Assignment 5: Data Visualization

Natalie von Turkovich

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] "/Users/natalievonturkovich/Documents/DUKE/Courses/Spring 22/ENV_872_EDA/Environmental_Data_Analy
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(cowplot)
library(lubridate)
```

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

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

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

```
PeterPaul.chem.nutrients <-
  read.csv("/Users/natalievonturkovich/Documents/DUKE/Courses/Spring 22/ENV_872_EDA/Environmental_Data_
           stringsAsFactors = TRUE)
PeterPaul.chem.nutrients.gathered <-
  read.csv("/Users/natalievonturkovich/Documents/DUKE/Courses/Spring 22/ENV_872_EDA/Environmental_Data_
           stringsAsFactors = TRUE)
Niwot.litter.processed <-
  read.csv("/Users/natalievonturkovich/Documents/DUKE/Courses/Spring 22/ENV_872_EDA/Environmental_Data_
           stringsAsFactors = TRUE)

#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
PeterPaul.chem.nutrients.gathered$sampledate <- ymd(PeterPaul.chem.nutrients.gathered$sampledate)
Niwot.litter.processed$collectDate <- ymd(Niwot.litter.processed$collectDate)
```

Define your theme

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

```
#3
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right") #alternative: legend.position + legend.justification

theme_set(mytheme)
```

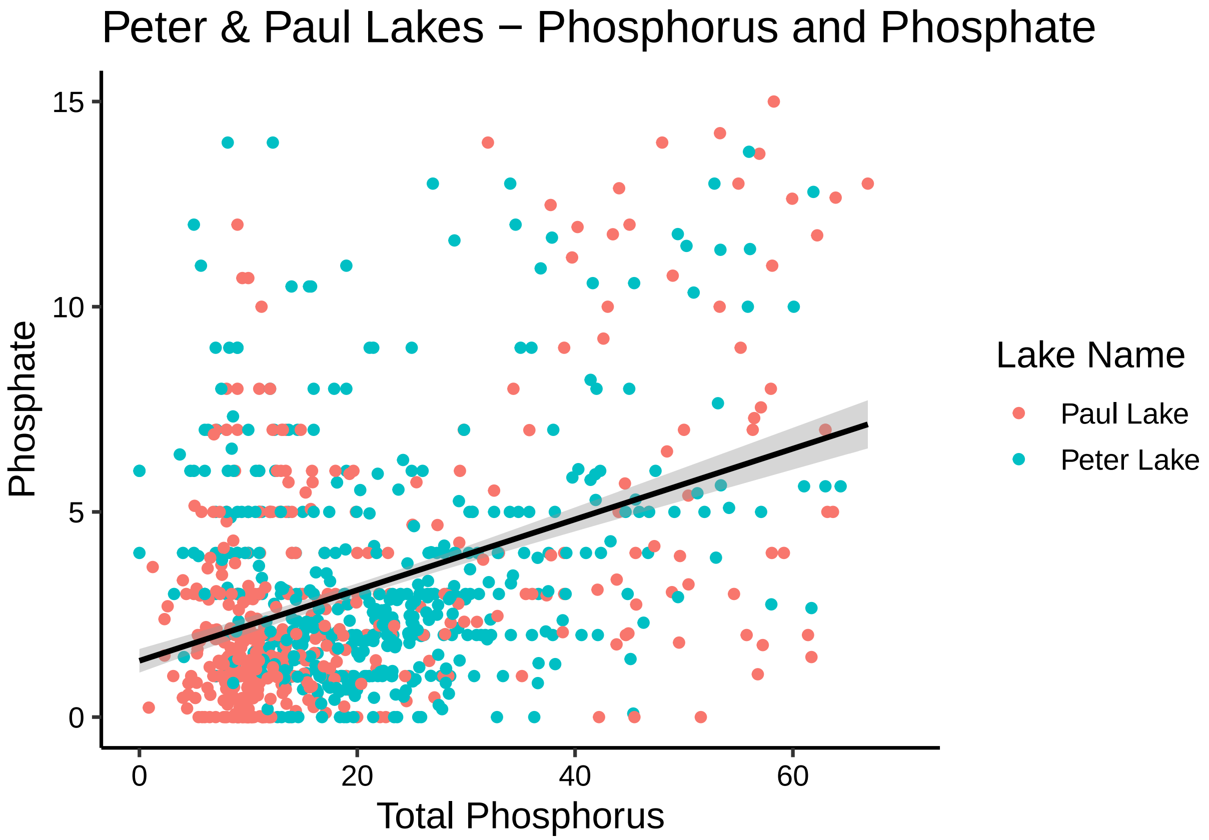
Create graphs

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
NTL_LITER_4 <-
  ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4, color = lakename))+
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  xlim(0, 70) +
  ylim(0, 15)+
  labs( x = "Total Phosphorus", y = "Phosphate", color = "Lake Name",
        title = "Peter & Paul Lakes - Phosphorus and Phosphate")
print(NTL_LITER_4)
```

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

## Warning: Removed 22020 rows containing non-finite values (stat_smooth).

## Warning: Removed 22020 rows containing missing values (geom_point).
```

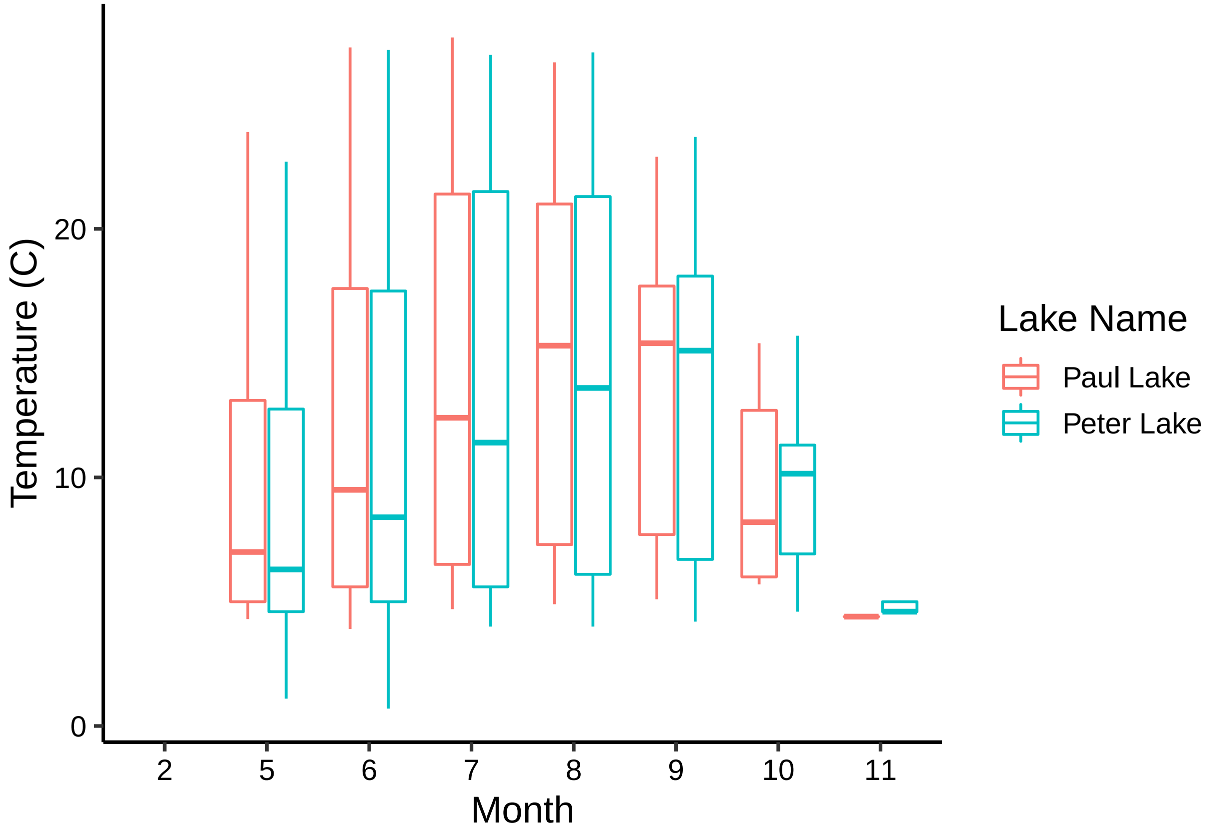


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
#5
PeterPaul.chem.nutrients$month<-as.factor(PeterPaul.chem.nutrients$month)
```

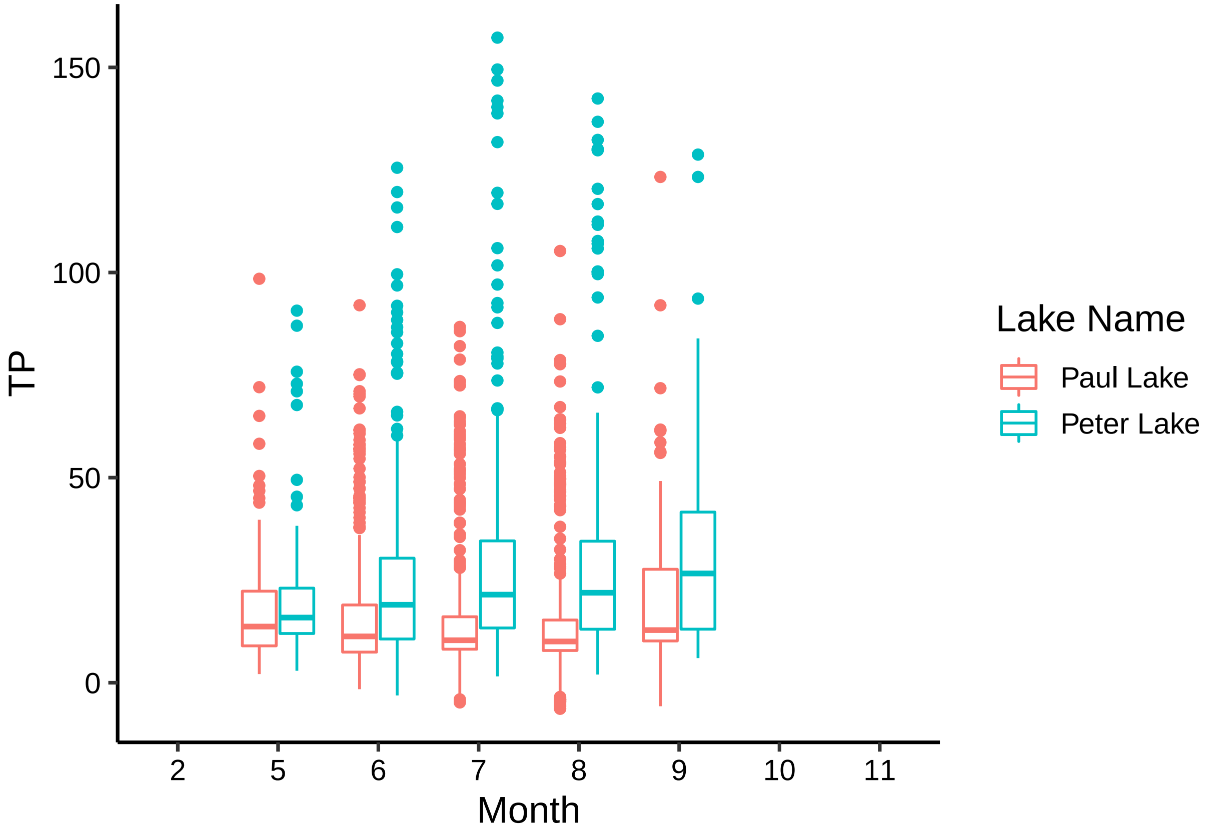
```
NTL_LITER_5_temp <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  labs(y = "Temperature (C)", x = "Month", color = "Lake Name")
print(NTL_LITER_5_temp)
```

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



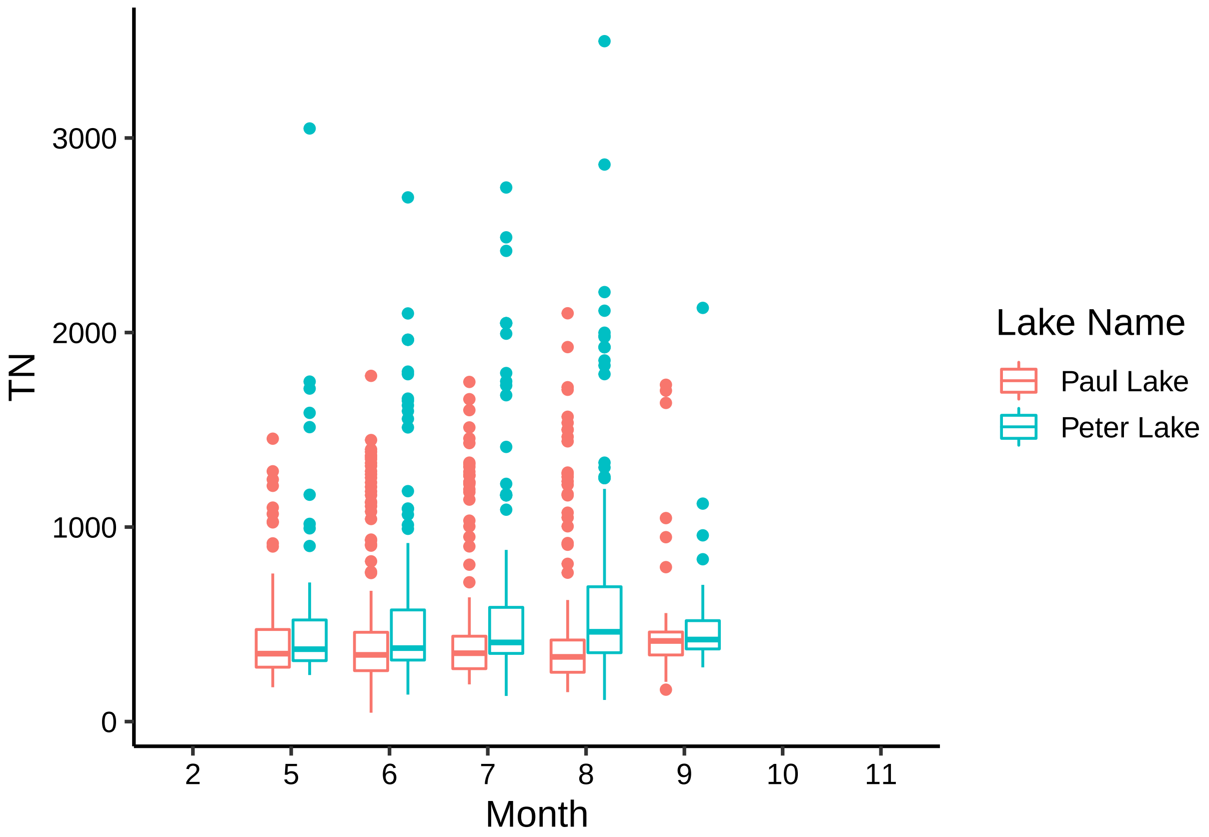
```
NTL_LITER_5_tp <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(y = "TP", x = "Month", color = "Lake Name")
print(NTL_LITER_5_tp)
```

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



```
NTL_LITER_5_tn <-
  ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(y = "TN", x = "Month", color = "Lake Name")
print(NTL_LITER_5_tn)
```

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



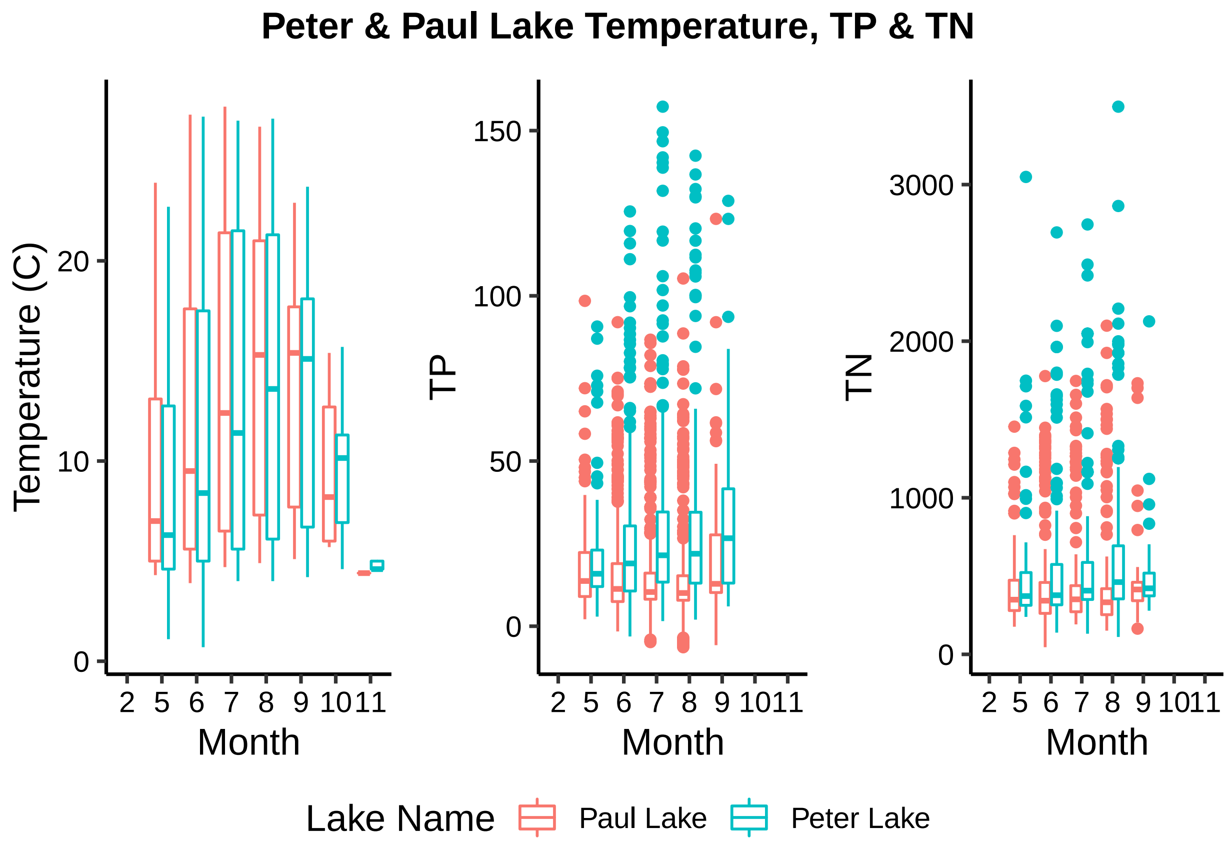
```
plot_5<-plot_grid(NTL_LITER_5_temp + theme(legend.position="none"),
                  NTL_LITER_5_tp + theme(legend.position="none"),
                  NTL_LITER_5_tn + theme(legend.position="none"),
                  nrow = 1, align = 'h', rel_heights = c(1, 1,1))
```

```
## 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).
```

```
legend_5 <- get_legend(
  NTL_LITER_5_temp +
    guides(color = guide_legend(nrow = 1)) +
    theme(legend.position = "bottom"))
```

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

```
title_5 <- ggdraw() + draw_label("Peter & Paul Lake Temperature, TP & TN", fontface='bold')

plot_grid(title_5, plot_5, legend_5, ncol = 1, rel_heights = c(.1, 1, .1))
```

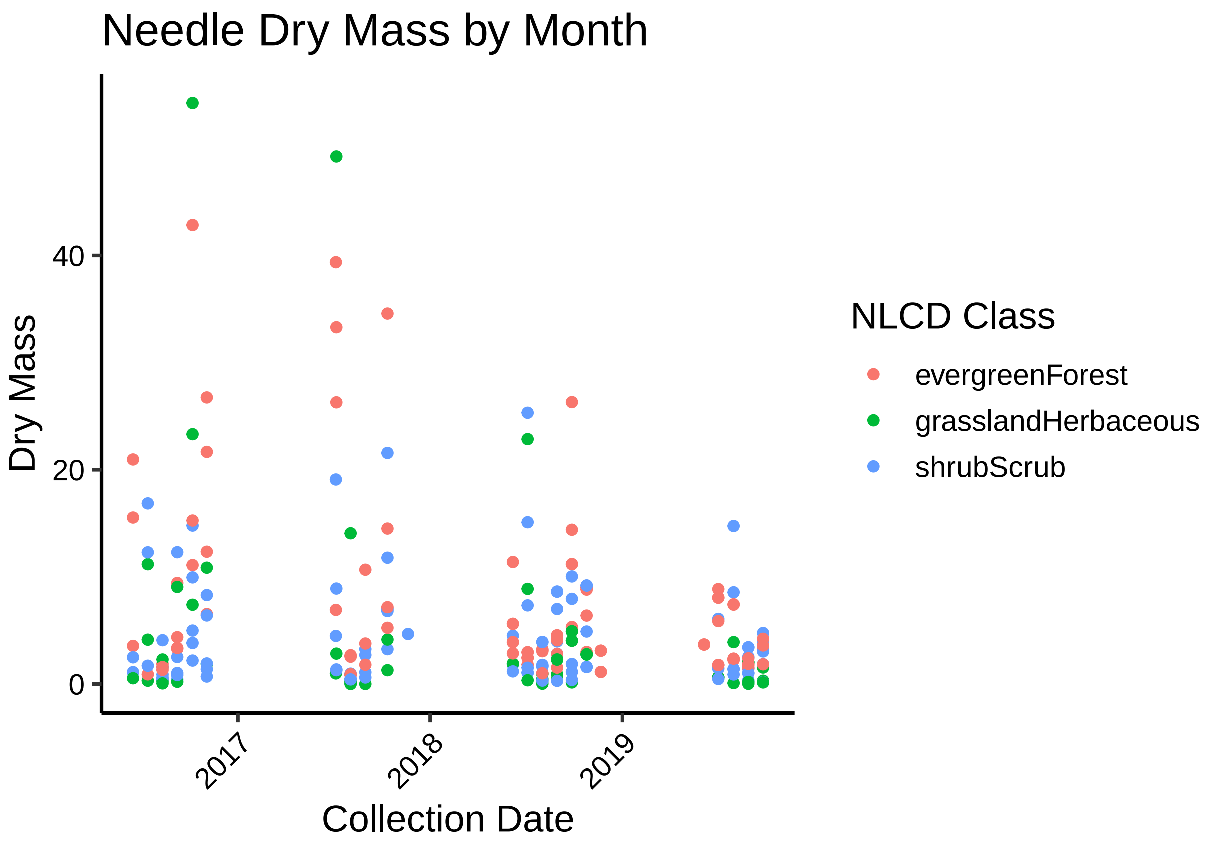


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Over seasons, temperature has the widest variabillity, where as TP and even more so TN have a smaller range of where most observations sit, with a larger number of outliers. Between lakes, Peter lake has a wider range of temperature readings for each month. For TP and TN, Peter lake also has a wider range, but in general Peter Lake values for these characteristics are higher than Paul Lake.

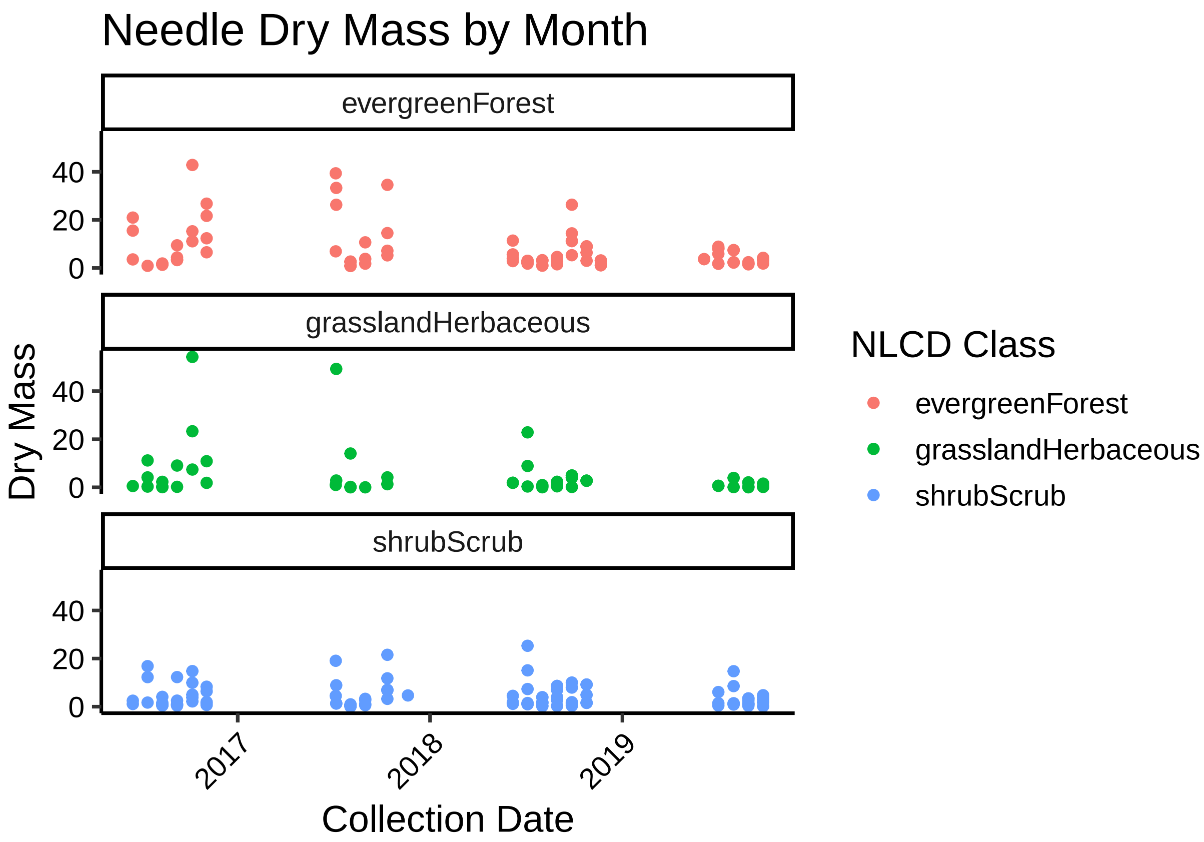
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

```
#6
Needles_plot <-
  ggplot(filter(Niwot.litter.processed, functionalGroup == "Needles") ,
         aes(x = collectDate, y = dryMass)) +
  geom_point(aes(color = nlcdClass)) +
  labs(y = "Dry Mass", x = "Collection Date", color = "NLCD Class",
       title = "Needle Dry Mass by Month") +
  #scale_x_date(date_breaks = "2 months", date_labels = "%b %y") +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))
print(Needles_plot)
```



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

```
#7
Needles_plot_facet <-
  ggplot(filter(Niwot.litter.processed, functionalGroup == "Needles") ,
         aes(x = collectDate, y = dryMass)) +
  geom_point(aes(color = nlcdClass)) +
  labs(y = "Dry Mass", x = "Collection Date", color = "NLCD Class",
       title = "Needle Dry Mass by Month") +
  #scale_x_date(date_breaks = "2 months", date_labels = "%b %y") +
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))+
  facet_wrap(vars(nlcdClass), nrow = 3)
print(Needles_plot_facet)
```



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

Answer: I ghink that plot 7 is more effective than plot 6. Plot 7 separates out each NLCD class, making it easier to see the distribiation of dry mass for each class. You are able to see what the distribution of each class looks like then compare them to eachother. With all of the classes in the same plot it is hard to see how they compare with all of the points overlapped.