Assignment 5: Data Visualization

Julia Weinberg

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/juliaweinberg/Desktop/github repos/Environmental_Data_Analytics_2022/Assignments"
```

```
#install(tidyverse)
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

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

```
#install(cowplot)
library(cowplot)

NTL.LTER <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
```

```
#import NTL.LTER
Niwot <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")
#import Niwot

#2
NTL.LTER$sampledate <- as.Date(NTL.LTER$sampledate)
#change NTL.LTER to date format


Niwot$collectDate <- as.Date(Niwot$collectDate)
#change Niwot to date format
```

Define your theme

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

```
#3
defined.theme <- theme_gray(base_size = 14) +
  theme(axis.text = element_text(color = "Darkblue"),
        legend.position = "bottom")
#define theme
```

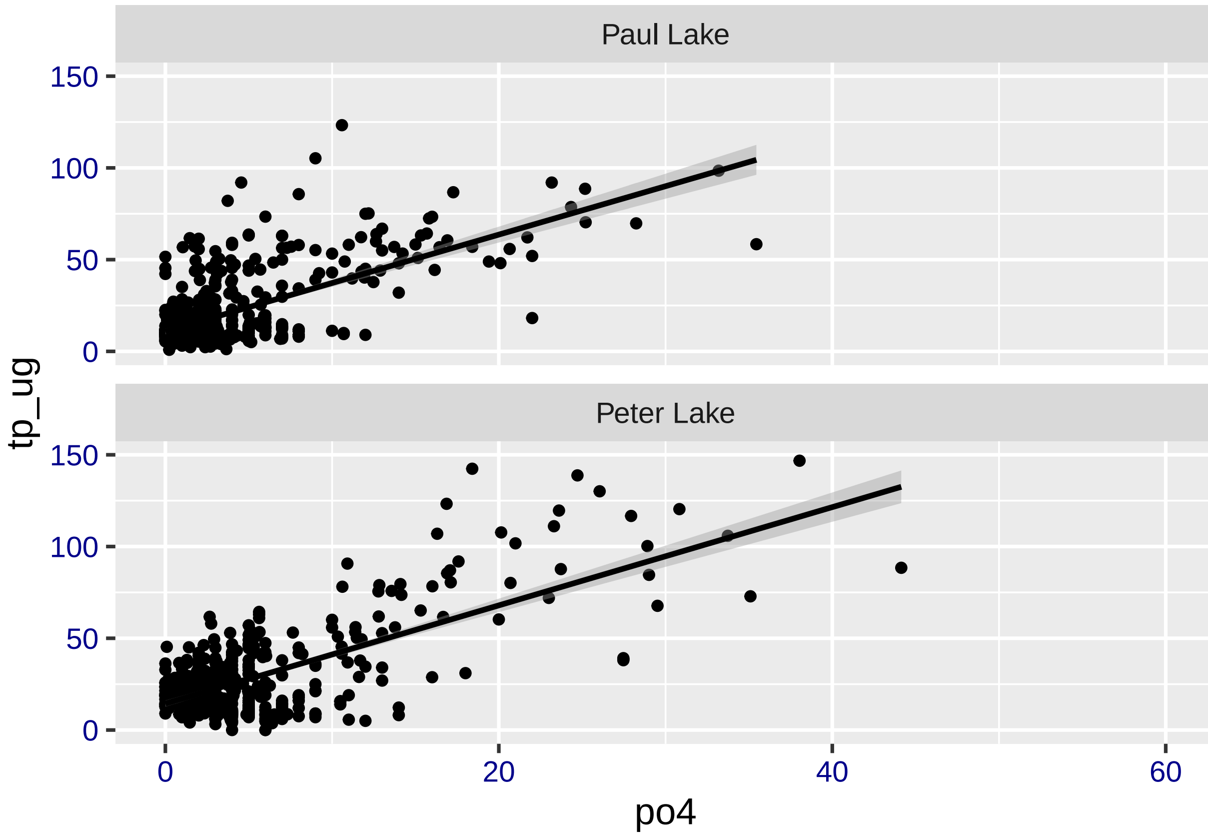
Create graphs

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
P.by.Po <-ggplot(NTL.LTER, aes(x = po4, y = tp_ug)) + #plot tp by po
  geom_point() +
  facet_wrap(vars(lakename), nrow = 2)+ #plot by lakename
  xlim(0, 60)+ #limit data plotted
  ylim(0, 150)+ #limit data plotted
  defined.theme+ #include theme

  geom_smooth(method = "lm", color = "black") #add best fit line

print(P.by.Po)
```

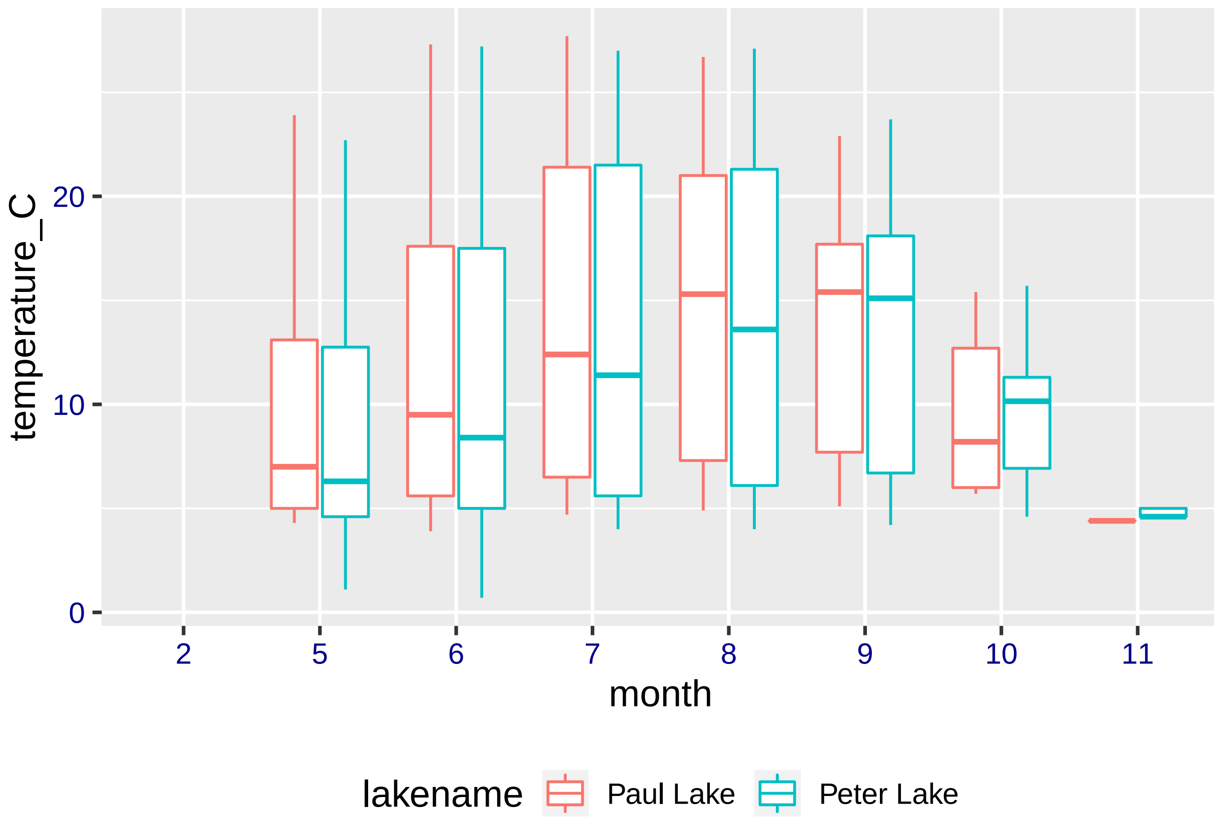
```
## `geom_smooth()` using formula 'y ~ x'

## 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.

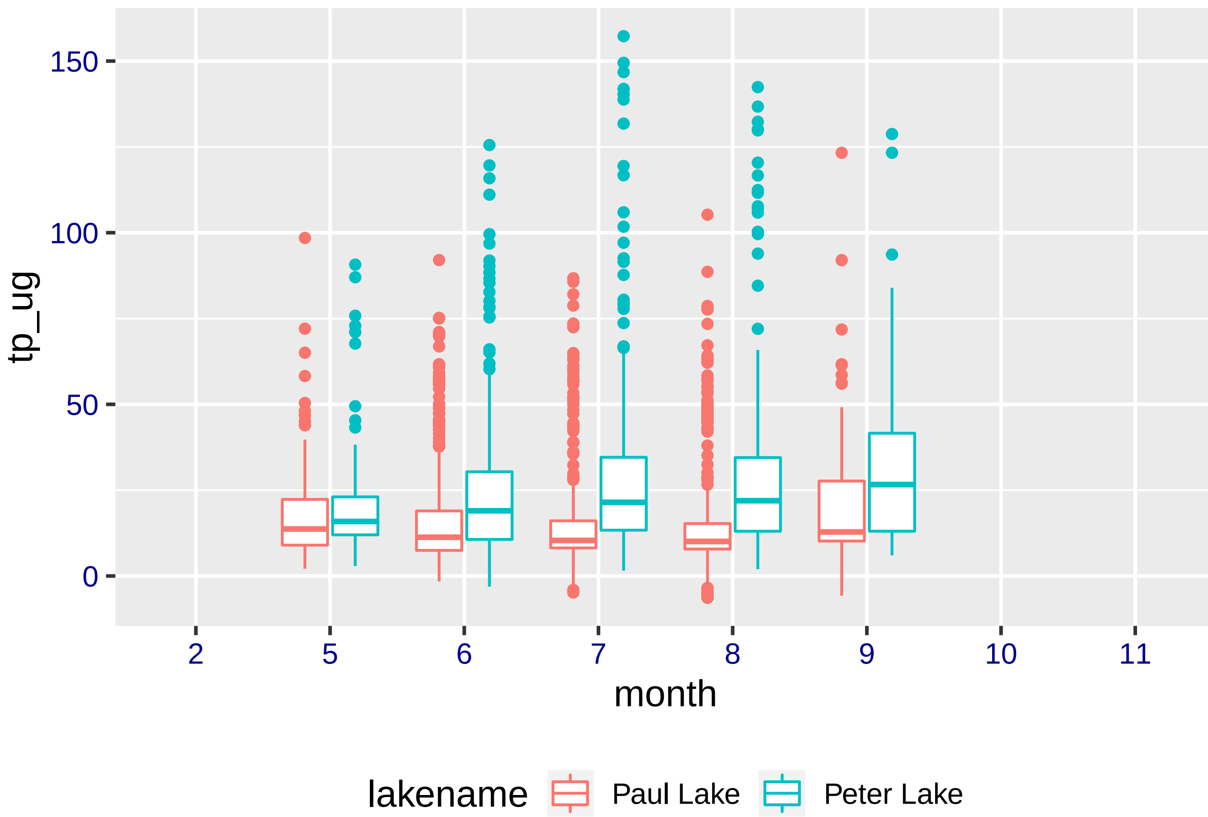
```
#5
NTL.LTER$month <- as.factor(NTL.LTER$month) #change date to factor

temp_plot <- ggplot(NTL.LTER, aes(x = month, y = temperature_C)) +
  #plot temp by month
  defined.theme+ #include theme
  geom_boxplot(aes(color = lakename)) #color by lake name
print(temp_plot)
```

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



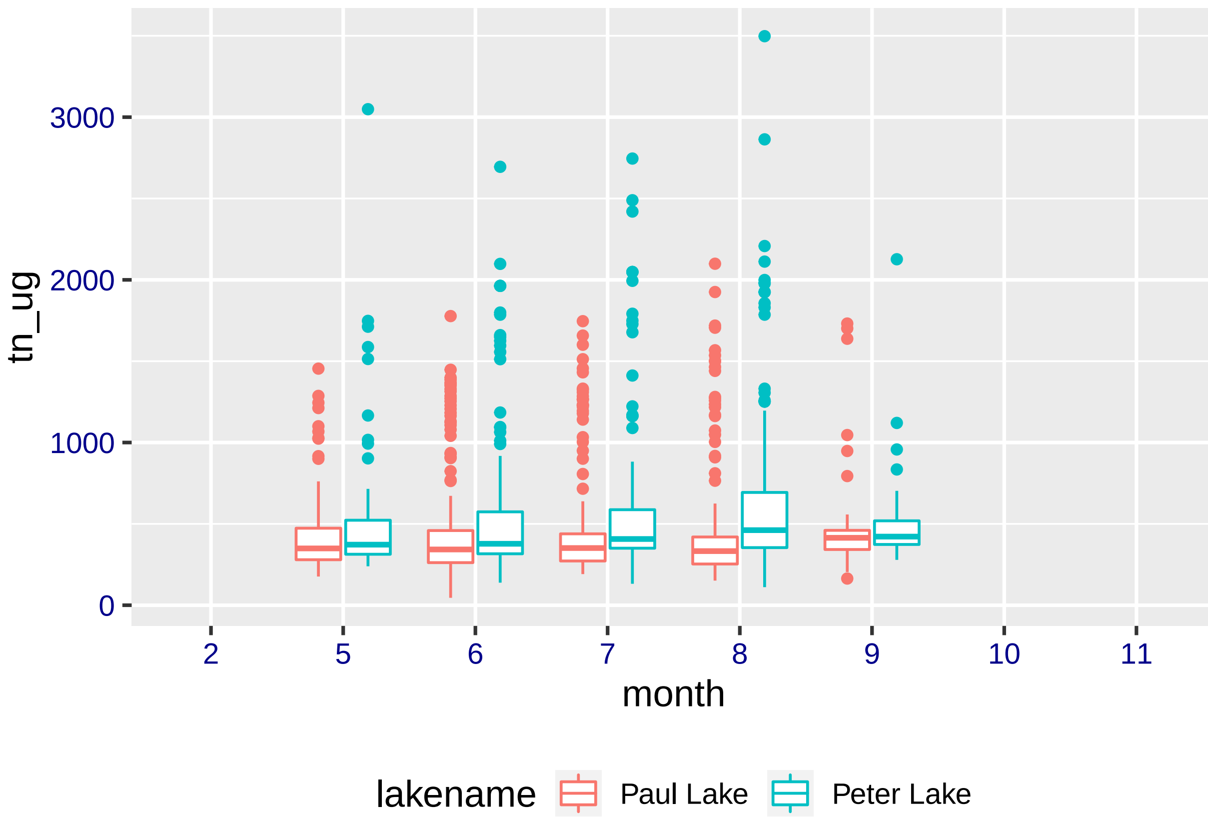

```
tp_plot <- ggplot(NTL.LTER, aes(x = month, y = tp_ug)) +
  #plot tp by month
  defined.theme+ #include theme
  geom_boxplot(aes(color = lakename)) #color by lakename
print(tp_plot)
```

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




```
tn_plot <- ggplot(NTL.LTER, aes(x = month, y = tn_ug)) +
  #plot tn by month
  defined.theme+ #include theme
  geom_boxplot(aes(color = lakename)) #color by lakename
print(tn_plot)
```

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



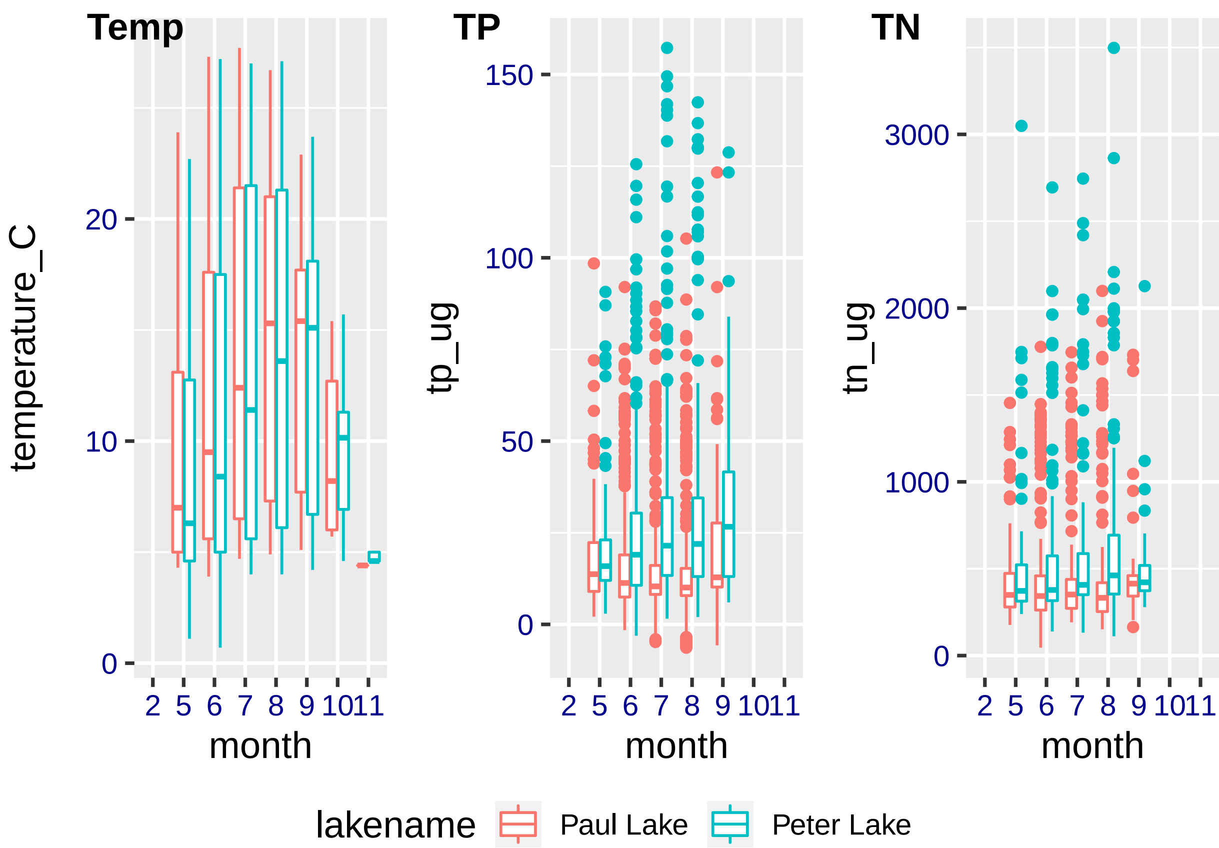

```
nutrient.cowplot <- plot_grid( #create cowplot of data
  temp_plot + theme(legend.position="none"), #remove temp legend
  tp_plot + theme(legend.position="none"), #remove tp legend
  tn_plot + theme(legend.position="none"), #remove tn legend
  align = 'vh', #align plots vertically and horizontally
  labels = c("Temp", "TP", "TN"), #label each graph
  hjust = -1,
  nrow = 1) #include one row graphs
```

```
## 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 <- get_legend(temp_plot + #create legend
  guides(color = guide_legend(nrow = 1)) +
  theme(legend.position = "bottom"))
```

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

```
nutrient.cowplot.w.legend <- plot_grid(nutrient.cowplot,
  legend,ncol=1,rel_heights = c(1, .1)) #add legend to cowplot

print(nutrient.cowplot.w.legend)
```



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

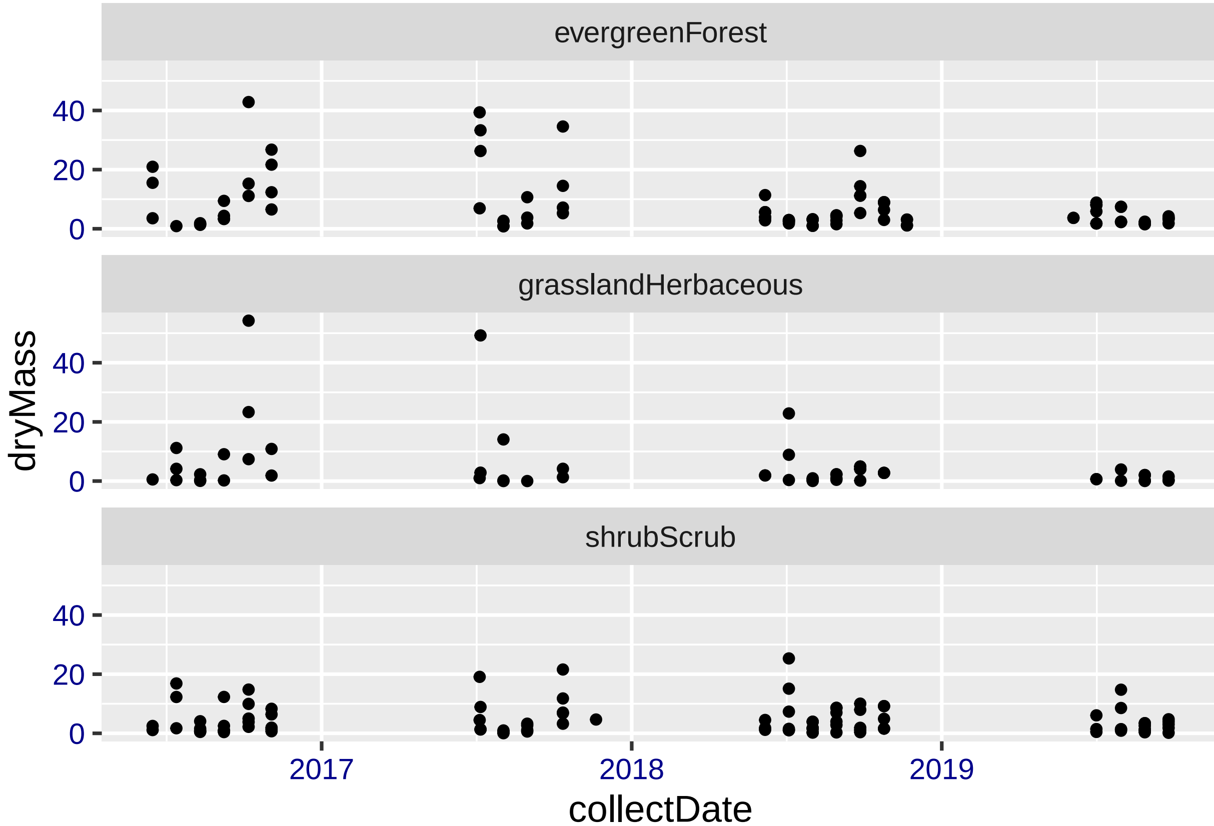
> Answer: Both Peter and Paul lake experience highest temperature values in months 7 and 8, which makes sense as those are summer months. Peter lake expereinces much higher levels of phosphorus and phosphate than Paul lake, with summer and fall readings being the highest. This could indicate more agricultural runoff entering Peter Lake 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)

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

```
#6
Needles.color <-
  ggplot(subset(Niwot, functionalGroup == "Needles"), #subset needles
       aes(x = collectDate, y = dryMass, color = nlcdClass))+
  #plot mass by collect date, colored by nlcd class
  defined.theme + #include theme
 geom_point()
print(Needles.color)
```


```
#7
Needles.facet <-
  ggplot(subset(Niwot, functionalGroup == "Needles"), #subset needles
       aes(x = collectDate, y = dryMass)) + #plot mass by collect date
  defined.theme + #include theme
 geom_point() +
   facet_wrap(vars(nlcdClass), nrow = 3) #create different plots for nlcd class
print(Needles.facet)
```



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

> Answer: The second plot (using facet_wrap) is more effective because it is easier to distinguish between the three different nlcd classes and observe trends. The first plot, using color to differentiate, is very busy and difficult to analyze as the three nlcd classes all blend together in that plot.