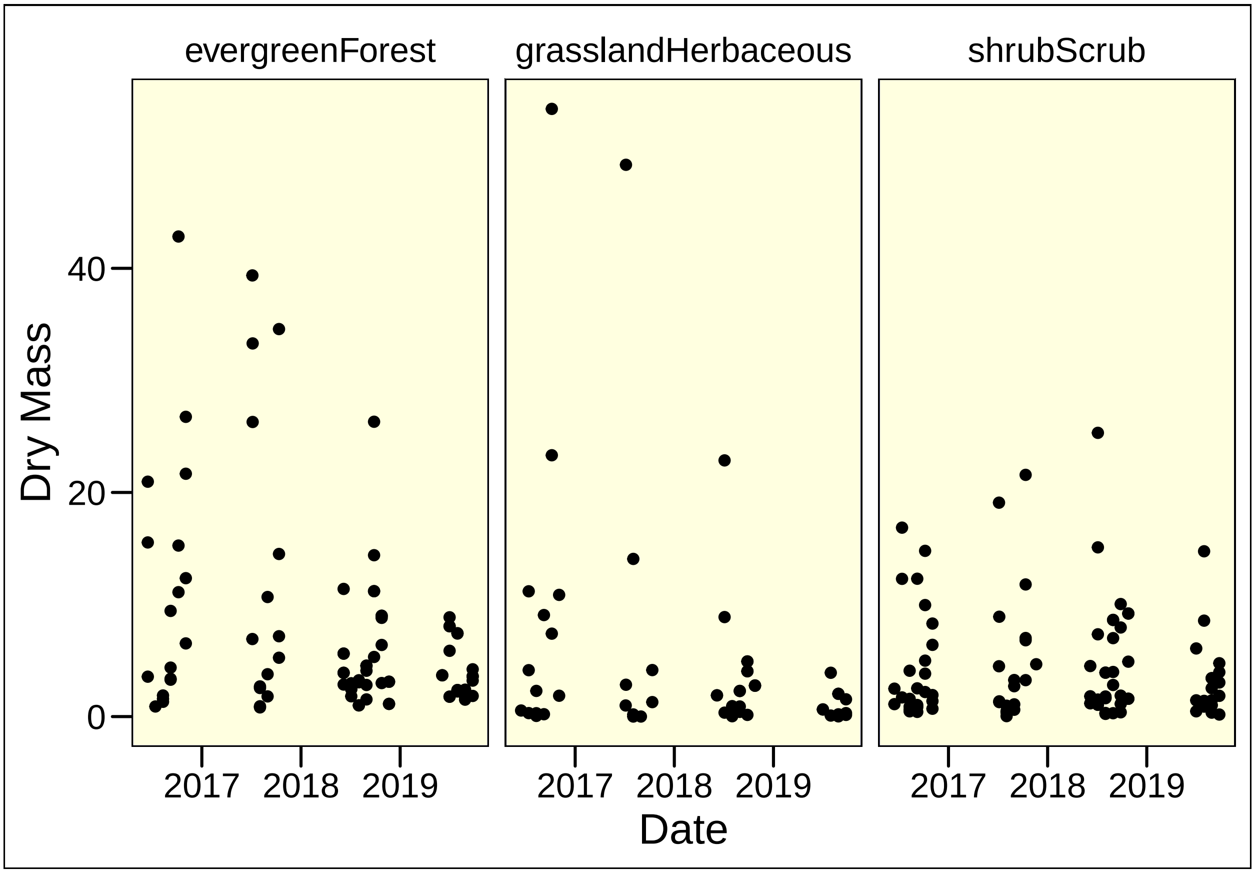
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6

# filtering data set for just functional group == Needles, defining new data frame
NiwotRidge_Needles <-
  NiwotRidge_Litter %>%
    filter(functionalGroup=="Needles")

# using new Needles data frame to create plot
Needles_Plot_Q6 <- ggplot(NiwotRidge_Needles,
                          aes(x=collectDate, y=dryMass, color=nlcdClass))+
  geom_point()+
  labs(x="Date",y="Dry Mass",color="Legend") # relabeling axes
print(Needles_Plot_Q6) # printing new plot
```



```
#7
# plotting needles separated by needle class
Needles_Plot_Q7 <- ggplot(NiwotRidge_Needles,
                          aes(x=collectDate, y=dryMass))+
  geom_point()+
  # adding "facet_wrap" command to separate data by class
  facet_wrap(vars(nlcdClass))+
  labs(x="Date",y="Dry Mass") # relabeling axes
print(Needles_Plot_Q7)
```



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 see year-over-year changes for each needle class. In other words, I'm better able to discern a trend for a particular class and compare that to the trend of another class. With plot 6, the data blends together and it takes more effort to extract patterns in the data.