Assignment 5: Data Visualization

Monisha Eadala

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., "Salk_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 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 (tidy and gathered) 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
getwd()

## [1] "/Users/monishaeadala/Environmental_Data_Analytics_2020"

library(tidyverse)

## ── Attaching packages
─────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ dplyr   0.8.3
## ✓ tidyr   1.0.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0

## ── Conflicts ──────────────────────────────────────────────────────────
tidyverse_conflicts() ──
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(cowplot)

##
## ************************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ************************************************************

PeterPaul.chem.nutrients <-
  read.csv("./Data/Processed/NTL-
LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
PeterPaul.chem.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-
LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
NIWO.Litter <-
  read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
PeterPaul.chem.nutrients$sampledate <- as.Date(
  PeterPaul.chem.nutrients$sampledate, format = "%Y-%m-%d")
PeterPaul.chem.nutrients.gathered$sampledate <- as.Date(
  PeterPaul.chem.nutrients.gathered$sampledate, format = "%Y-%m-%d")
NIWO.Litter$collectDate <- as.Date(
  NIWO.Litter$collectDate, format = "%Y-%m-%d")

class(PeterPaul.chem.nutrients$sampledate) # Just to check if it is now a
date

## [1] "Date"

class(PeterPaul.chem.nutrients.gathered$sampledate) # Just to check if it is
now a date

## [1] "Date"

class(NIWO.Litter$collectDate) # Just to check if it is now a date

## [1] "Date"
```

Define your theme

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

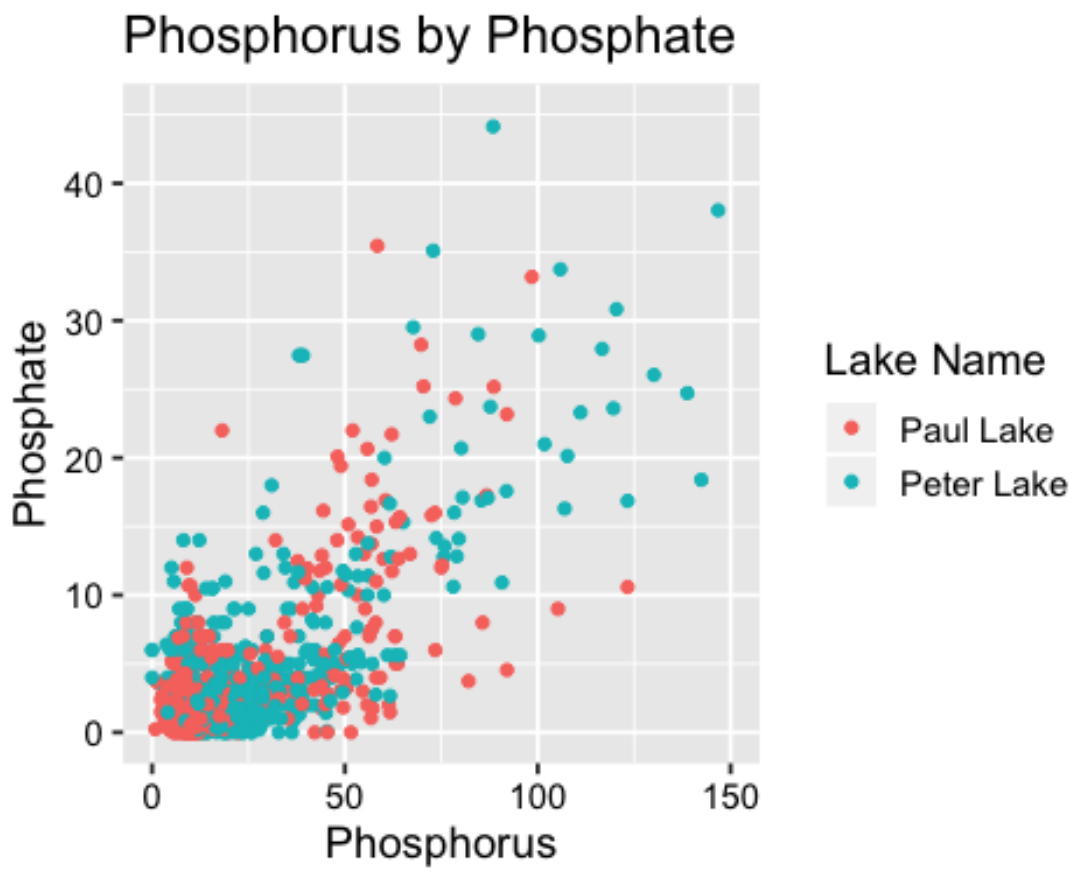
```
mytheme <- theme_gray(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
```

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.

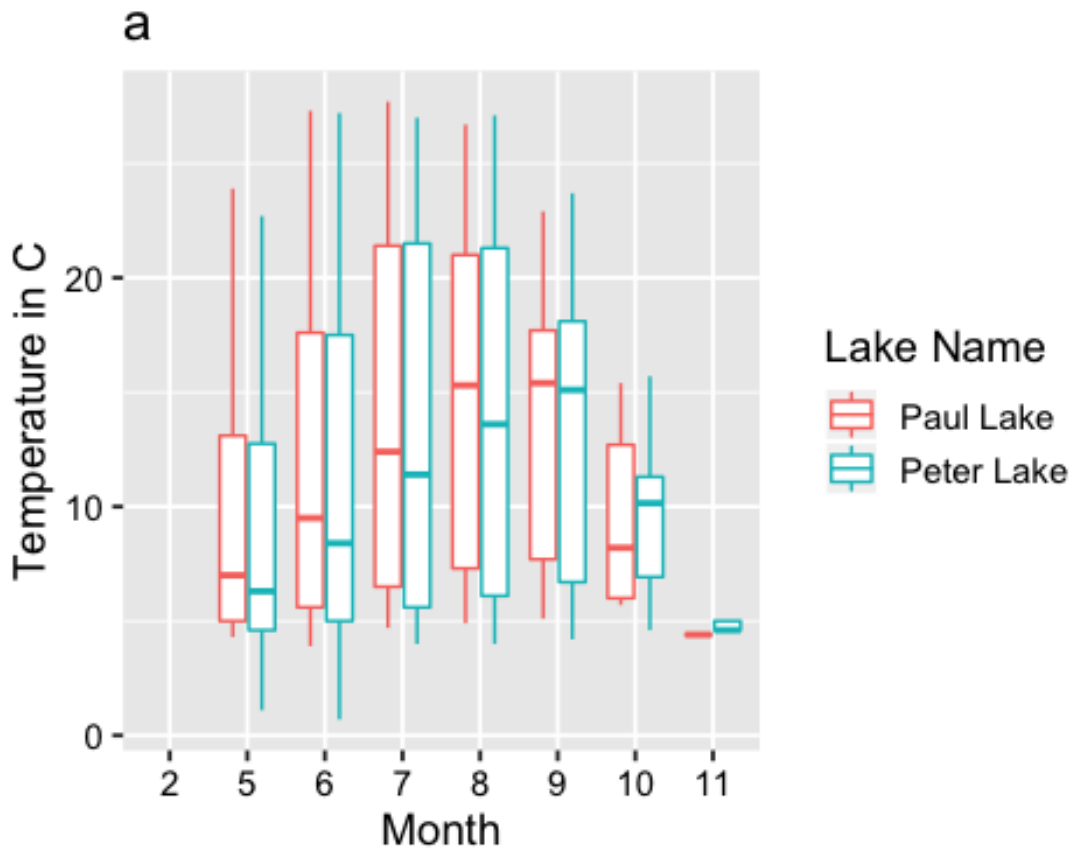
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values.

```
plot1 <- ggplot(PeterPaul.chem.nutrients) +
  geom_point(aes(x = tp_ug, y = po4, color = lakename )) + # To plot
phosphorus by phosphate with different color assigned to Peter and Paul lakes
  mytheme +
  ylim(0, 45) + xlim(0,150) + # To adjust the axes to hide extreme values
  ggtitle("Phosphorus by Phosphate") + # To insert the main title
  xlab("Phosphorus") + # To label the x axis
  ylab("Phosphate") + # To label the y axis
  labs (color = "Lake Name") # To label the legend
print(plot1) # Gives the final plot

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

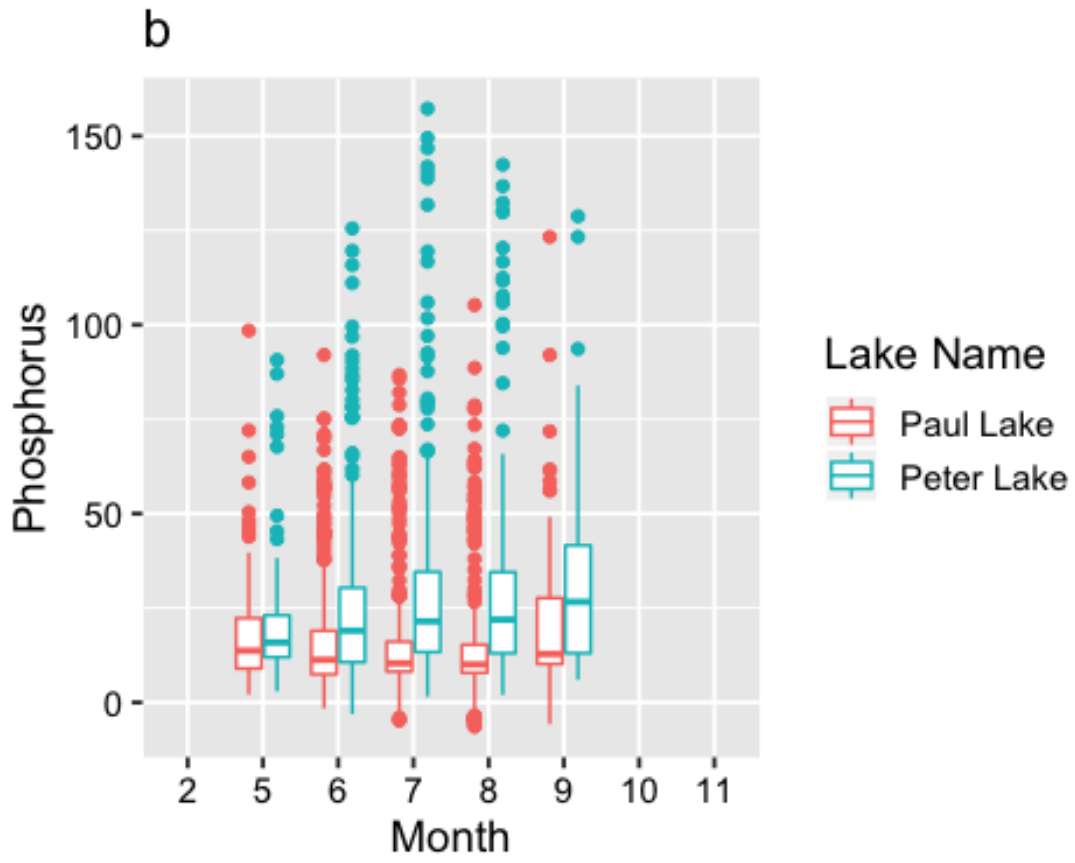


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

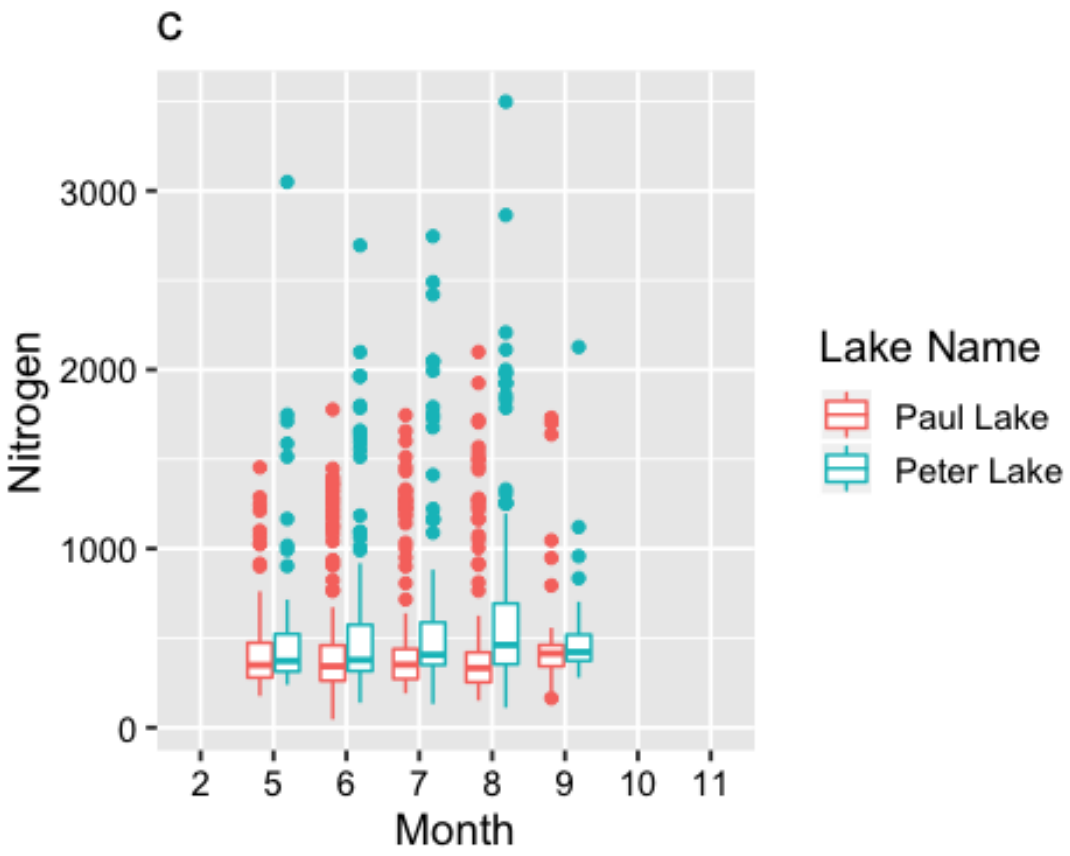
```
#a
plot2 <- ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month) , y =
temperature_C)) +
  geom_boxplot(aes(color = lakename)) + # makes boxplots of temperature along
the months for both the lakes
  mytheme +
  ggtitle("a") + # To insert the main title
  xlab("Month") + # To label the x axis
  ylab("Temperature in C") + # To label the y axis
  labs(color = "Lake Name") # To label the legend
print(plot2)

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



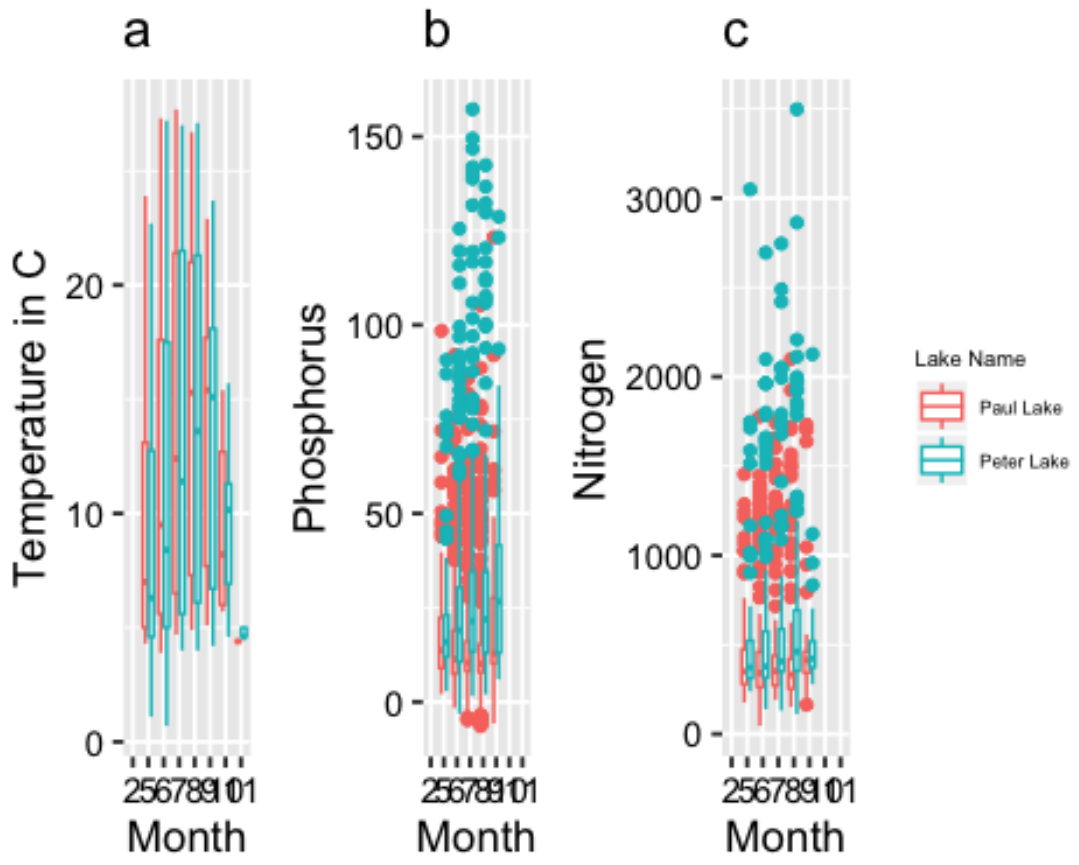

```
#b
plot3 <- ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month), y =
tp_ug)) +
  geom_boxplot(aes(color = lakename)) + # makes boxplots of phosphorus
content along the months for both the lakes
  mytheme +
  ggtitle("b") + # To insert the main title
  xlab("Month") + # To label the x axis
  ylab("Phosphorus") + # To label the y axis
  labs (color = "Lake Name") # To label the legend
print(plot3)

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




```
#c
plot4 <- ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month), y =
tn_ug)) +
  geom_boxplot(aes(color = lakename)) + # makes boxplots of nitrogen content
along the months for both the lakes
  mytheme +
  ggtitle("c") + # To insert the main title
  xlab("Month") + # To label the x axis
  ylab("Nitrogen") + # To label the y axis
  labs (color = "Lake Name") # To label the legend
print(plot4)

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




```
#cowpplot
library(cowplot)
plot2 = plot2 + theme(legend.position="none") # Removes the legend from plot2
plot3 = plot3 + theme(legend.position="none") # Removes the legend from plot2
plot4 = plot4 + theme(legend.position ="right", legend.title =
element_text(size = 7),
  legend.text = element_text(size = 6)) # Asserts the position of the legend
on the right side of the plot
plot_grid(plot2, plot3, plot4, nrow = 1, align = 'h', rel_widths = c(1, 1,
2)) # Combines all the three plots

## 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).
```



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

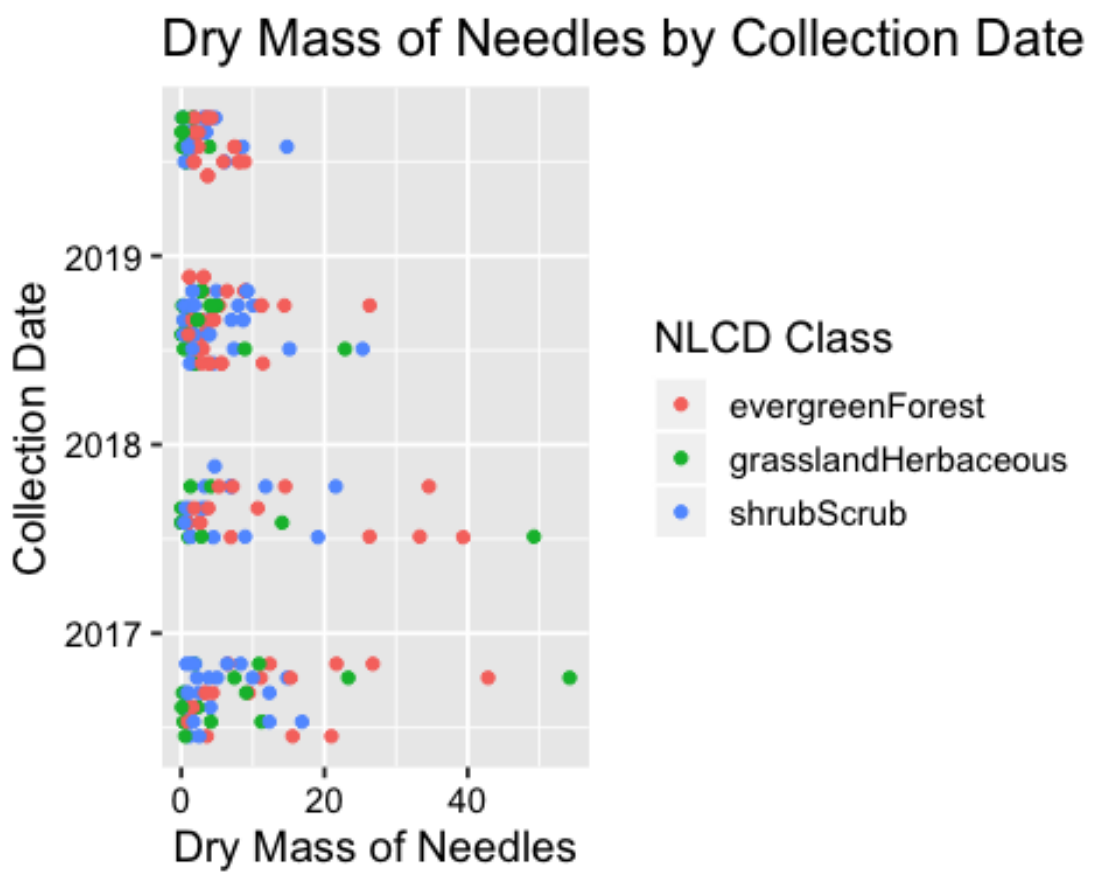
Answer: Over seasons, it seems like the median temperature of Peter Lake starts off (in May) as lower than that of Paul lake, but it eventually becomes higher than that of Paul Lake by the month of October through November. On the other hand, the median of the phosphorus in Peter lake over the seasons seems to be consistently higher than those of Paul lake, and the difference between the two increases marginally over the months. The median of the nitrogen is also higher in the Peter lake over the seasons, but the difference between the nitrogen contents between the two lakes increases consistently till August, but decreases in September (even though Peter Lake is still higher).

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.

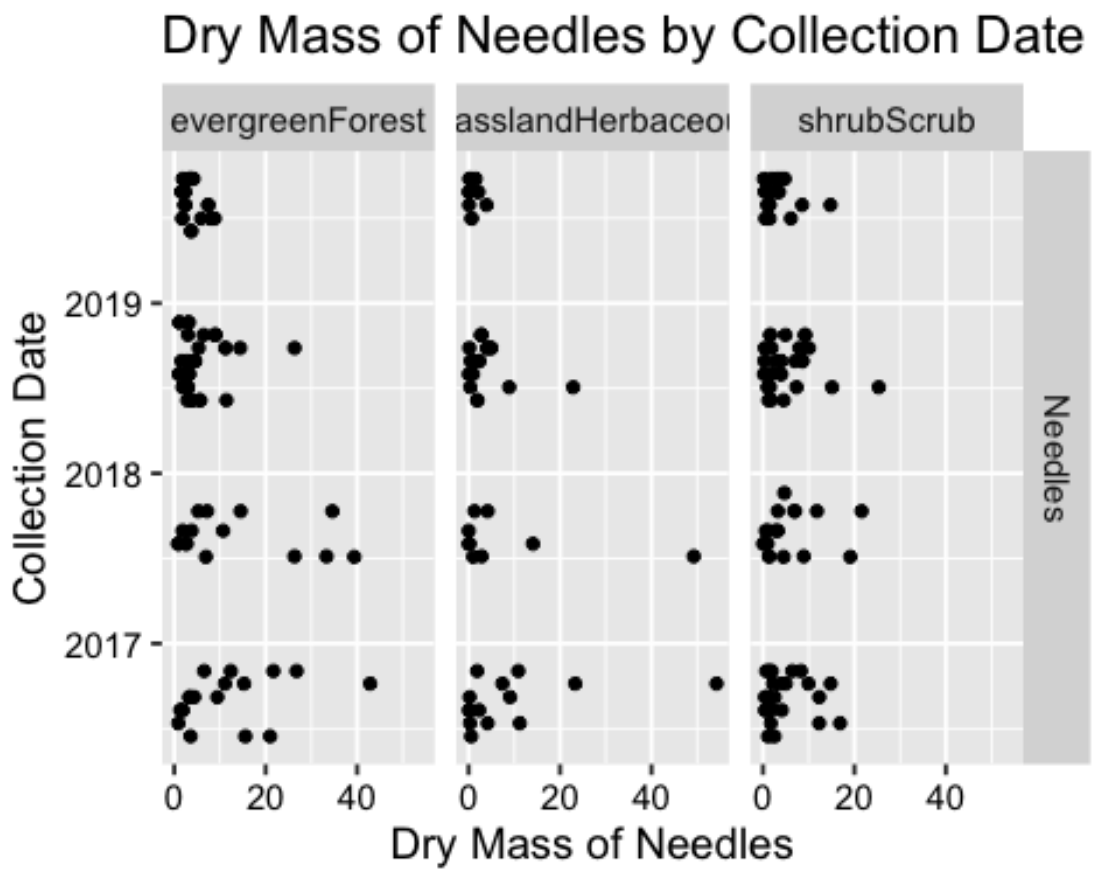
```
#6
plot6.color <-
  ggplot(subset(NIWO.Litter, functionalGroup == "Needles"), # Plots a subset
of the litter dataset by displaying only the "Needles" functional group
         aes(dryMass, collectDate, color = nlcdClass)) + # Plots the dry
mass of needle litter by date and separates by NLCD class with a color
```

```
aesthetic
  geom_point() +
  mytheme +
  ggtitle("Dry Mass of Needles by Collection Date") + # To insert the main
title
  xlab("Dry Mass of Needles") + # To label the x axis
  ylab("Collection Date") + # To label the y axis
  labs (color = "NLCD Class") # To label the legend
print(plot6.color)
```




```
#7
plot7.facets <-
  ggplot(subset(NIWO.Litter, functionalGroup == "Needles"),
         aes(dryMass, collectDate)) +
  geom_point() +
  facet_grid(functionalGroup ~ nlcdClass) + # Plots the same plot but with
NLCD classes separated into three facets rather than separated by color
  mytheme +
  ggtitle("Dry Mass of Needles by Collection Date") + # To insert the main
title
  xlab("Dry Mass of Needles") + # To label the x axis
  ylab("Collection Date") + # To label the y axis
```

```
  labs (color = "NLCD Class") # To label the legend
print(plot7.facets)
```



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

Answer: I think plot6 is more effective since we can compare dry mass of the Needles found in different NLCD classes more effectively, since they are in the same frame with different colors. It is difficut to visualize and compare the same in plot7 when they are in three different frames.