Assignment 5: Data Visualization

Lambert Ngenzi

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.

```
#
#getwd()
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)
NTL_LTER_processed <-
read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)
NEON_NIWO_Processed <-
read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)
```

```
#2
NTL_LTER_processed$sampledate <- as.Date(NTL_LTER_processed$sampledate,
                                         format = "%Y-%m-%d")
class(NTL_LTER_processed$sampledate)
```

```
## [1] "Date"
```

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

```
## [1] "Date"
```

```
colSums(!is.na(NEON_NIWO_Processed))
```

```
##           plotID           trapID      collectDate  functionalGroup
##             1692             1692             1692             1692
##          dryMass        qaDryMass        subplotID  decimalLatitude
##             1692             1692             1692             1692
## decimalLongitude        elevation        nlcdClass         plotType
##             1692             1692             1692             1692
##    geodeticDatum
##             1692
```

Define your theme

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

```
#3 Setting my theme with some elements
Mytheme <- theme_classic(base_size = 16) +
  theme(axis.text = element_text(color = "Black"),
        legend.position = "right")
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()`).

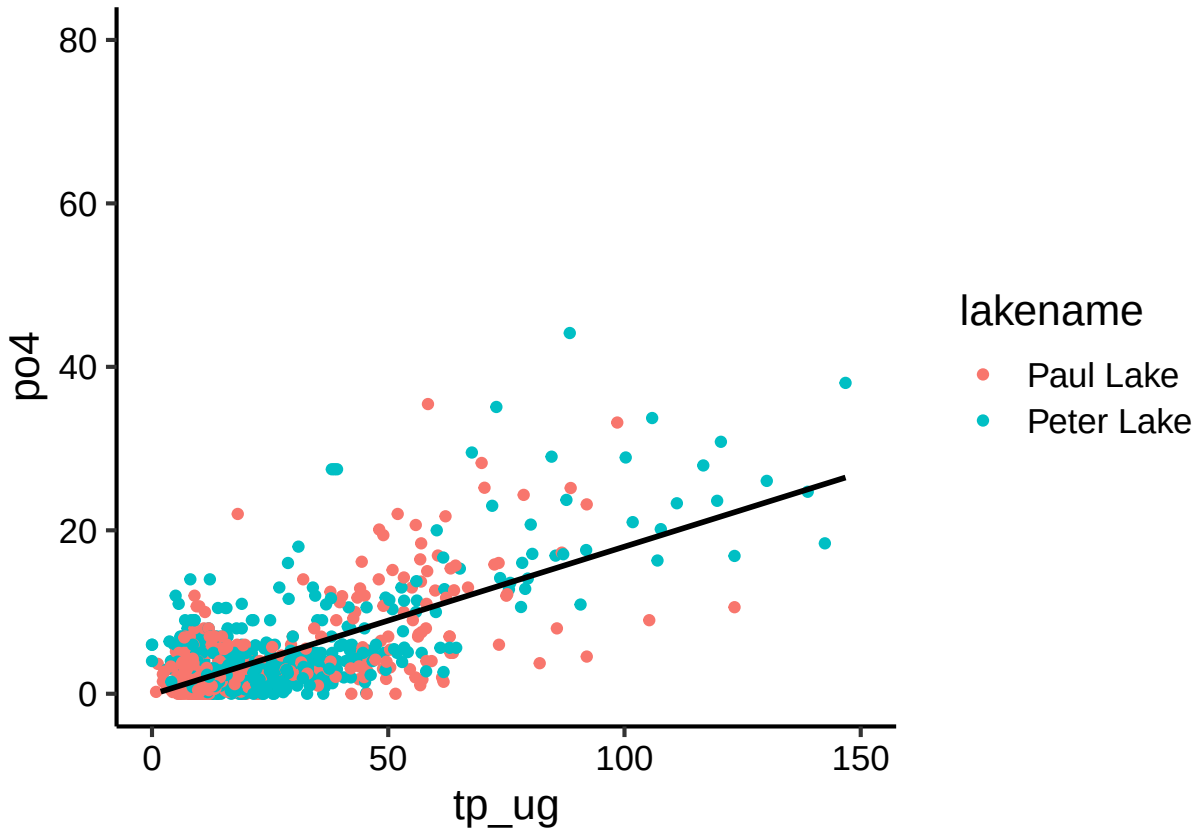
```
#4 Plot of Total Phosphorus by phosphate
tp_po4_plot <- ggplot(NTL_LTER_processed, aes(x = tp_ug, y = po4,
                                              color = lakename)) +
   geom_point() +
  geom_smooth(method='lm', se = FALSE, col = "black") +
  xlim(0,150) +
  ylim(0,80)
  print(tp_po4_plot)
```

```
## `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).
```

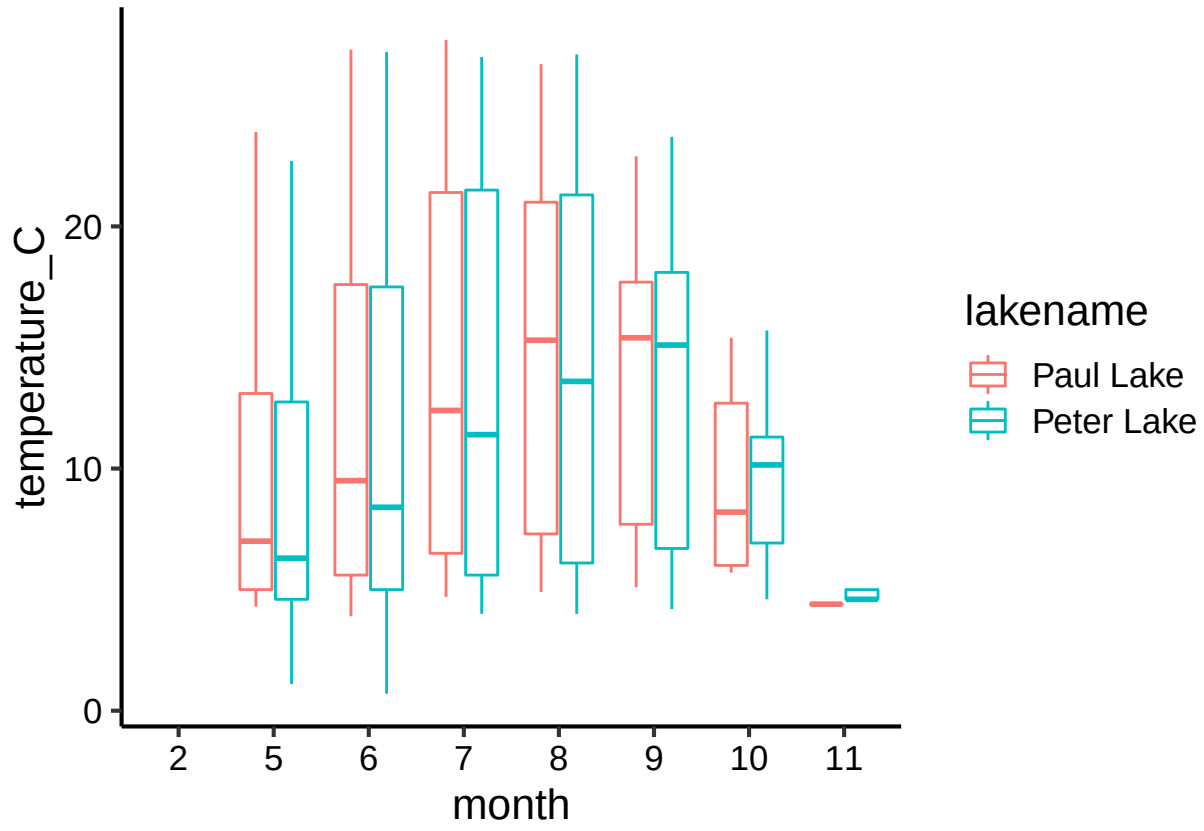
```
## Warning: Removed 1 rows containing missing values (geom_smooth).
```



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.

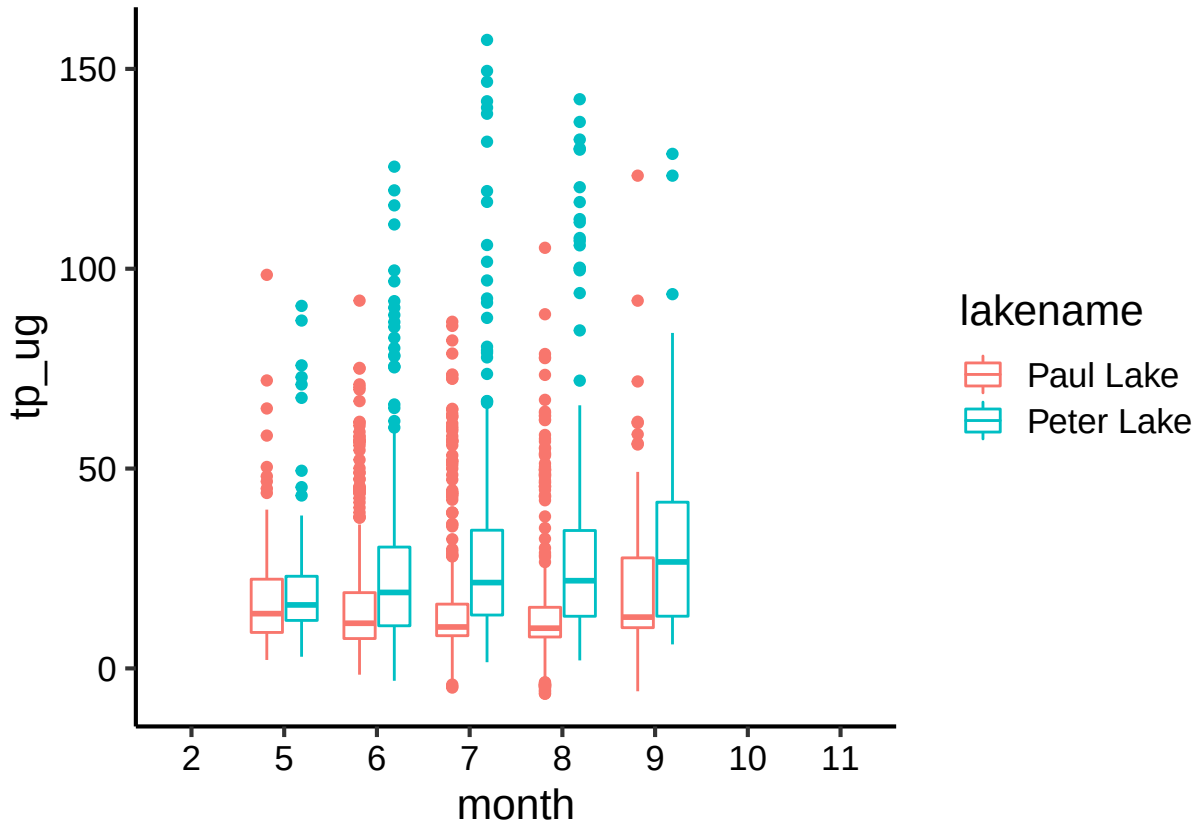
```
#5 Making 3 separated boxplots with a combined at the end
NTL_LTER_processed$month <- as.factor(NTL_LTER_processed$month)

Temp_plot <- ggplot(NTL_LTER_processed, aes(month, temperature_C,
                                            color = lakename)) +
  geom_boxplot()
print(Temp_plot)
```

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



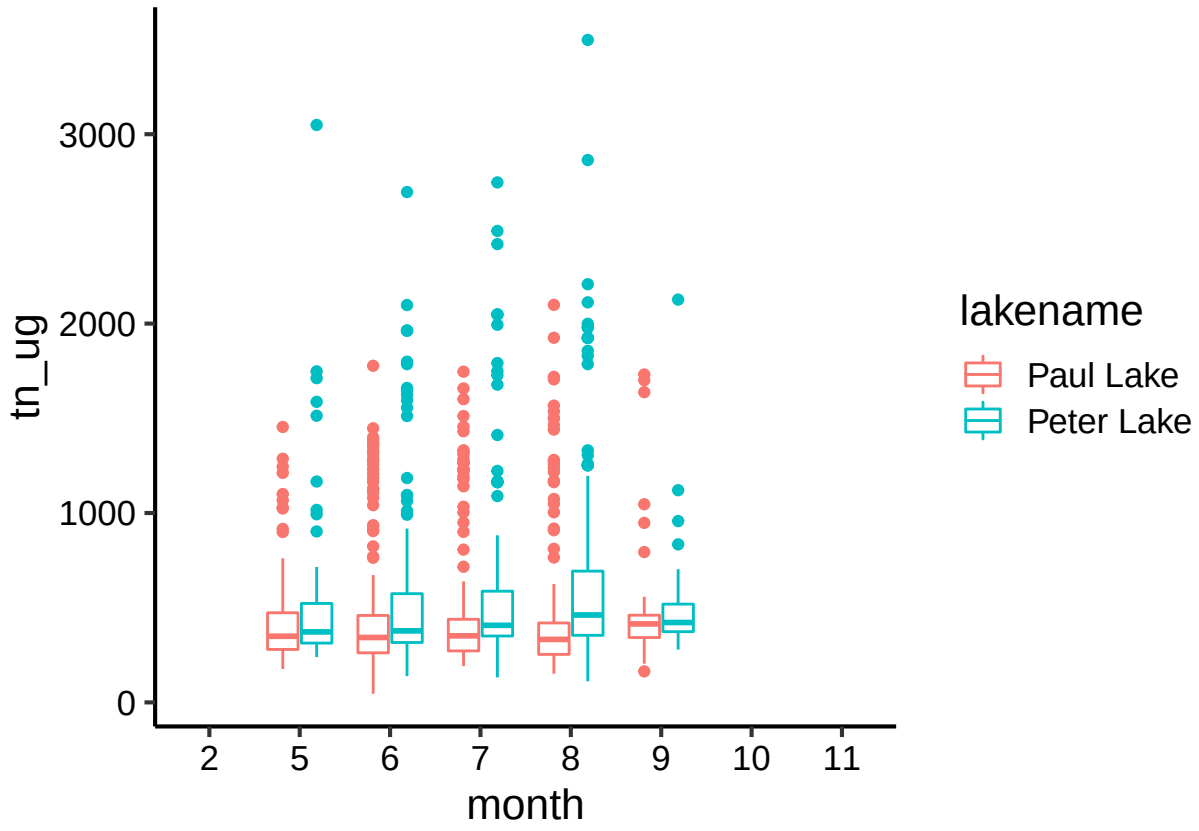

```
TP_Plot <- ggplot(NTL_LTER_processed, aes(month, tp_ug, color = lakename)) +
  geom_boxplot()
print(TP_Plot)
```

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




```
TN_Plot <- ggplot(NTL_LTER_processed, aes(month, tn_ug, color = lakename)) +
  geom_boxplot()
print(TN_Plot)
```

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



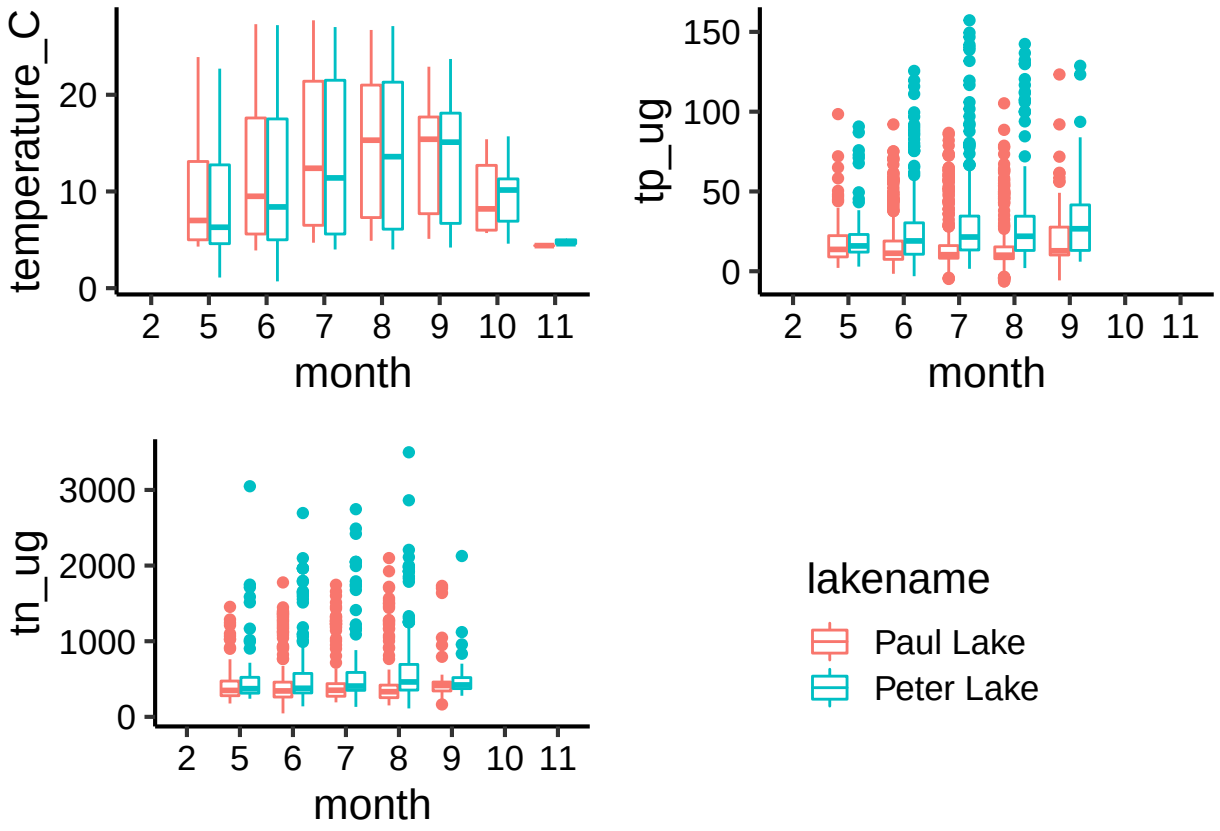

```
mylegend <- get_legend(Temp_plot)
```

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

```
CombineGraph <- plot_grid(Temp_plot + theme(legend.position = "none"),
                          TP_Plot + theme(legend.position = "none"),
                          TN_Plot + theme(legend.position = "none"),
                          mylegend, nrow = 2, align = 'h',
                          rel_heights = c(.75, .75), axis = "bt")
```

```
## 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).
```

```
print(CombineGraph)
```



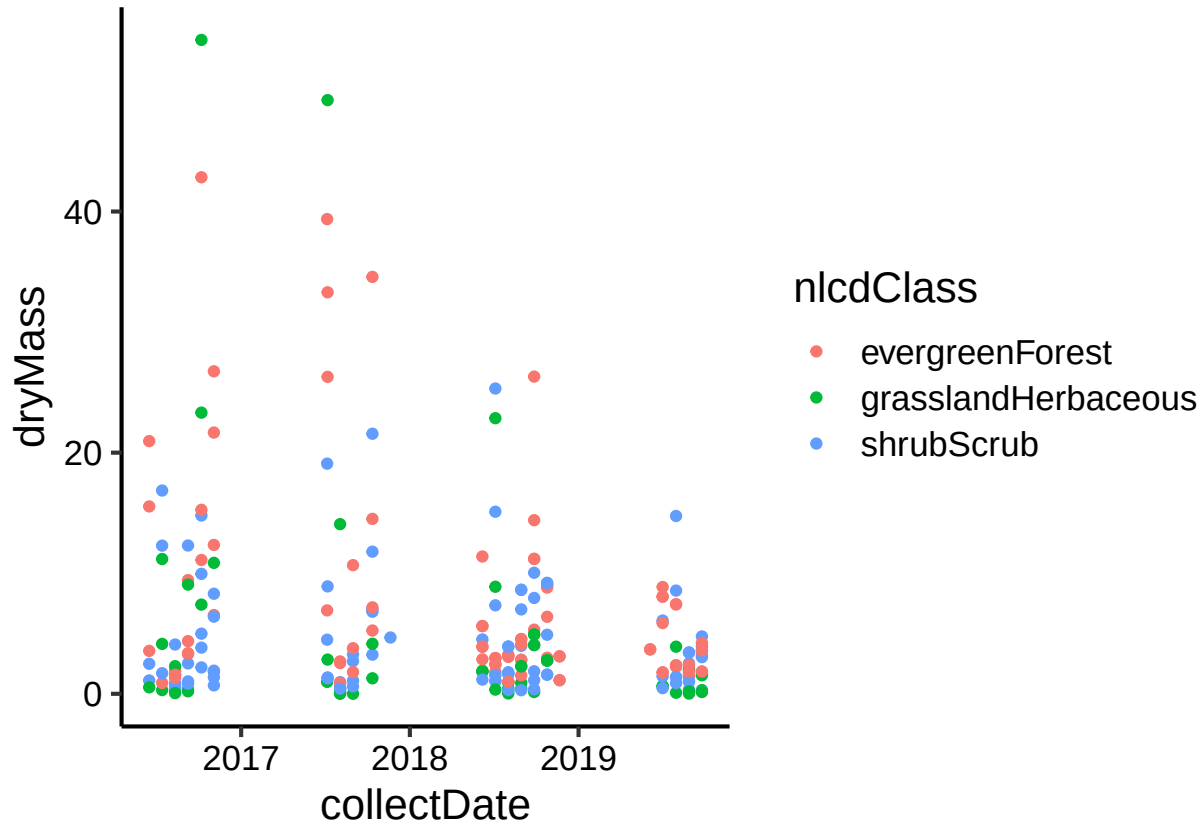
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The mean distributions are sentive to season (for example, for both lakes, temperature peaks around summer (July and August) and starts decreasing in September and onward. While Total phosphorus and Total nutrients are less sensitive to season in comparison ot temperature. But we see a linear increase in mean of total phosphorus over season in Peter lake and less in Paul lake While Total Nutrients mean in Peter lake peaks in mid August and stays consistent in Paul lake. These differences might be explained by lakes depths as well.

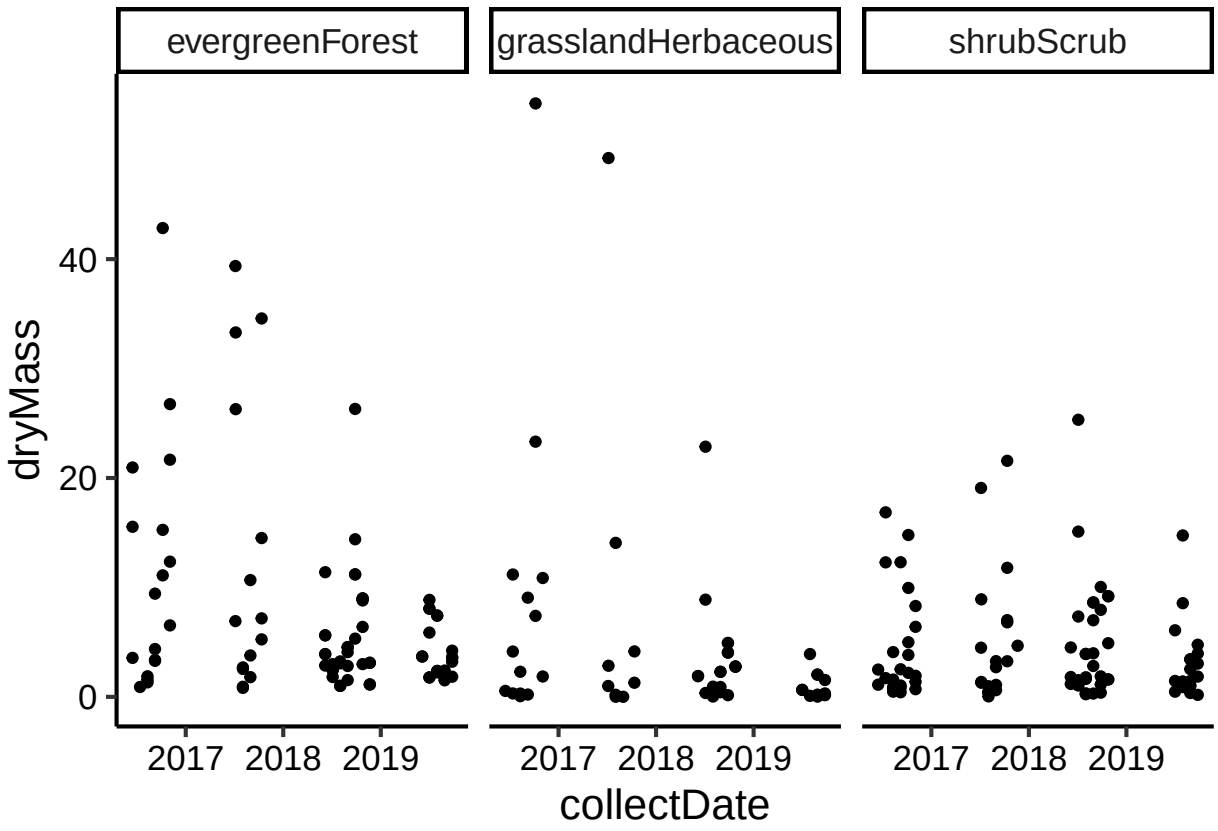
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
NiwotRidge_Litter <- ggplot(subset(NEON_NIWO_Processed,
                                   functionalGroup == "Needles"),
                            aes(x = collectDate, y = dryMass,
                                color= nlcdClass)) + geom_point()

print(NiwotRidge_Litter)
```




```
#7
NiwotRidge_Litter_Facet <- ggplot(subset(NEON_NIWO_Processed,
                                         functionalGroup == "Needles"),
                            aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass))
print(NiwotRidge_Litter_Facet)
```



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

Answer:Personaly, I prefer the facet wrap graph in question 7 becaause it helps the reader to understand what is going each year needles were collected While graph in question 6 is visually appearing, it is hard to understand what is going on in the data.