Assignment 5: Data Visualization

Tasha Griffiths

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 set up session
getwd()
```

```
## [1] "C:/Users/Tasha Griffiths/Documents/Duke Year 1/Spring 22 Classes/Environmental Data Analytics/G
```

```
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   1.1.4     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()
```

```
library(cowplot)

PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
           stringsAsFactors = TRUE)
Niwot.Ridge.litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                               stringsAsFactors = TRUE)


#2 update so that dates are being read as dates properly
class(Niwot.Ridge.litter$collectDate)
```

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

```
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
Niwot.Ridge.litter$collectDate <- as.Date(Niwot.Ridge.litter$collectDate, format = "%Y-%m-%d")
PeterPaul.chem.nutrients$sampledate <- as.Date(PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
```

Define your theme

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

```
#3

mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")

#future code for if I wanted to add different colors to the theme
#scale_color_manual(values=c("olivedrab4", "steelblue3"))

theme_set(mytheme) #set theme for all subsequent plots
```

Create graphs

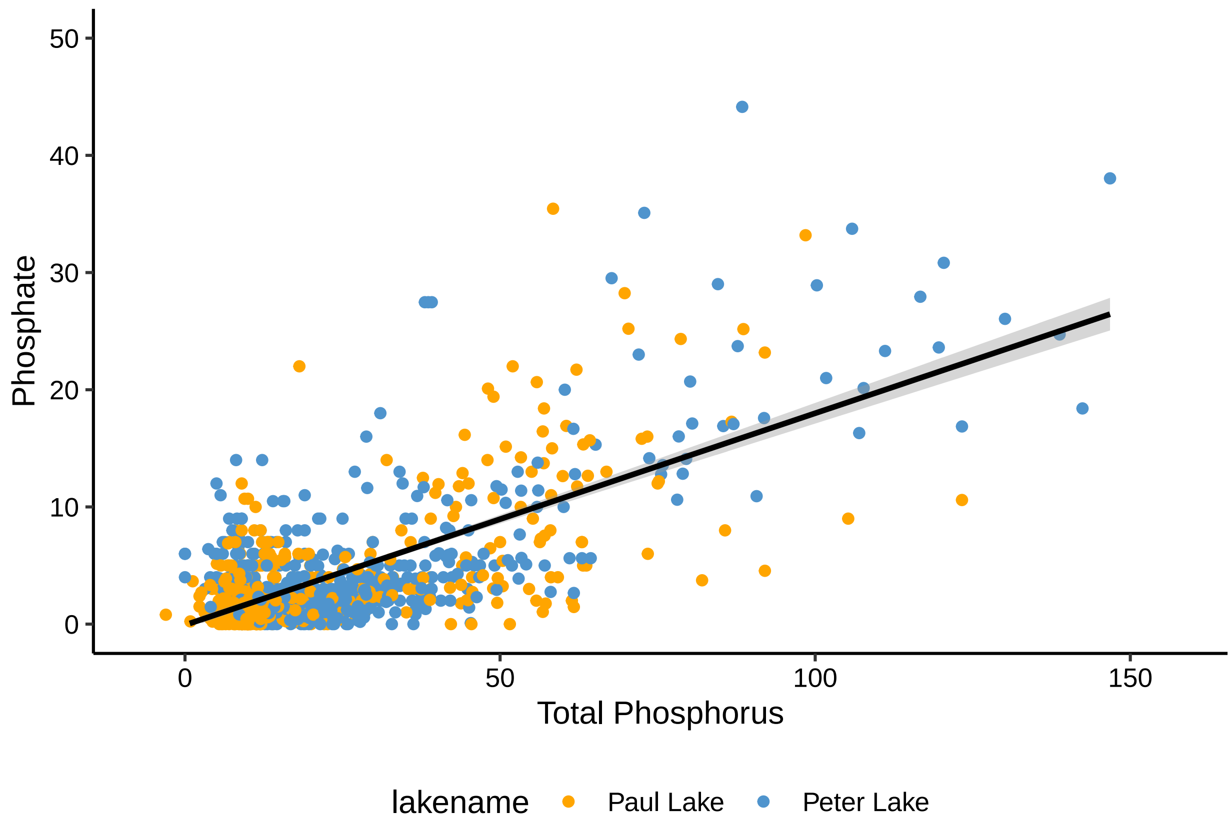
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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

library(ggplot2)
```

```r
NTL.LTER.plot <- ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4,
                                                        color = lakename)) +
  geom_point() + ylim(0, 50) + #add points and limit outliers
  scale_color_manual(values=c("orange", "steelblue3")) + #separate colors for each lake
  xlab("Total Phosphorus") + ylab("Phosphate") + #change axis labels to be more clear
  geom_smooth(colour="black", method = lm) #adding linear best fit line
print(NTL.LTER.plot)
```

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

## Warning: Removed 21947 rows containing non-finite values (stat_smooth).

## Warning: Removed 21947 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_smooth).
```



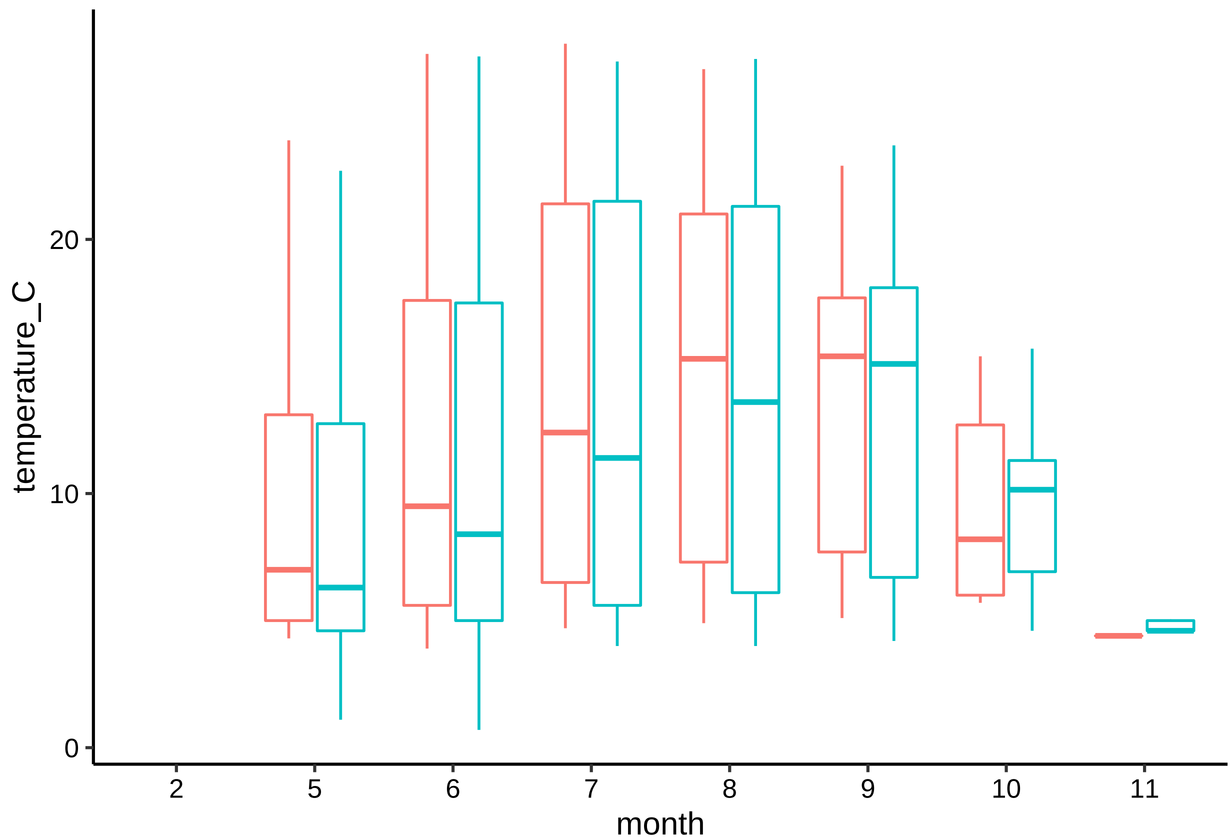
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```r
#5
#used to fix month error with boxplots. Needed to be a factor
class(PeterPaul.chem.nutrients$month)
```

```
## [1] "integer"
```

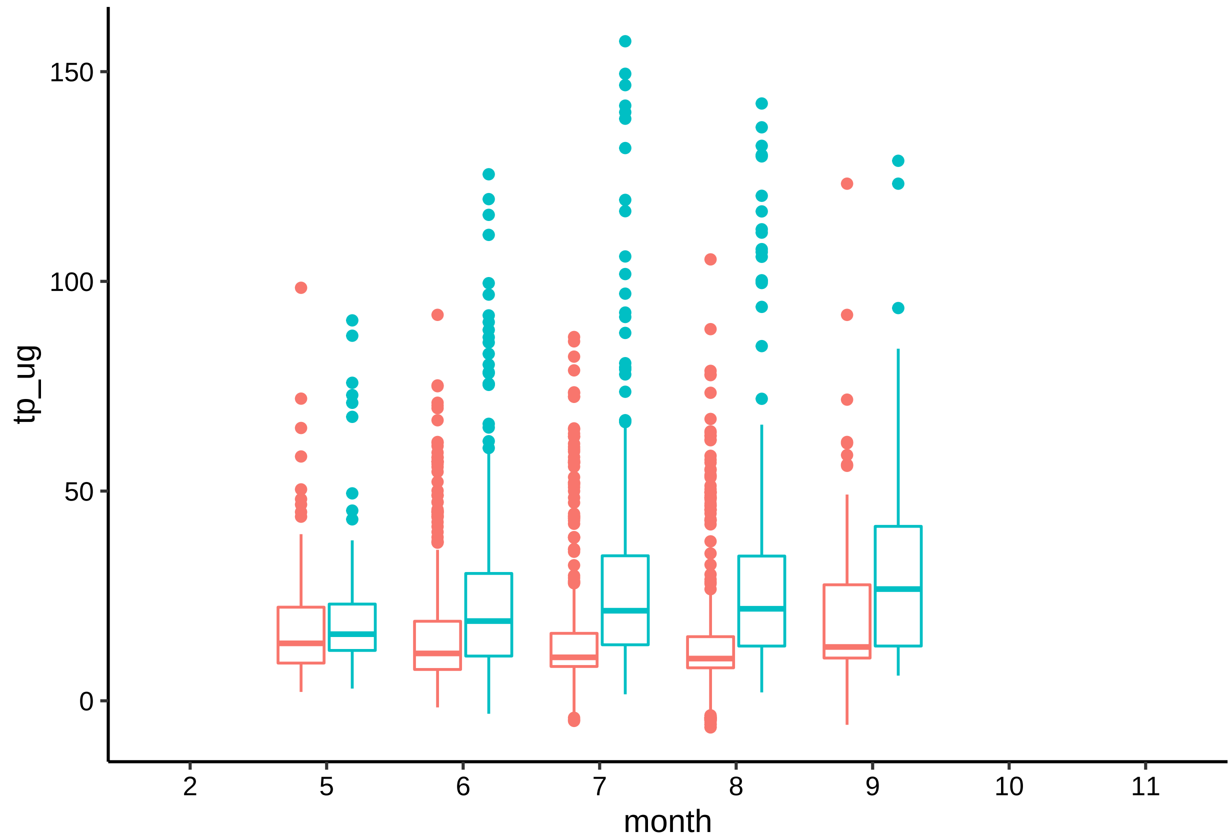
```r
PeterPaul.chem.nutrients$month <- as.factor(PeterPaul.chem.nutrients$month)

#boxplot for a - temperature
boxplot.temp <- ggplot(PeterPaul.chem.nutrients, aes( x = month,
                                                      y = temperature_C, )) +
  theme(legend.position="none") + #used to remove legend
  geom_boxplot(aes(color = lakename))
print(boxplot.temp)
```

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



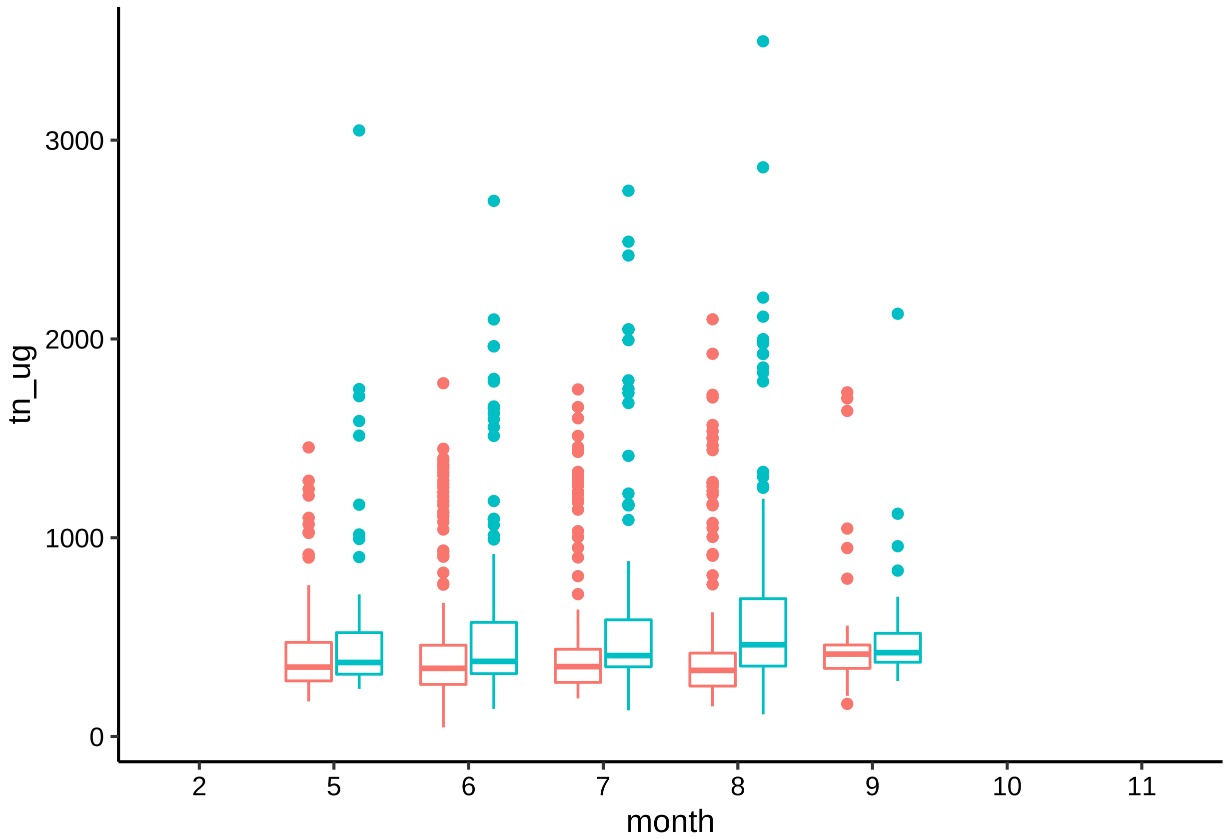

```r
#boxplot for b - TP
boxplot.tp <- ggplot(PeterPaul.chem.nutrients, aes( x = month,
                                                    y = tp_ug, )) +
  theme(legend.position="none") + #used to remove legend
  geom_boxplot(aes(color = lakename))
print(boxplot.tp)
```

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




```
#boxplot for c - TN
boxplot.tn <- ggplot(PeterPaul.chem.nutrients, aes( x = month,
                                                   y = tn_ug, )) +
  theme(legend.position="none") + #used to remove legend
  geom_boxplot(aes(color = lakename))
print(boxplot.tn)
```

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



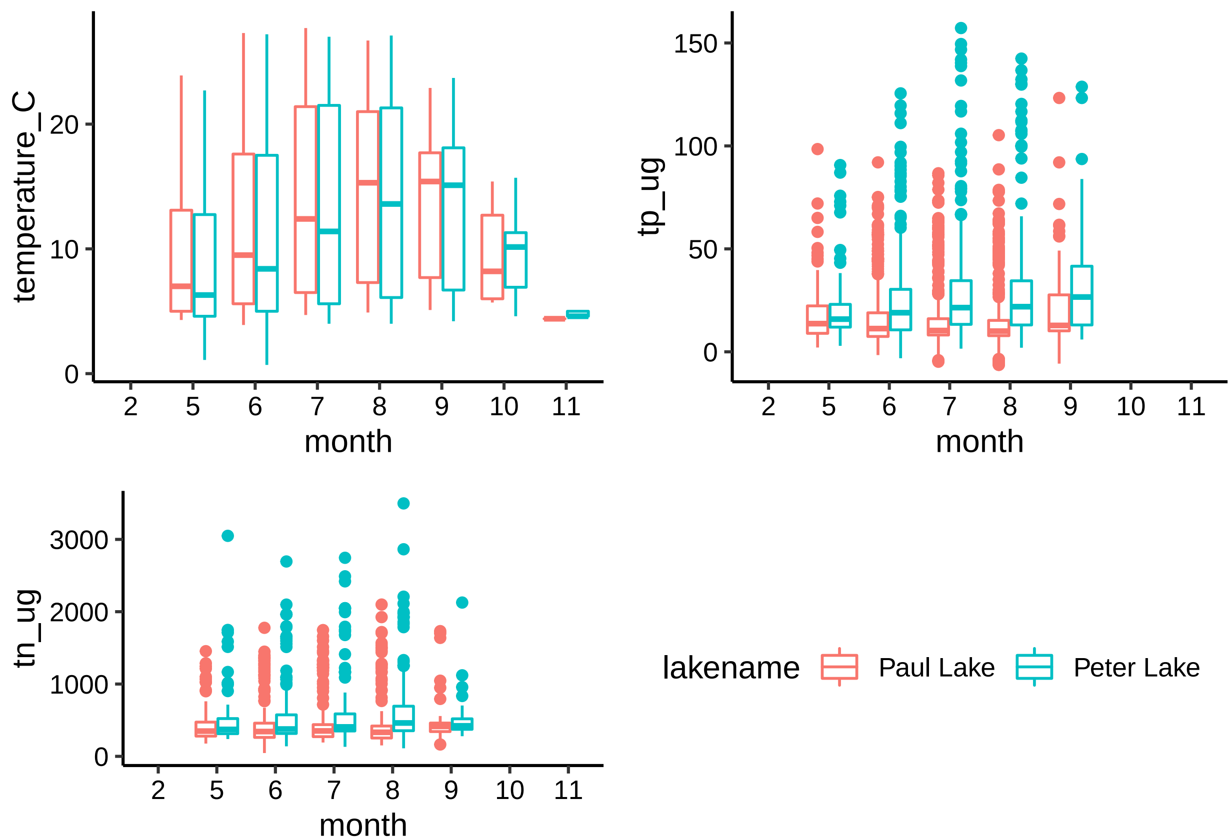

```
#boxplot for just a legend
boxplot.temp.legend <- ggplot(PeterPaul.chem.nutrients, aes( x = month,
                                                              y = temperature_C, )) +
  geom_boxplot(aes(color = lakename))

cowplot.legend <- get_legend(boxplot.temp.legend) #create new element for legend
```

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

```
#cowplot that combines all 3 boxplots and legend
cowplot.temp.tn.tp <- plot_grid(boxplot.temp, boxplot.tp, boxplot.tn,
                                cowplot.legend,
                                nrow = 2, align = 'h', rel_heights = c(1.25, 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).

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

```
print(cowplot.temp.tn.tp)
```

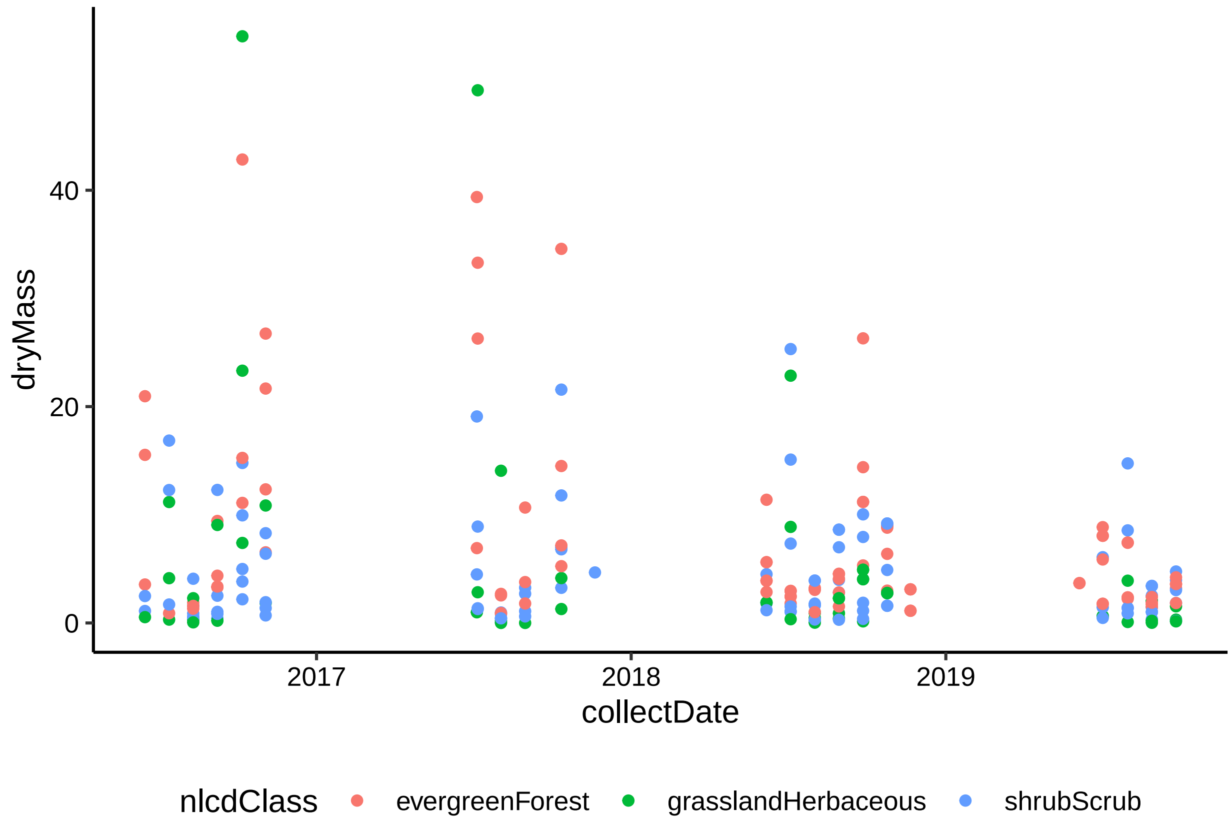


Question: What do you observe about the variables of interest over seasons and between lakes?

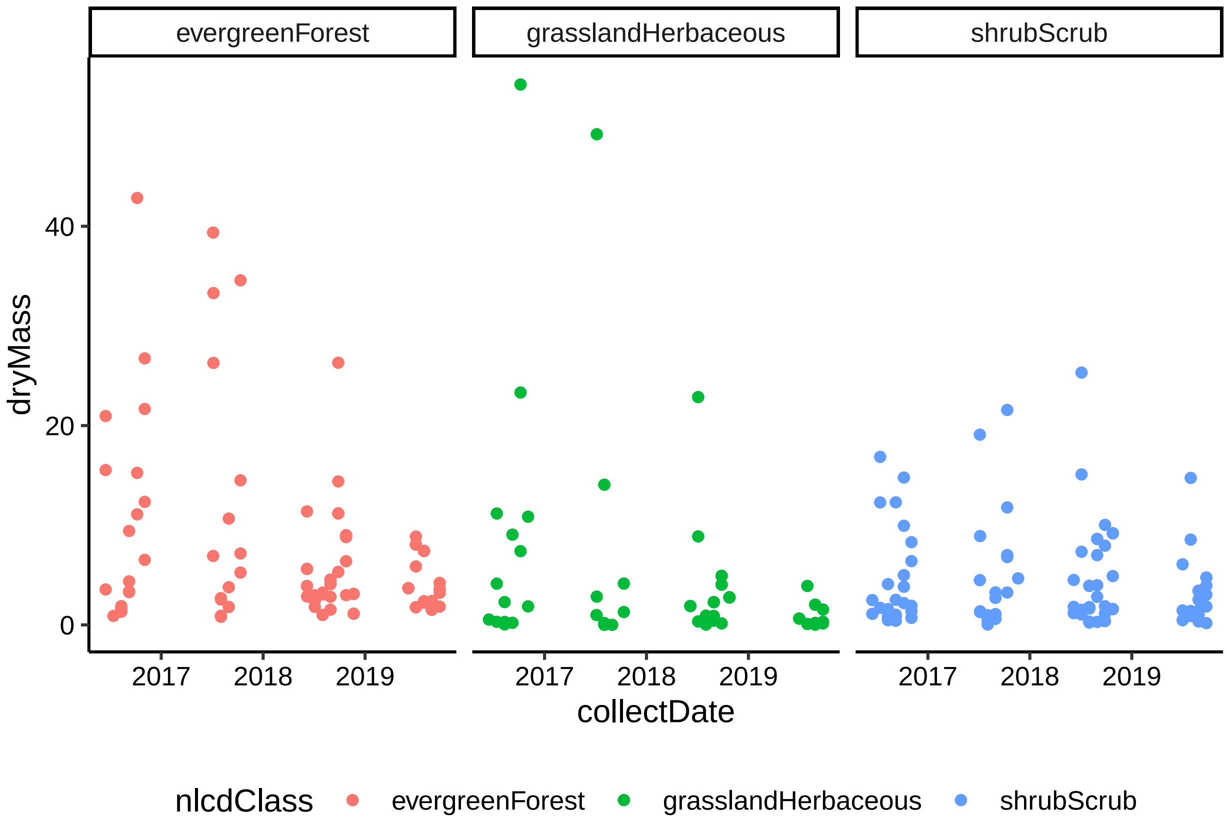
> Answer: It appears that temperature_C for both lakes is pretty consistant however, larger variation between the two lakes is shown in late fall (month 10 - Oct). It also appears that Peter Lake has more frequent outliers in tp_ug and tn_ug than Paul Lake. In the warmest months (7 - Jul, 8 - Aug), tp_ug is also highest which seems to sugest a correlation between the two. However, tn_ug does not follow that pattern as distinctly. It Appears that tn_ug is highest in month 8-Aug but only slightly so.

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
Litter.Needles.Subset <- ggplot(filter(Niwot.Ridge.litter,
                                        functionalGroup == 'Needles')) +
  geom_point(aes( x= collectDate, y = dryMass, color = nlcdClass))

print(Litter.Needles.Subset)
```




```
#old code where subset section didn't work so used filter instead
#Litter.Needles.Subset <- ggplot(subset(Niwot.Ridge.litter, funtionalGroup ==
#"Needles"), aes(x = collectDate, y = dryMass, color = nlcdClass)) + geom_point()

#7
Litter.Needles.Subset.facets <- ggplot(filter(Niwot.Ridge.litter,
                                              functionalGroup == 'Needles')) +
  geom_point(aes( x= collectDate, y = dryMass, color = nlcdClass)) +
  facet_wrap('nlcdClass')

print(Litter.Needles.Subset.facets)
```



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

Answer: I think plot 7 is more effective. It is easier to determine quickly, what land type had higher levels of drymass comparatively - or if the data samples cluster more densely across land type. However, with the plot for #6, it is easier to see that overall drymass decreased over time, and the lowest levels at the end of 2019. Overall though, plot 7 shows how variation in drymass differs among land type clearer than plot 6 and also shows how drymass decreased over time.