Assignment 5: Data Visualization

Natasha Jacob

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 Checking working directory
getwd()
```

```
## [1] "/Users/admin/Desktop/ENV872_EDA/Environmental_Data_Analytics_2022"
```

```
# Loading tidyverse and cowplot packages
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)
# Uploading the processed data files
PeterPaul_Processed <- read.csv("../Environmental_Data_Analytics_2022/Data/Processed/NTL-LTER_Lake_Chem
    stringsAsFactors = TRUE)
Litter_Processed <- read.csv("../Environmental_Data_Analytics_2022/Data/Processed/NEON_NIWO_Litter_mass
    stringsAsFactors = TRUE)

# 2 Changing date to date format
PeterPaul_Processed$sampledate <- as.Date(PeterPaul_Processed$sampledate,
    format = "%Y-%m-%d")
Litter_Processed$collectDate <- as.Date(Litter_Processed$collectDate,
    format = "%Y-%m-%d")

# converting month as a factor
PeterPaul_Processed$month <- as.factor(PeterPaul_Processed$month)

# Checking the class of month
class(PeterPaul_Processed$month)
```

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

Define your theme

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

```
# 3 Building a theme
mytheme <- theme_classic(base_size = 14) + theme(axis.text = element_text(color = "black"),
    legend.position = "top")
theme_set(mytheme)
```

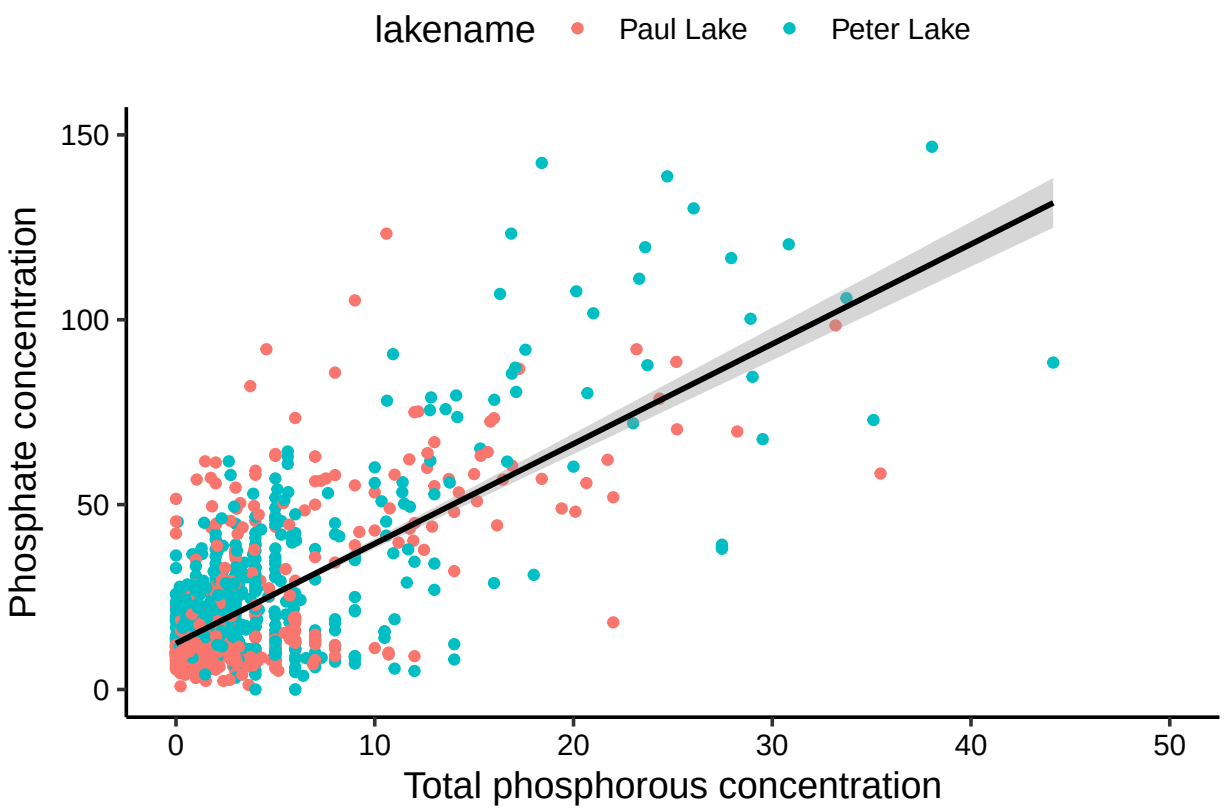
Create graphs

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 Plotting total phosphorous by phosphate for Peter and
# Paul Lakes
Total_phosphorous_by_phosphate <- ggplot(PeterPaul_Processed,
    aes(x = po4, y = tp_ug)) + geom_point(aes(color = lakename)) +
    geom_smooth(method = lm, color = "black") + xlim(0, 50) +
    ylim(0, 150) + ylab(expression("Phosphate concentration")) +
    xlab(expression("Total phosphorous concentration")) + mytheme
print(Total_phosphorous_by_phosphate)
```

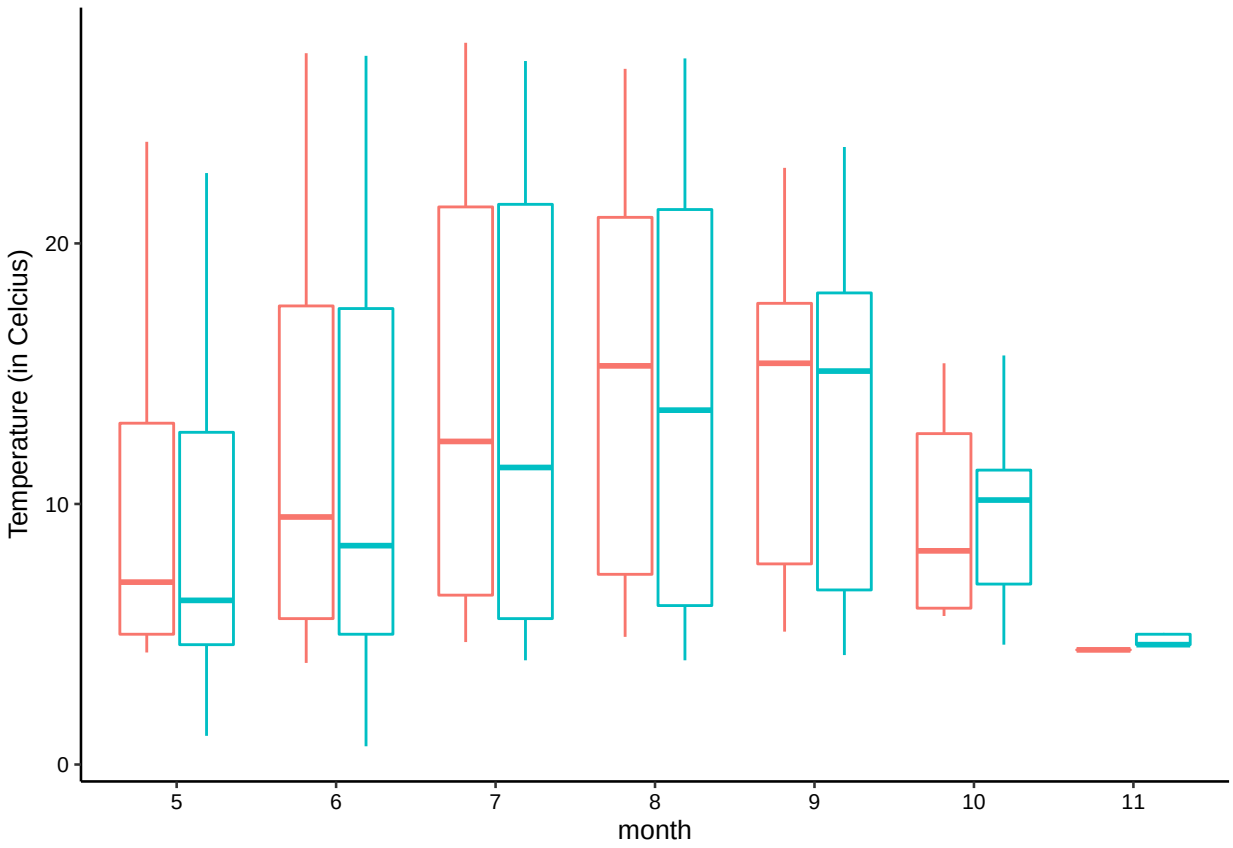
```
## `geom_smooth()` using formula 'y ~ x'

## 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 Creating three separate boxplots of month versus
# temperature, total phosphorous and total nitrogen.
# Legends were not generated for these plots.

Temperature_boxplot <- ggplot(PeterPaul_Processed, aes(x = month,
    y = temperature_C)) + geom_boxplot(aes(color = lakename)) +
    scale_x_discrete(limits = factor("5":"11")) + theme_classic(base_size = 10) +
    ylab(expression("Temperature (in Celcius)")) + theme(axis.text = element_text(color = "black"),
    legend.position = "none")
print(Temperature_boxplot)
```

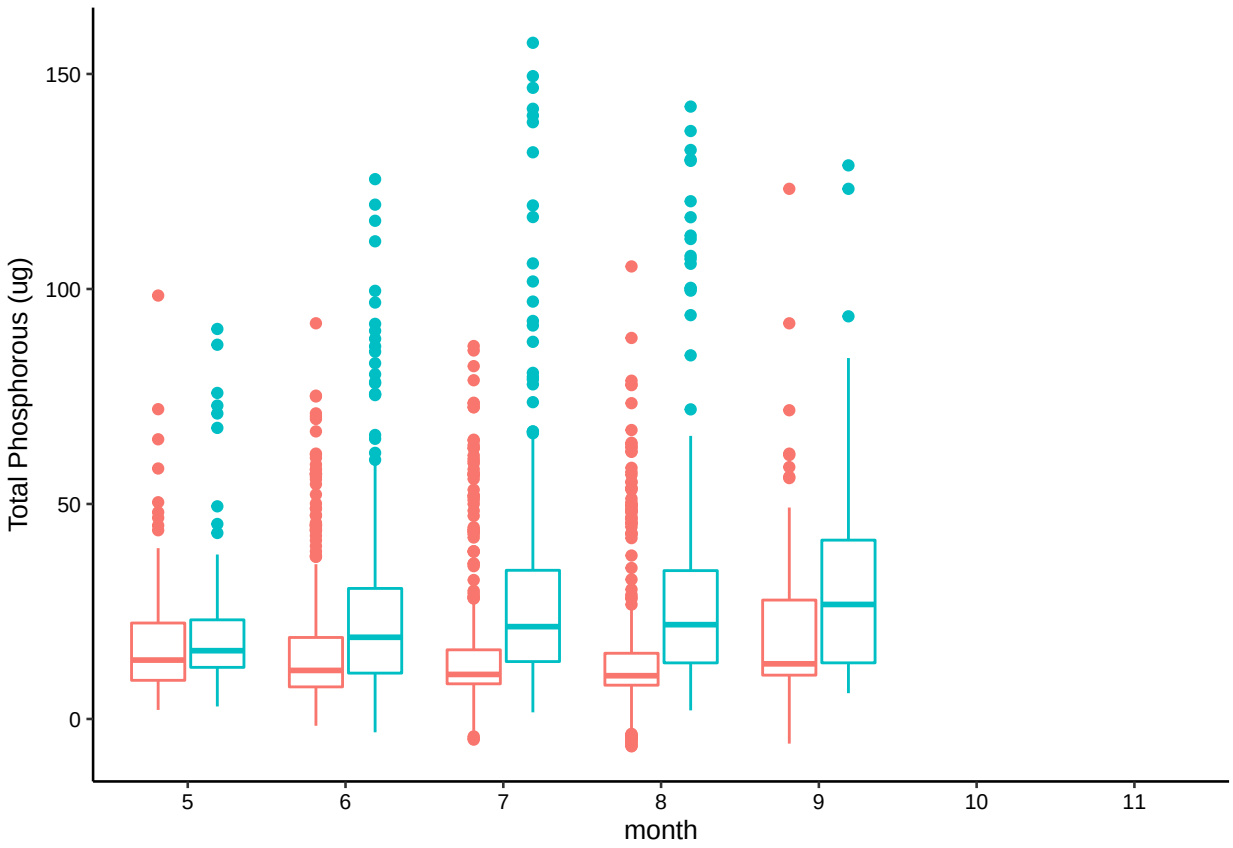
```
## Warning: Removed 16 rows containing missing values (stat_boxplot).

## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```




```
Phosphorous_boxplot <- ggplot(PeterPaul_Processed, aes(x = month,
    y = tp_ug)) + geom_boxplot(aes(color = lakename)) + scale_x_discrete(limits = factor("5":"11")) +
    theme_classic(base_size = 10) + ylab(expression("Total Phosphorous (ug)")) +
    theme(axis.text = element_text(color = "black"), legend.position = "none")
print(Phosphorous_boxplot)
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
```

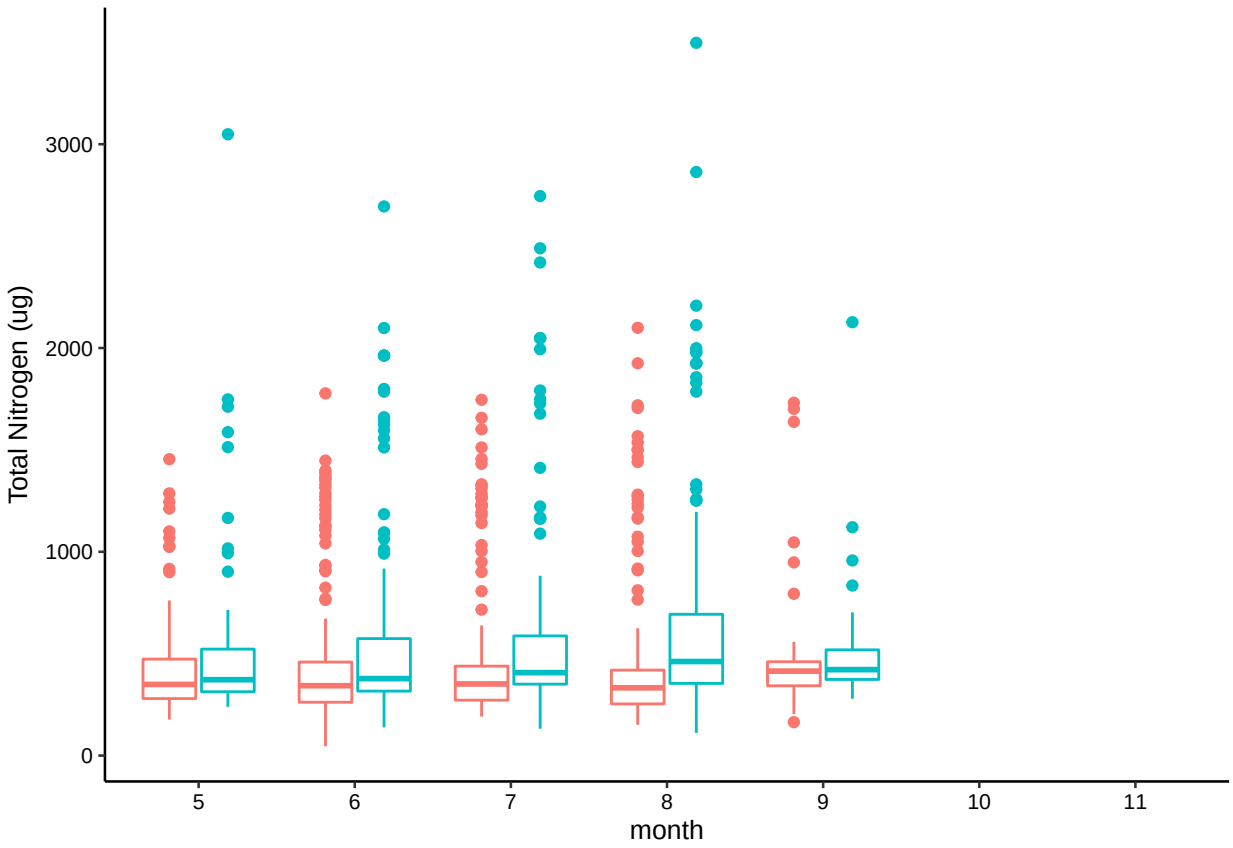
```
## Warning: Removed 20713 rows containing non-finite values (stat_boxplot).
```




```
Nitrogen_boxplot <- ggplot(PeterPaul_Processed, aes(x = month,
    y = tn_ug)) + geom_boxplot(aes(color = lakename)) + scale_x_discrete(limits = factor("5":"11")) +
    ylab(expression("Total Nitrogen (ug)")) + theme_classic(base_size = 10) +
    theme(axis.text = element_text(color = "black"), legend.position = "none")
print(Nitrogen_boxplot)
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 21567 rows containing non-finite values (stat_boxplot).
```




```
# Creating a separate boxplot of month vs temperature to
# obtain the legend. (Notice that legend position is
# specified as 'top' in the code)
Temperature_boxplot1 <- ggplot(PeterPaul_Processed, aes(x = month,
    y = temperature_C)) + geom_boxplot(aes(color = lakename)) +
    scale_x_discrete(limits = factor("5":"11")) + theme_classic(base_size = 10) +
    theme(axis.text = element_text(color = "black"), legend.position = "bottom")

# Using the get_legend() function to obtain the legend for
# the final combined graph
NTL_legend <- get_legend(Temperature_boxplot1)
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```

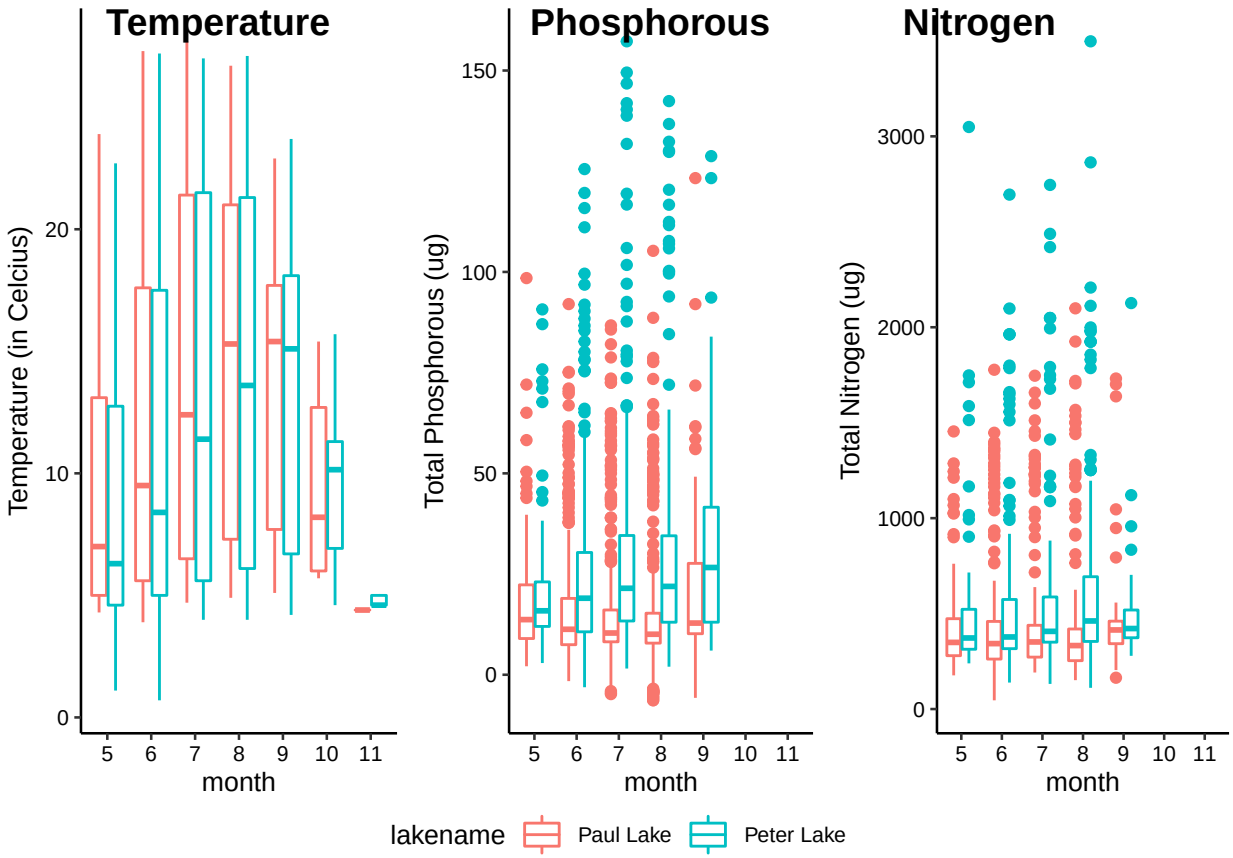
```
# Combining the three boxplots (without the legend)
plot1 <- plot_grid(Temperature_boxplot, Phosphorous_boxplot,
    Nitrogen_boxplot, labels = c("Temperature", "Phosphorous",
        "Nitrogen"), ncol = 3, rel_heights = c(3, 0.3))
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
## Removed 3550 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 16 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 20713 rows containing non-finite values (stat_boxplot).

## Warning: Removed 16 rows containing missing values (stat_boxplot).

## Warning: Removed 21567 rows containing non-finite values (stat_boxplot).
```

```
# Combining three boxplots with one legend present
Combined_plot <- plot_grid(plot1, NTL_legend, ncol = 1, nrow = 2,
    rel_heights = c(5, 0.3))
Combined_plot
```



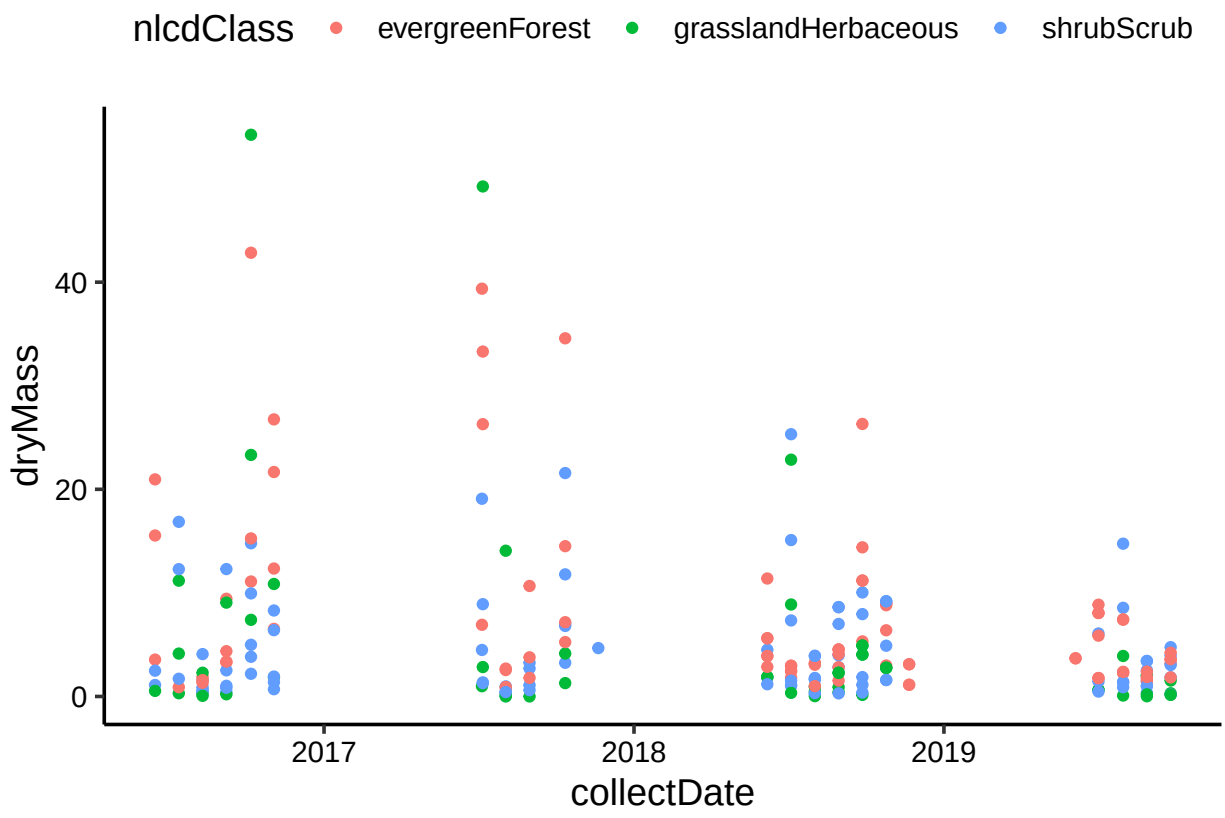
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature plot: A pattern of normally distributed values can be observed in the temperature plot (for both lakes) as the temperature seems to increase starting from May, reaching its peak in July and a slow decline starting from August can be observed. The lowest recorded temperture, however, can be observed in the month of November. Phosphorous plot: A steady increase in phosphorous content can be observed for Peter Lake. However, a steady decrease in phosphorous content up until August can be seen in Paul Lake after which the phosphorous content seems to increase in September. Nitrogen plot: There isn't much variation in Nirogen levels observed in Peter and Paul Lakes excpet for the minor incline for Peter Lake observed in the month of August.

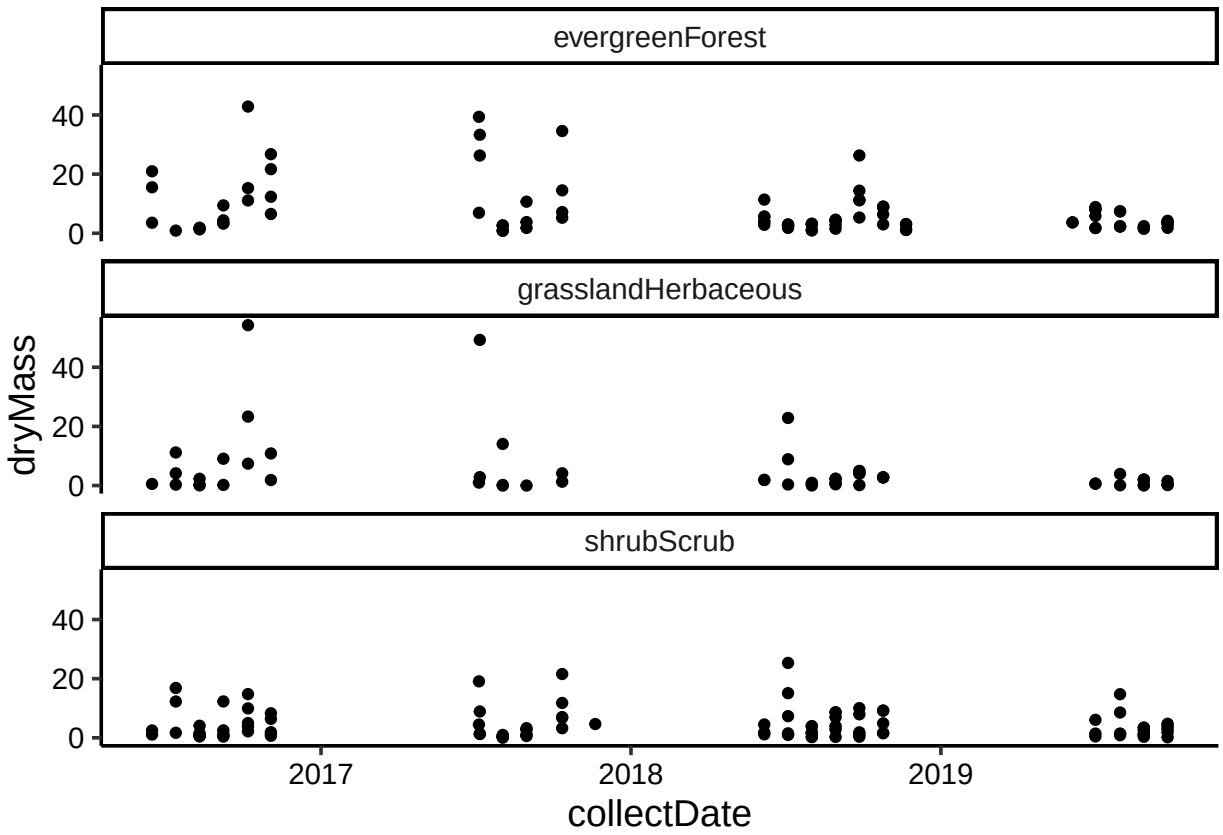
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
# 6 Plotting dry mass of needle by Date, separated by NLCD
# class (using a color aesthetic)
DryMass_by_date_nlcd <- ggplot(subset(Litter_Processed, functionalGroup ==
    "Needles"), aes(x = collectDate, y = dryMass, color = nlcdClass)) +
    geom_point() + mytheme
print(DryMass_by_date_nlcd)
```




```
# 7 Plotting dry mass of needle by Date with NLCD classes
# separated into three facets
DryMass_by_date_facet <- ggplot(subset(Litter_Processed, functionalGroup ==
    "Needles"), aes(x = collectDate, y = dryMass)) + facet_wrap(vars(nlcdClass),
    nrow = 3) + geom_point() + mytheme
print(DryMass_by_date_facet)
```



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

Answer: Plot 7 where we plotted dry mass of needle by Date with NLCD classes separated into three facets is a more effective plot to visualise data since the facets make it easier to observe the seperate nlcd classes. In plot #6, it is difficult to visualise data since the points appear clustered.