Assignment 5: Data Visualization

Reed Leon-Hinton

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1 doing the initial setup

# clearing the environment (It's a pet peeve)
remove(list = ls())

getwd() #showing the correct working directory.
```

```
## [1] "C:/Users/shado/Documents/Graduate School Stuff/ENVIRON 872 - Environmental Data Analytics/Envir
```

```
# install.packages("tidyverse")
# install.packages("cowplot")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(cowplot)

# importing the tidy NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv dataset
nutrients <- read.csv(file = "./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
                       stringsAsFactors = TRUE)

# importing the gathered NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv dataset
gathered <- read.csv(file = "./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv",
                      stringsAsFactors = TRUE)

# importing the processed data file for the Niwot Ridge litter dataset
litter <- read.csv(file = "./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                    stringsAsFactors = TRUE)
#2 fixing the incorrectly showing dates

# due to the stringsAsFactors argument, they imported as factors.
class(gathered$sampledate)
```

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

```
# getting lubridate to better handle these conversions
# install.packages("lubridate")
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
```

```
# fixing the date in the gathered dataset
gathered$sampledate <- ymd(gathered$sampledate)

# fixing the date in the litter dataset
litter$collectDate <- ymd(litter$collectDate)

# fixing the date in the nutrients dataset
nutrients$sampledate <- ymd(nutrients$sampledate)

# checking their classes to make certain it was completed correctly
class(gathered$sampledate)
```

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

```
class(litter$collectDate)
```

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

```
class(nutrients$sampledate)
```

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

Define your theme

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

```
# creating a super cool theme for use in this assignment
super_cool_theme <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        axis.title = element_text(color = "firebrick3"),
        panel.background = element_rect(fill = "white", color = "black"),
        panel.grid.major = element_line(size = 0.5, color = "gray93"))

# setting the super_cool_theme as the default
theme_set(super_cool_theme)
```

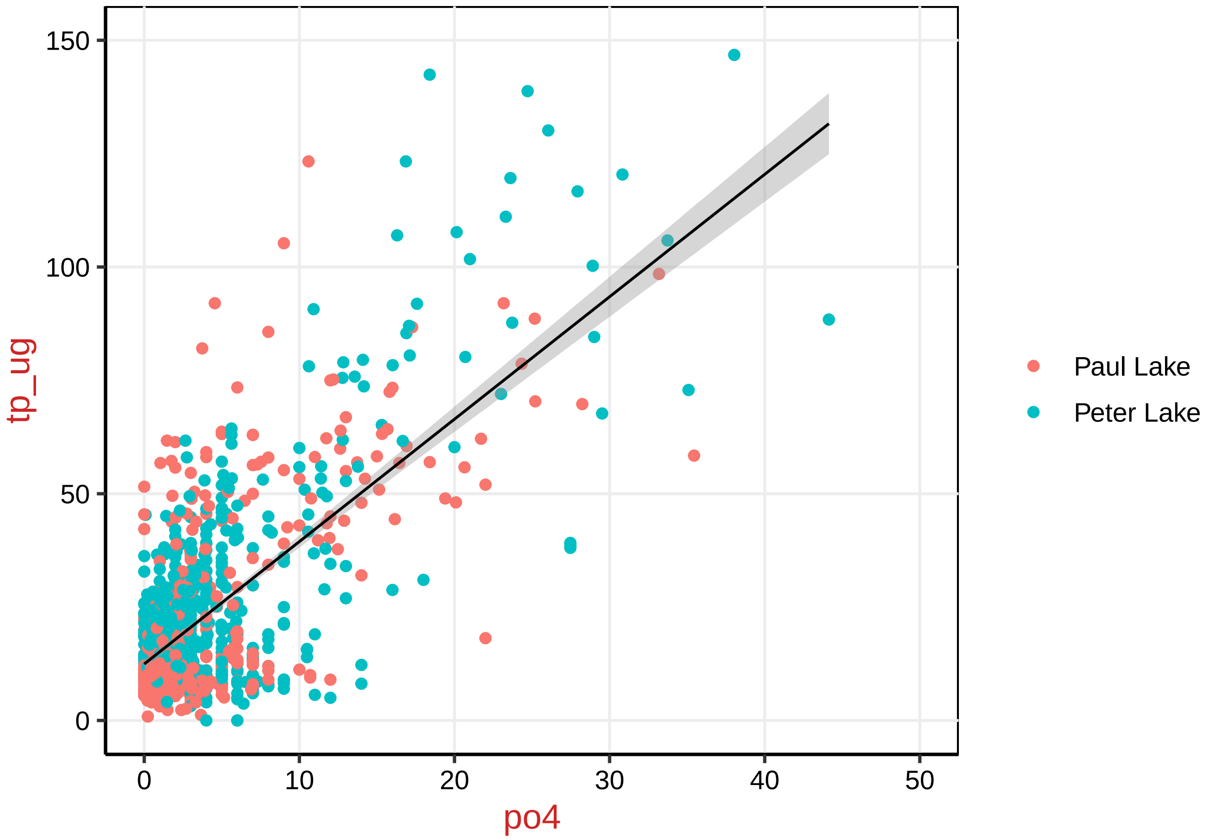
Create graphs

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.

```
# creating the desired graph
tp_po_plot <- ggplot(nutrients, aes(x = po4, y = tp_ug)) +
  geom_point(aes(color = lakename)) +
  xlim(0, 50) +
  ylim(0, 150) +
  geom_smooth(method = lm, color = "black", size = 0.5) +
  theme(legend.title = element_blank())
print(tp_po_plot)
```

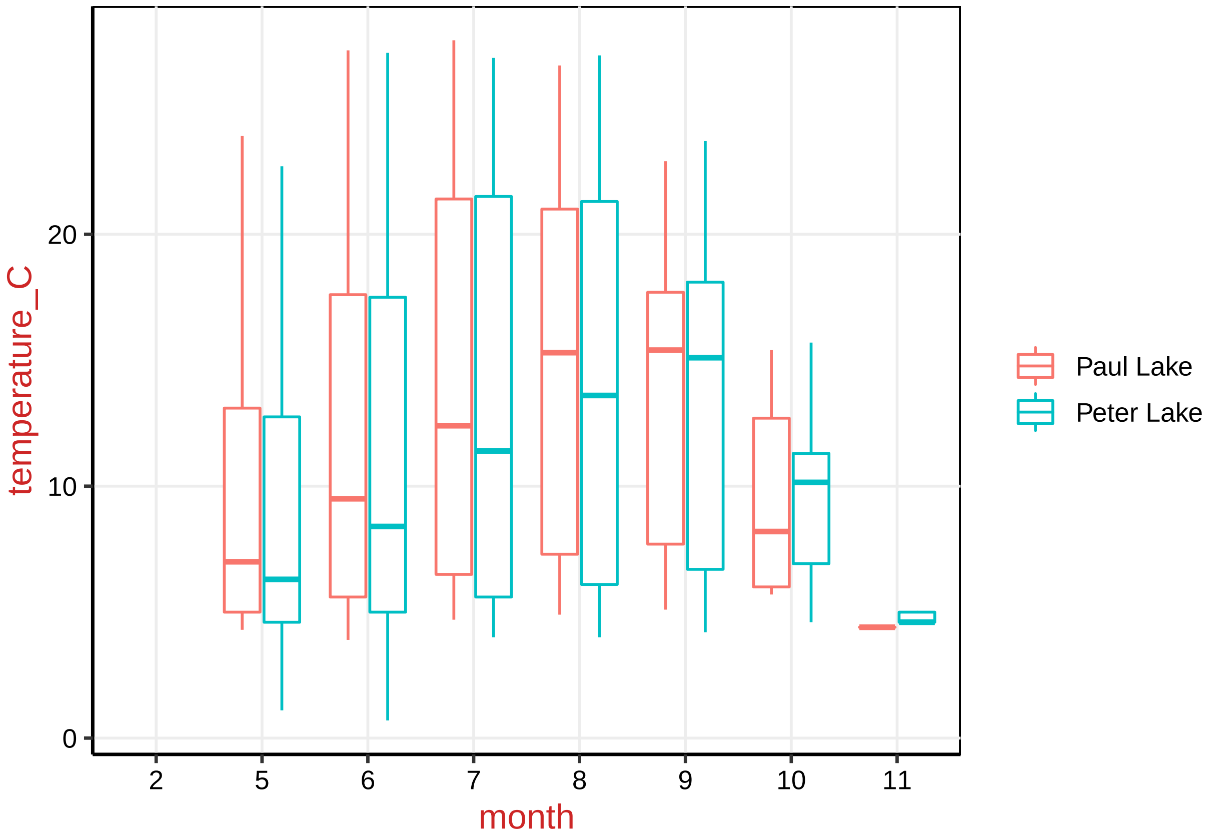
```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 21948 rows containing non-finite values (stat_smooth).

## Warning: Removed 21948 rows containing missing values (geom_point).
```



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.

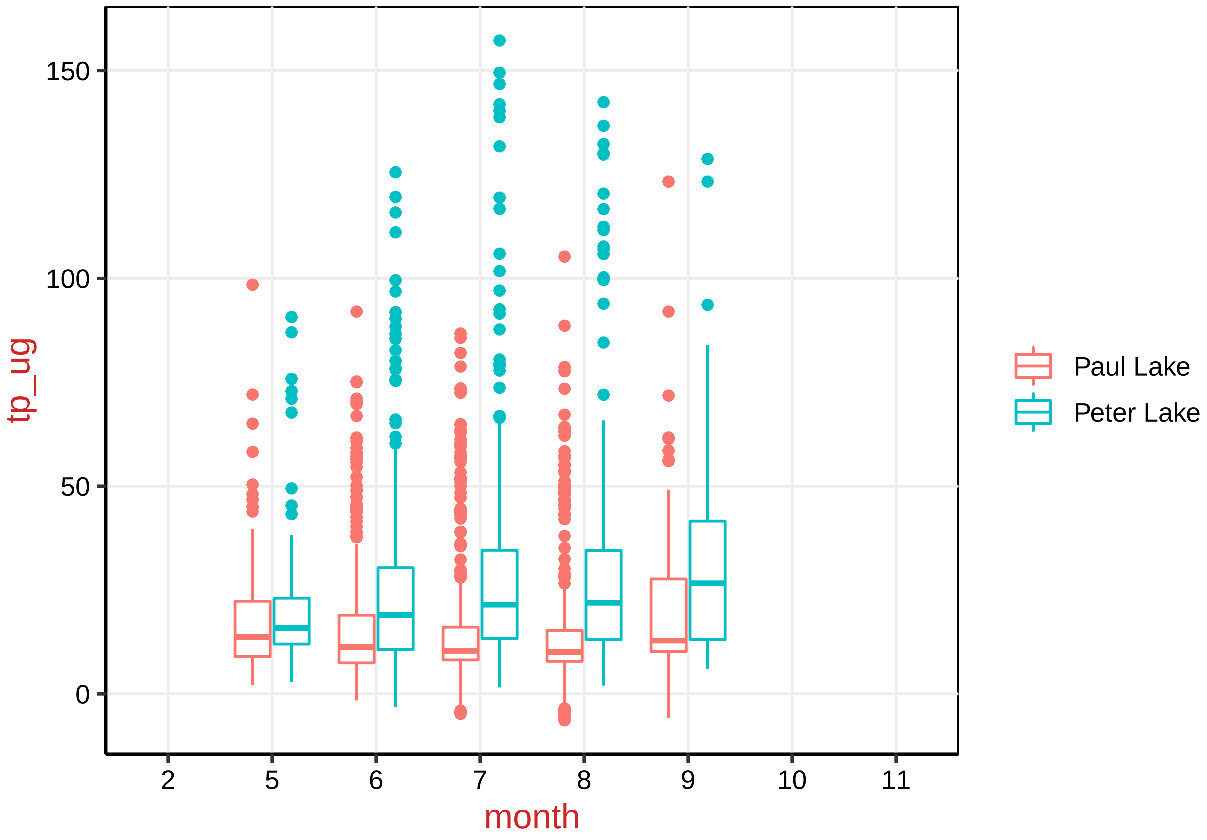
```
# for this to work right, month needs to be a factor
nutrients$month <- as.factor(nutrients$month)

# making the boxplot for temperature
temp_box <- ggplot(nutrients) +
  geom_boxplot(aes(x = month, y = temperature_C, color = lakename)) +
  theme(legend.title = element_blank())
print(temp_box)
```

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



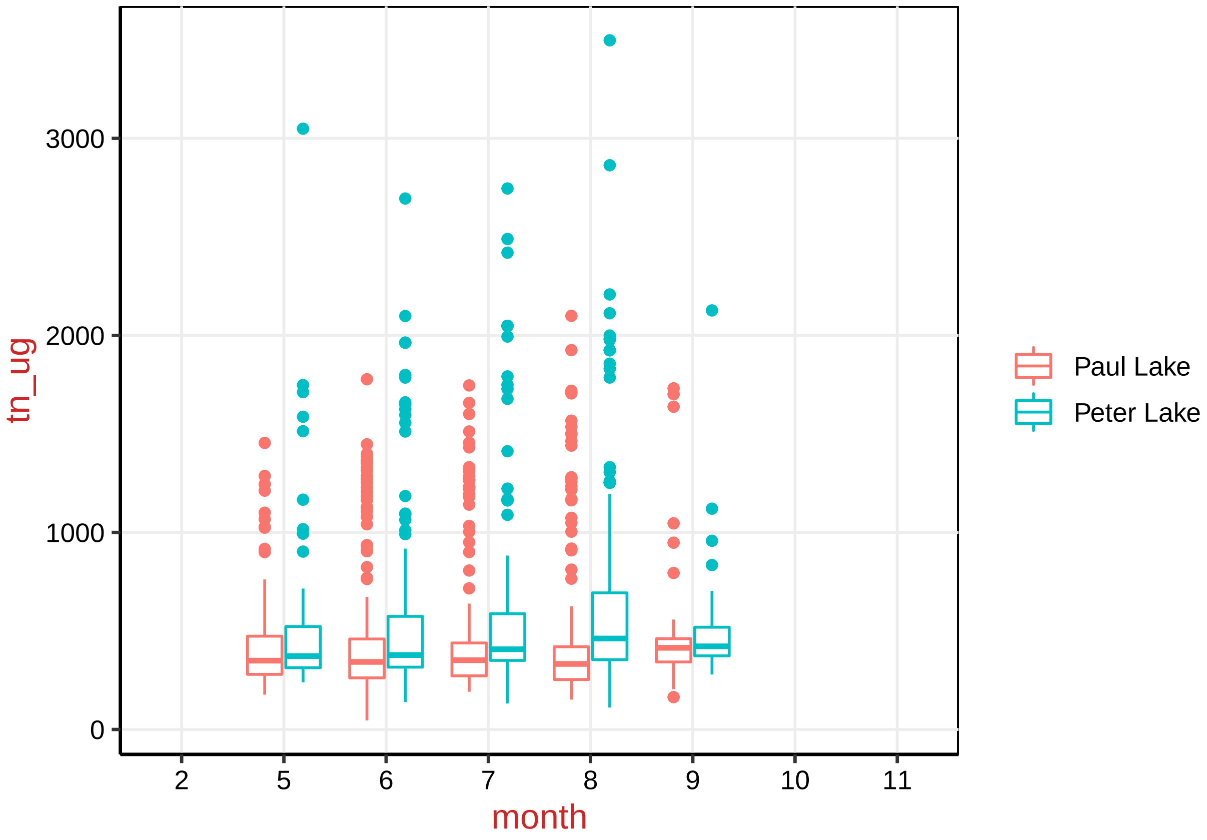
```
# making the boxplot for TP_ug
tp_box <- ggplot(nutrients) +
  geom_boxplot(aes(x = month, y = tp_ug, color = lakename)) +
  theme(legend.title = element_blank())
print(tp_box)
```

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



```
# making the boxplot for TN_ug
tn_box <- ggplot(nutrients) +
  geom_boxplot(aes(x = month, y = tn_ug, color = lakename)) +
  theme(legend.title = element_blank())
print(tn_box)
```

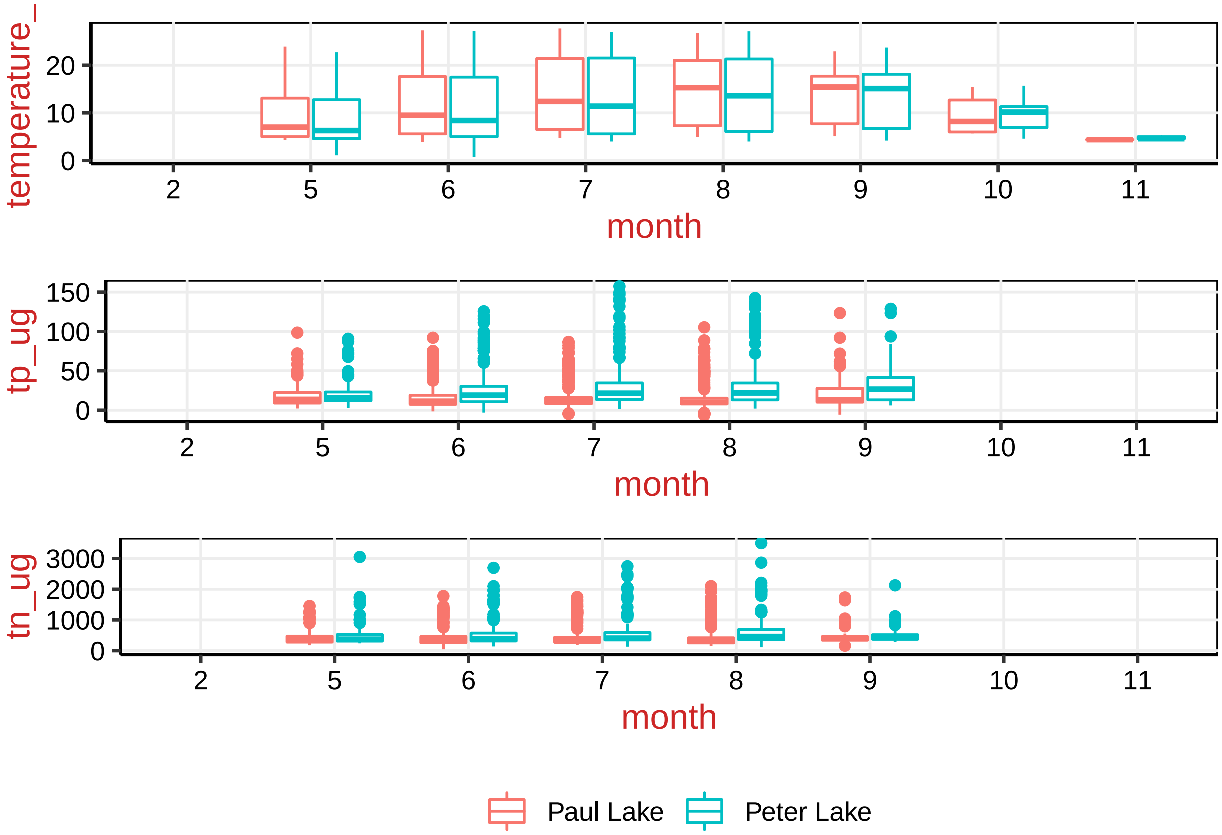
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```



```
combined_box <- plot_grid(
  temp_box + theme(legend.position = "none"),
  tp_box + theme(legend.position = "none"),
  tn_box + theme(legend.position = "bottom", legend.title = element_blank()),
  nrow = 3,
  rel_heights = c(1, 1, 1.35)
)
```

```
## 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(combined_box)
```



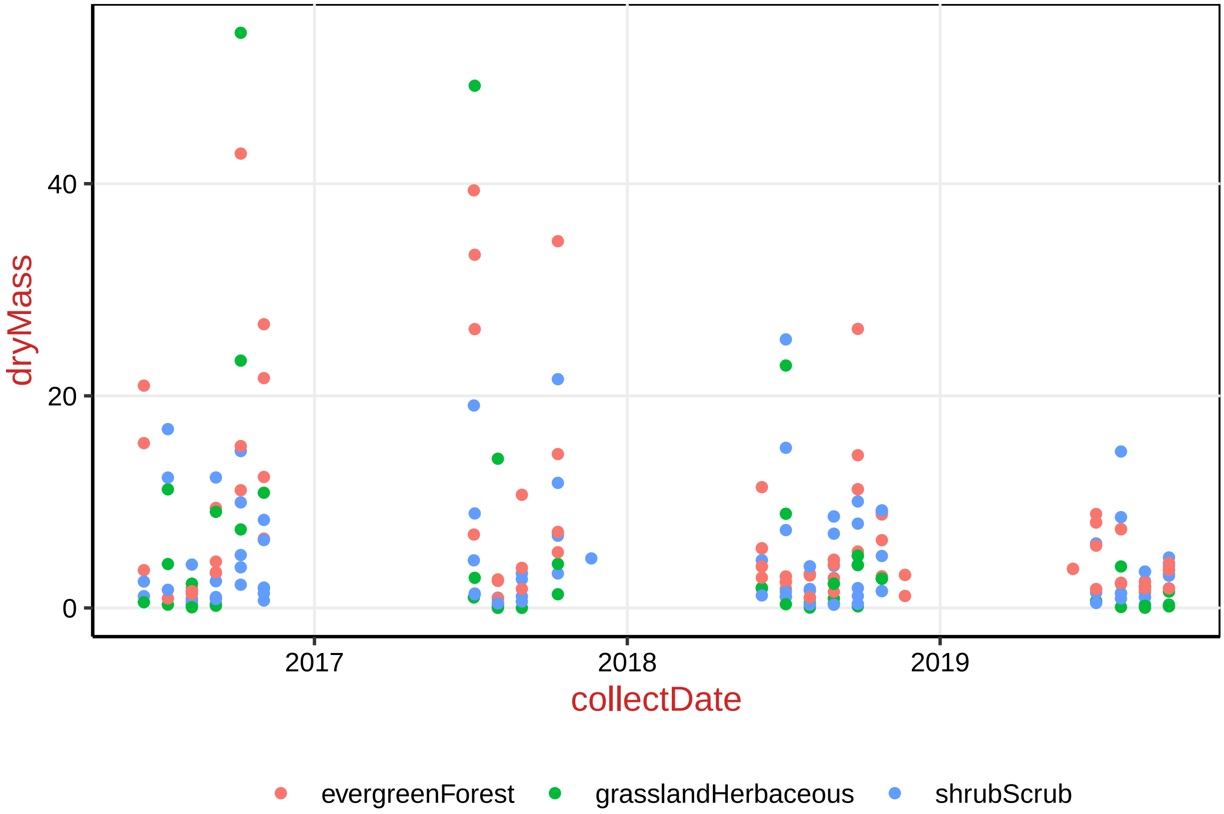
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: As expected, temperature has a clear cycle associated with the season getting warmer in both lakes in the summer and cooling off in the spring, fall and winter. These lakes are in close proximity and, thus, the temperature profile is the same between both lakes aside from some small disparity in median temperature. There is a clear increase in "tp_ug" accompanying the summer months, which is not present in "tn_ug". "tn_ug" appears to be affected by a variable not captured in this analysis, as it increases and decreases without clear correlation to either "tp_ug" or temperature. In general, Peter Lake seems to have higher concentrations of the nutrients in question than does Paul Lake.

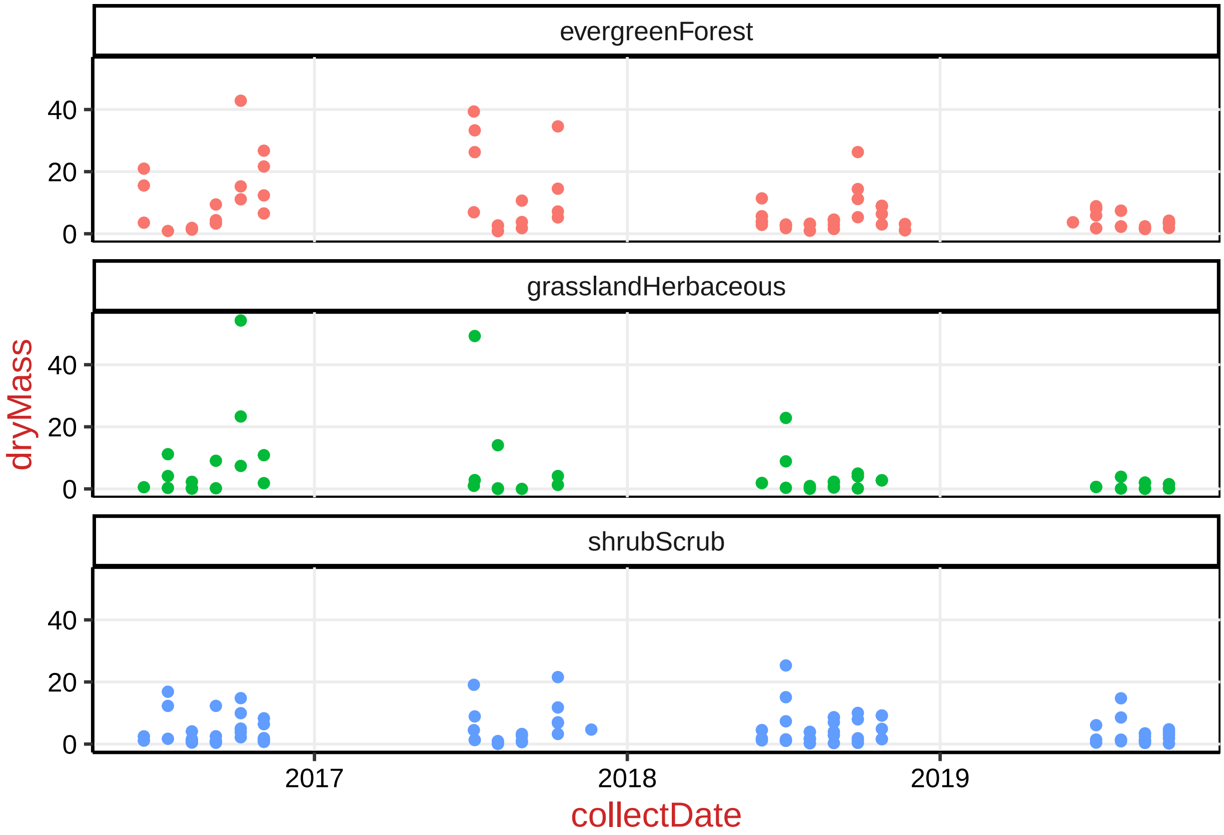
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
# 6 making a subset of Needles only and plotting it

#making a subset of the data in the functionalGroup for Needles
needles <-
  litter %>%
  subset(functionalGroup == "Needles") %>%
  droplevels()

# plotting the dry mass of litter by date and separating by NLCD class
needle_plot <- ggplot(needles) +
  geom_point(aes(y = dryMass, x = collectDate, color = nlcdClass)) +
  theme(legend.position = "bottom", legend.title = element_blank())
print(needle_plot)
```



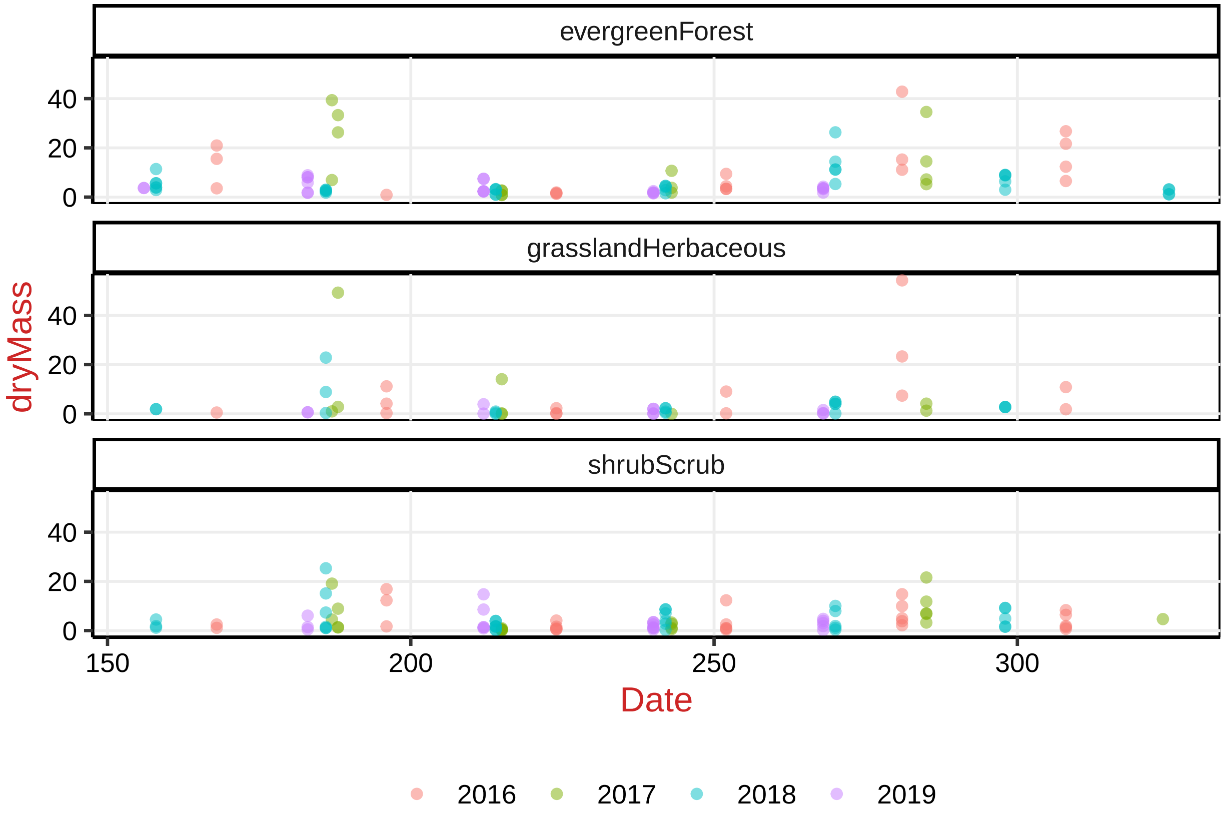
```
# 7 doing the same thing again but with facets
needle_facet_plot <- ggplot(needles) +
  geom_point(aes(y = dryMass, x = collectDate, color = nlcdClass)) +
  theme(legend.position = "none") +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(needle_facet_plot)
```



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

> Answer: Plot 7 is more effective than plot 6, as it makes the distinction between the variables easier and you can better track the trends within each nlcdClass. With all of them on the same graph, it makes it much more difficult to distinguish individual patterns and serves no purpose since we are not comparing between the different values for nlcdClass. I think the best way to view this data would be a faceted plot with the year as a color aesthetic. I have included what I am discussing below, I realize it was not part of the assignment but I did not know how else to explain the advantages I was discussing.

```
# This is me experimenting, no need to even look at this unless you are interested

# First, we need a date from the collect date
needles$Date = yday(needles$collectDate)
needles$Year = as.factor(year(needles$collectDate))

# now we graph it
needle_experiment <- ggplot(needles) +
  geom_point(aes(y = dryMass, x = Date, color = Year),  alpha = 0.5) +
  theme(legend.position = "bottom", legend.title = element_blank()) +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(needle_experiment)
```


evergreenForest
grasslandHerbaceous
shrubScrub
dryMass
0
20
40
Date
150
200
250
300
2016
2017
2018
2019