Assignment 5: Data Visualization

GuruBandaa Khalsa

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct 21st @ 5:00pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse, lubridate, & 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_PeterP`] version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processe`] version).

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

```
# 1. I checked my working directory using
# the getwd() function, loaded the
# `tidyverse`, `lubridate`, and cowplot
# packages, and uploaded the NTL-LTER
# processed data files for nutrients and
# chemistry/physics for Peter and Paul
# Lakes.
getwd()
```

```
## [1] "/Users/survivormangb/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analy
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2

## Warning: package 'ggplot2' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
NTL1 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
    stringsAsFactors = TRUE)
NTL2 <- read.csv("~/Desktop/Masters at Duke/Second Year/Fall Semester/Environmental Data Analytics/EDA-
    stringsAsFactors = TRUE)

# 2. I used the View() function to check
# that R is reading dates as date format.
View(NTL1)
```

```
View(NTL2)

NTL1$sampledate <- as.Date(NTL1$sampledate, format = "%Y-%m-%d")
NTL2$collectDate <- as.Date(NTL2$collectDate,
    format = "%Y-%m-%d")
```

Define your theme

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

```
# 3. I assigned a pre-built theme from
# Ggplot as a common theme across all plots
# in my analysis session.
mytheme <- theme_classic(base_size = 14) + theme(axis.text = element_text(color = "black"),
    legend.position = "top")
```

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/or `ylim()`).

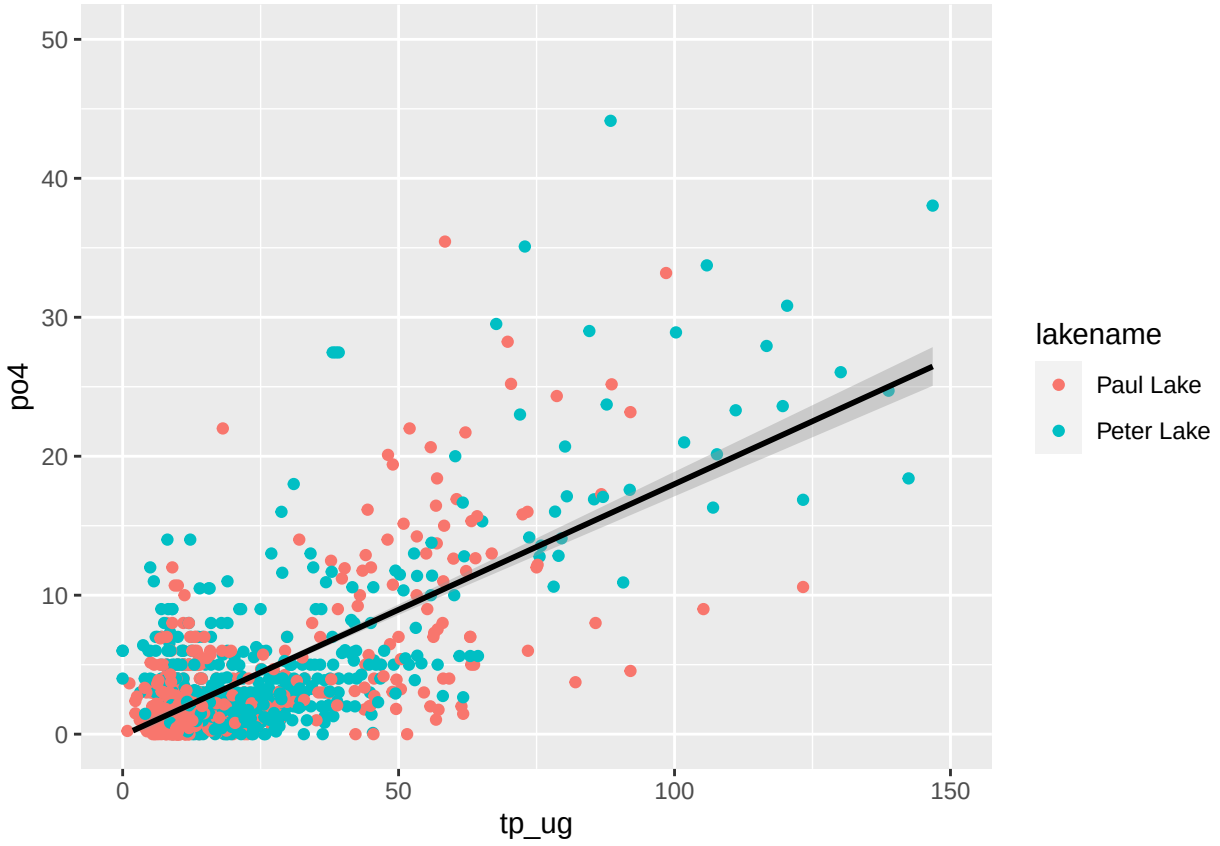
```
# 4. I created a ggplot graph with
# appropriate adjusted aesthetics for total
# phosphorus (`tp_ug`) by phosphate (`po4`),
# with separate aesthetics for Peter and
# Paul lakes.
phosphorus.phosphate.plot <- ggplot(NTL1, aes(x = tp_ug,
    y = po4)) + geom_point(aes(color = lakename)) +
    geom_smooth(method = lm, color = "black") +
    xlim(0, 150) + ylim(0, 50)
print(phosphorus.phosphate.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.

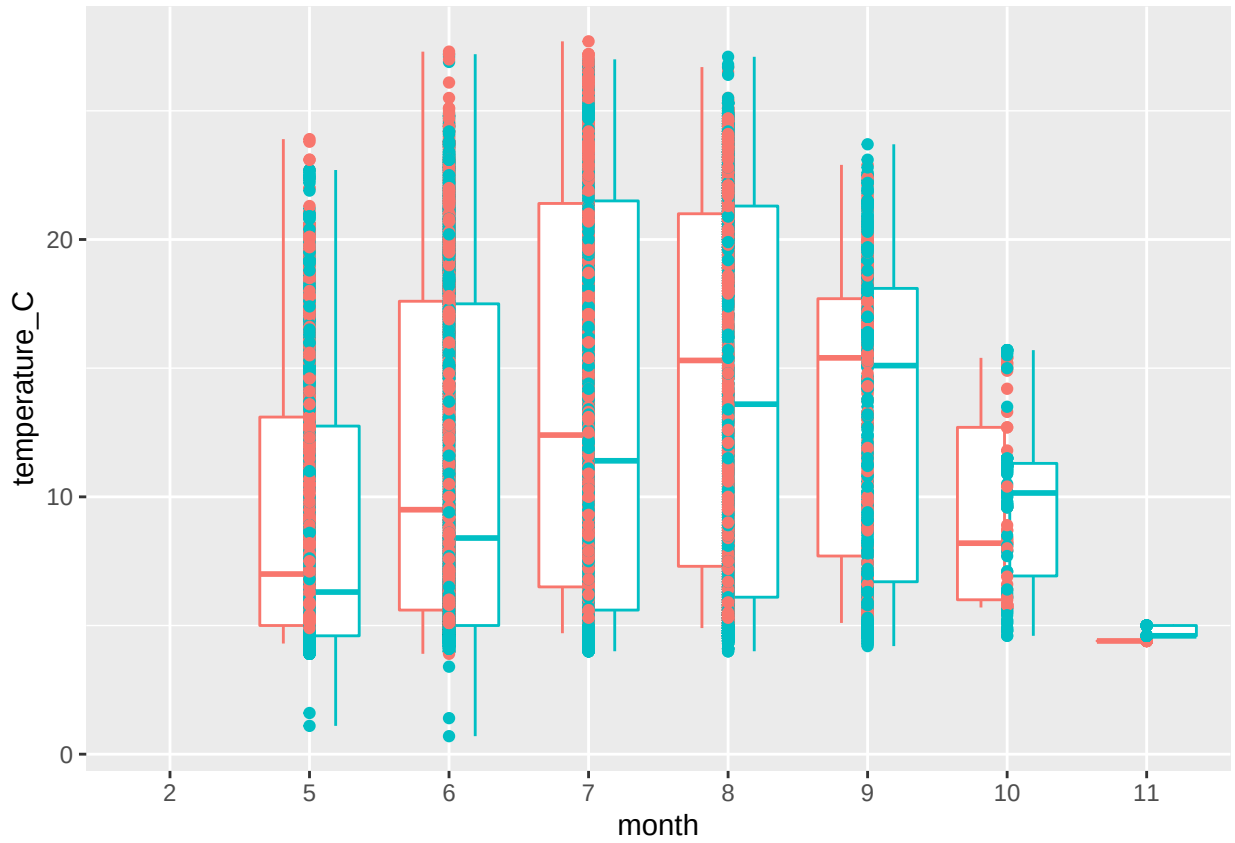
Tip: R has a build in variable called `month.abb` that returns a list of months; see https://r-lang.com/month-abb-in-r-with-example

```
# 5. I created three separate boxplots of
# (a) temperature, (b) TP, and (c) TN, with
# month as the x axis and lake as a color
# aesthetic.  I also created a cowplot that
# combines the three graphs.
NTL1$month <- as.factor(NTL1$month)

temp.boxplot <- ggplot(NTL1, aes(x = month, y = temperature_C,
    color = lakename)) + geom_boxplot() + geom_point() +
    theme(legend.position = "none")
print(temp.boxplot)
```

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

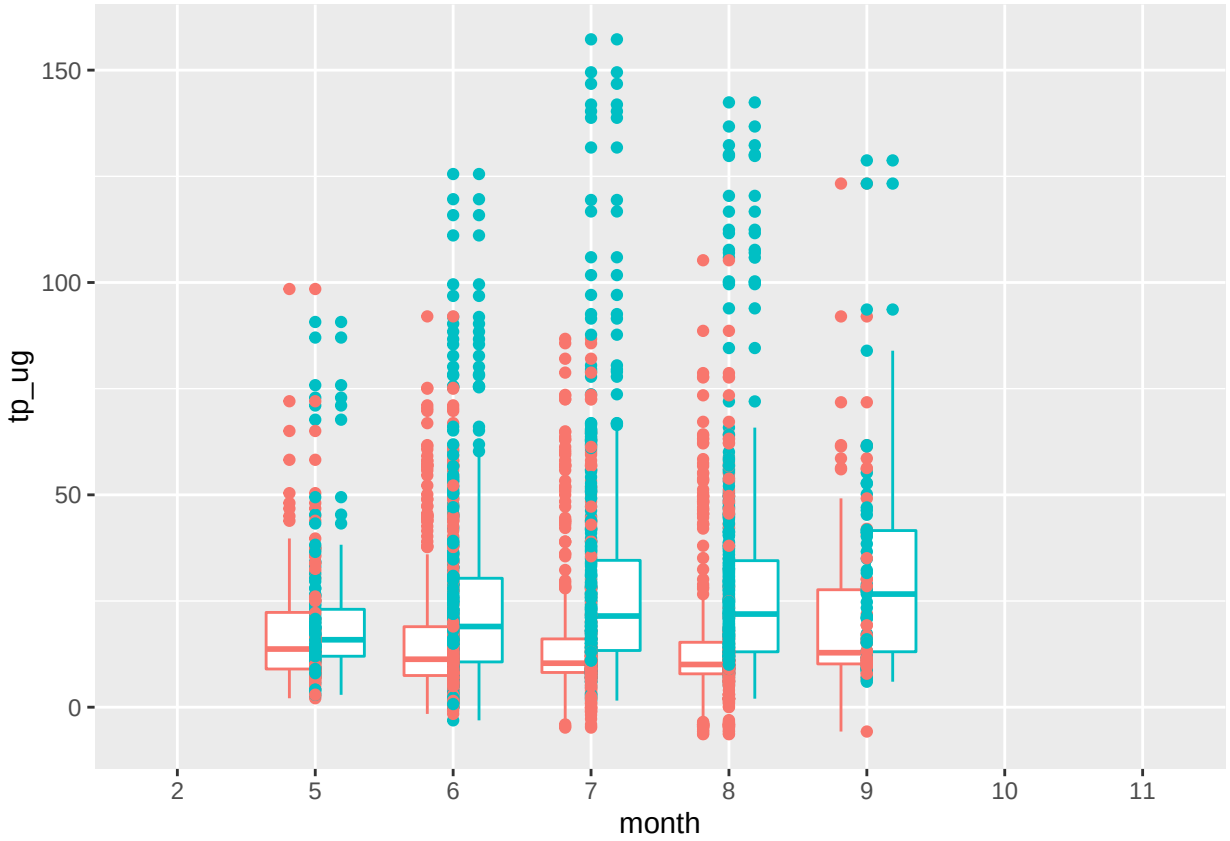
```
## Warning: Removed 3566 rows containing missing values (geom_point).
```




```
tp.boxplot <- ggplot(NTL1, aes(x = month, y = tp_ug,
    color = lakename)) + geom_boxplot() + geom_point() +
    theme(legend.position = "none")
print(tp.boxplot)
```

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

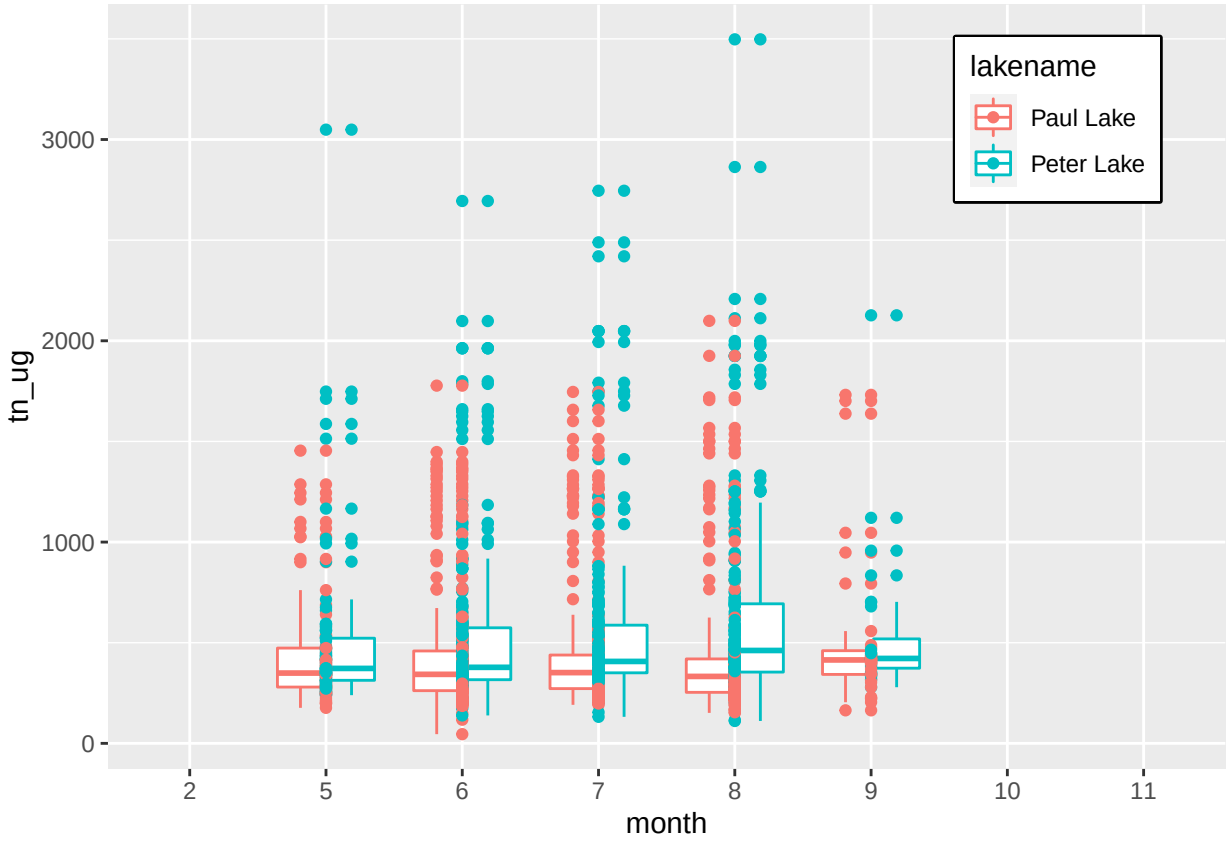
```
## Warning: Removed 20729 rows containing missing values (geom_point).
```




```
tn.boxplot <- ggplot(NTL1, aes(x = month, y = tn_ug,
    color = lakename)) + geom_boxplot() + geom_point() +
    theme(legend.position = c(0.85, 0.85), legend.box.background = element_rect(color = "black",
        size = 1))
print(tn.boxplot)
```

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

```
## Warning: Removed 21583 rows containing missing values (geom_point).
```




```
plot_grid(temp.boxplot, tp.boxplot, tn.boxplot,
    nrow = 3, align = "v", rel_heights = c(2,
        2, 2), rel_widths = c(3.5, 3.5, 3.5))
```

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

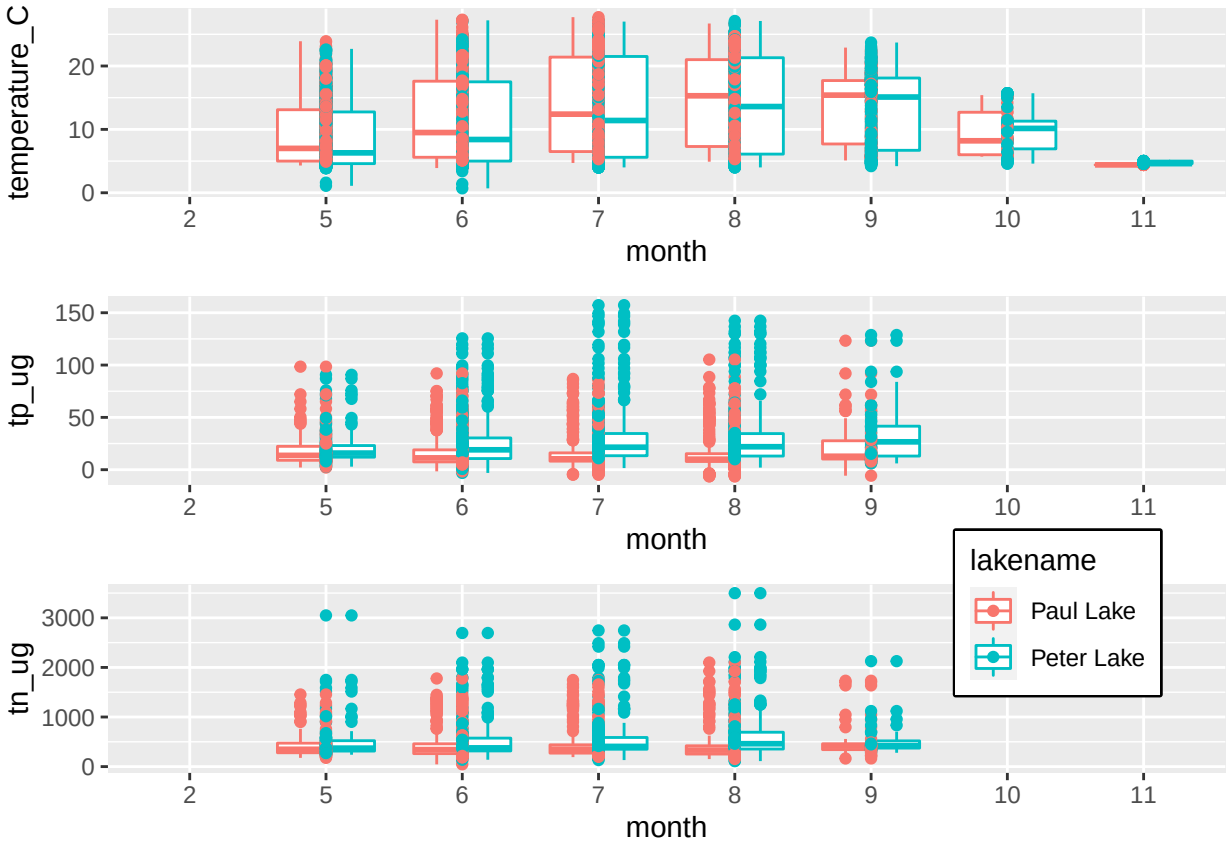
```
## Warning: Removed 3566 rows containing missing values (geom_point).
```

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

```
## Warning: Removed 20729 rows containing missing values (geom_point).
```

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

```
## Warning: Removed 21583 rows containing missing values (geom_point).
```



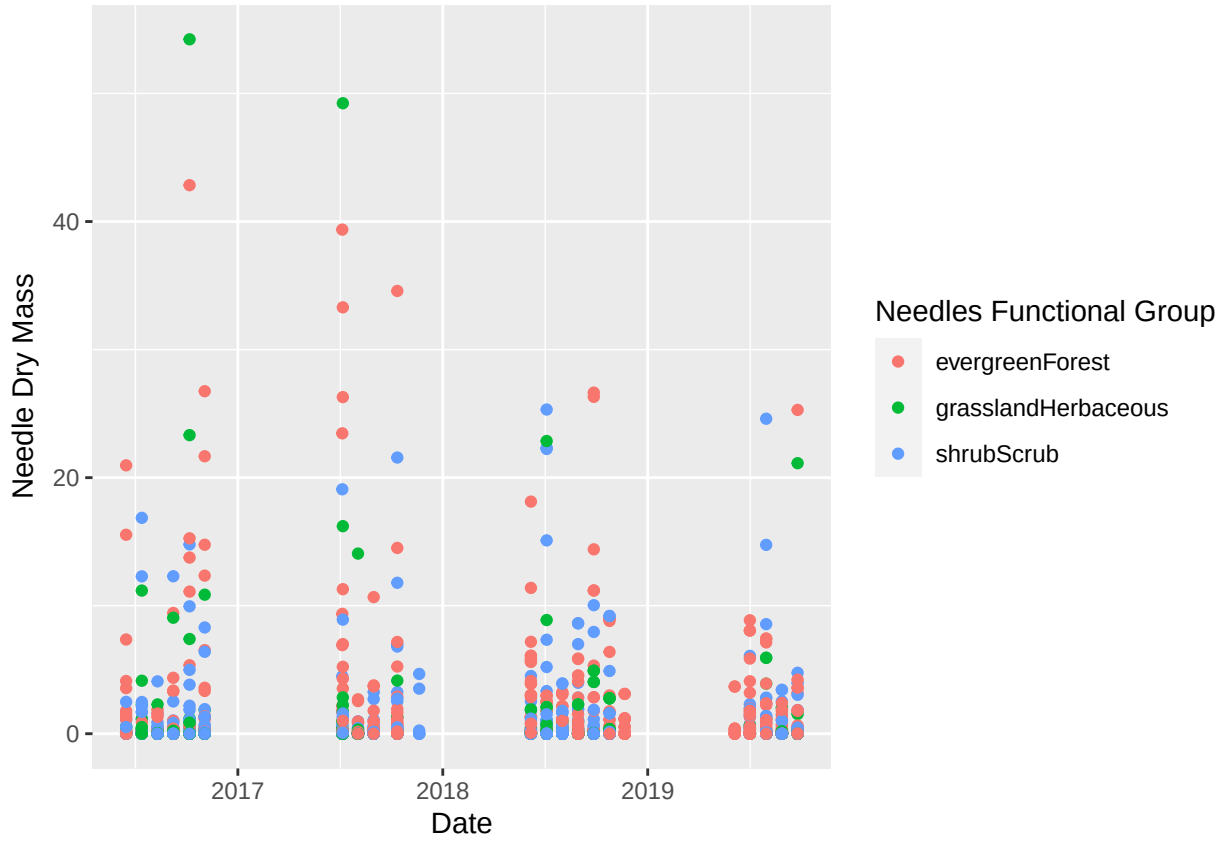
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature greatly increases from month 5 to month 9 and decreases drastically from month 9 to month 11 for both Paul Lake and Peter Lake. There are slightly lower temperatures present across all months in Peter Lake compared to Paul Lake. TP slightly decreases from month 5 to month 6 and slightly increases from month 8 to month 9 in Paul Lake. TP gradually increases from months 5 through 9 in Peter Lake. TN gradually increases over time fpr both Paul Lake and Peter Lake.

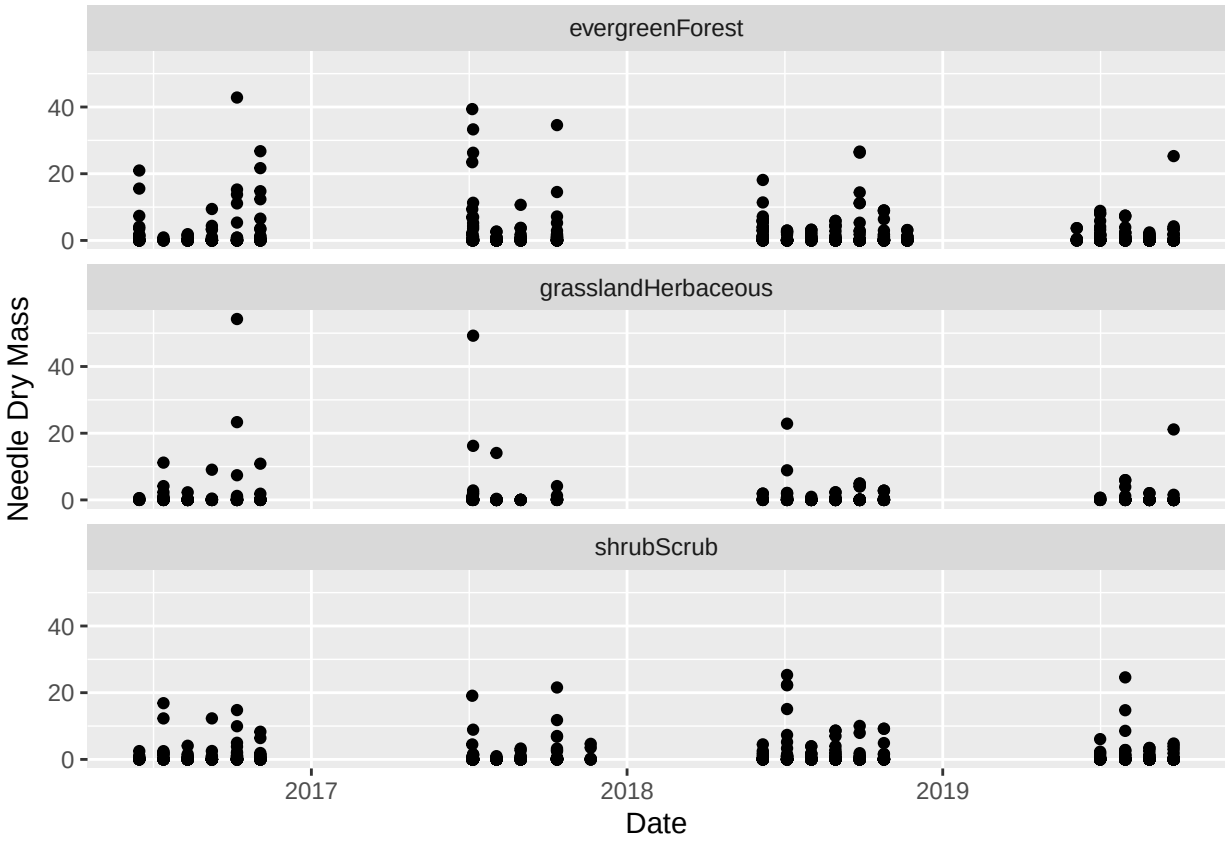
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
# 6. I plotted a subset of the litter
# dataset by displaying only the 'Needles'
# functional group.  I also plotted the dry
# mass of needle litter by date and
# separated by NLCD class with a color
# aesthetic.
needles.plot <- ggplot(NTL2, aes(x = collectDate,
    y = dryMass)) + geom_point(aes(color = nlcdClass)) +
    xlab(expression("Date")) + ylab(expression("Needle Dry Mass")) +
    labs(color = "Needles Functional Group")
print(needles.plot)
```




```
# 7. I created the same plot as number 6.
# but with NLCD classes separated into three
# facets rather than separated by color.
needles.plot <- ggplot(NTL2, aes(x = collectDate,
    y = dryMass)) + geom_point() + xlab(expression("Date")) +
    ylab(expression("Needle Dry Mass")) + labs(color = "Needles Functional Group") +
    facet_wrap(vars(nlcdClass), nrow = 3)
print(needles.plot)
```



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

Answer: The plot for 7. solves the issue of overlapping points that occurs in the plot for 6. The data for each functional group are easier to distinguish in the plot for 7.