Assignment 5: Data Visualization

Eva May

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
getwd()
```

```
## [1] "/Users/evamay/Desktop/ENV872/Environmental.Data.Analytics.2021/Assignments"
```

```
setwd("~/Desktop/ENV872/Environmental.Data.Analytics.2021")
library(tidyverse)
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

```
library(cowplot)
#2
P.P.Nutrients <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                          header=TRUE)
```

```
P.P.Nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv",
                                    header=TRUE)
litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                   header=TRUE)

class(P.P.Nutrients$sampledate)
```

```
## [1] "character"
```

```
P.P.Nutrients$sampledate <- as.Date(P.P.Nutrients$sampledate,
                                    format = "%Y-%m-%d")
class(P.P.Nutrients$sampledate)
```

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

```
class(P.P.Nutrients.gathered$sampledate)
```

```
## [1] "character"
```

```
P.P.Nutrients.gathered$sampledate <- as.Date(P.P.Nutrients.gathered$sampledate,
                                             format = "%Y-%m-%d")
class(P.P.Nutrients.gathered$sampledate)
```

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

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

```
## [1] "character"
```

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

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

Define your theme

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

```
#3
EM.theme <- theme_light(base_size = 15, base_family = "Times") +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(EM.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.

```
#4
Lakes.plot1 <- ggplot(P.P.Nutrients, aes(x = po4, y = tp_ug, color=lakename)) +
  geom_point() + ylim(0, 150) +
  xlim(0, 40) + xlab(expression(paste("Phosphate, ", mu, "g"))) +
```

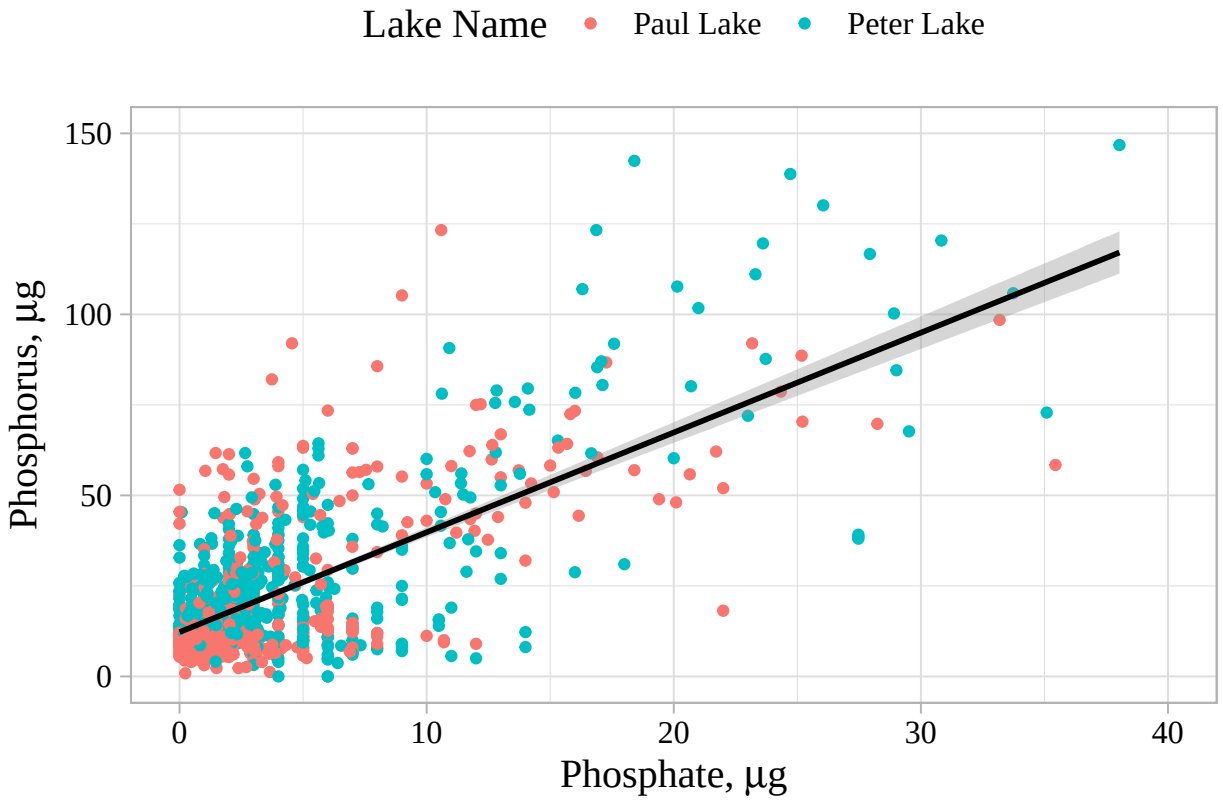
```
  ylab(expression(paste("Phosphorus, ", mu, "g"))) +
  geom_smooth(method = lm, color="black") + labs(color="Lake Name")

print(Lakes.plot1)
```

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

```
## Warning: Removed 21949 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 21949 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.

```
class(P.P.Nutrients$month)
```

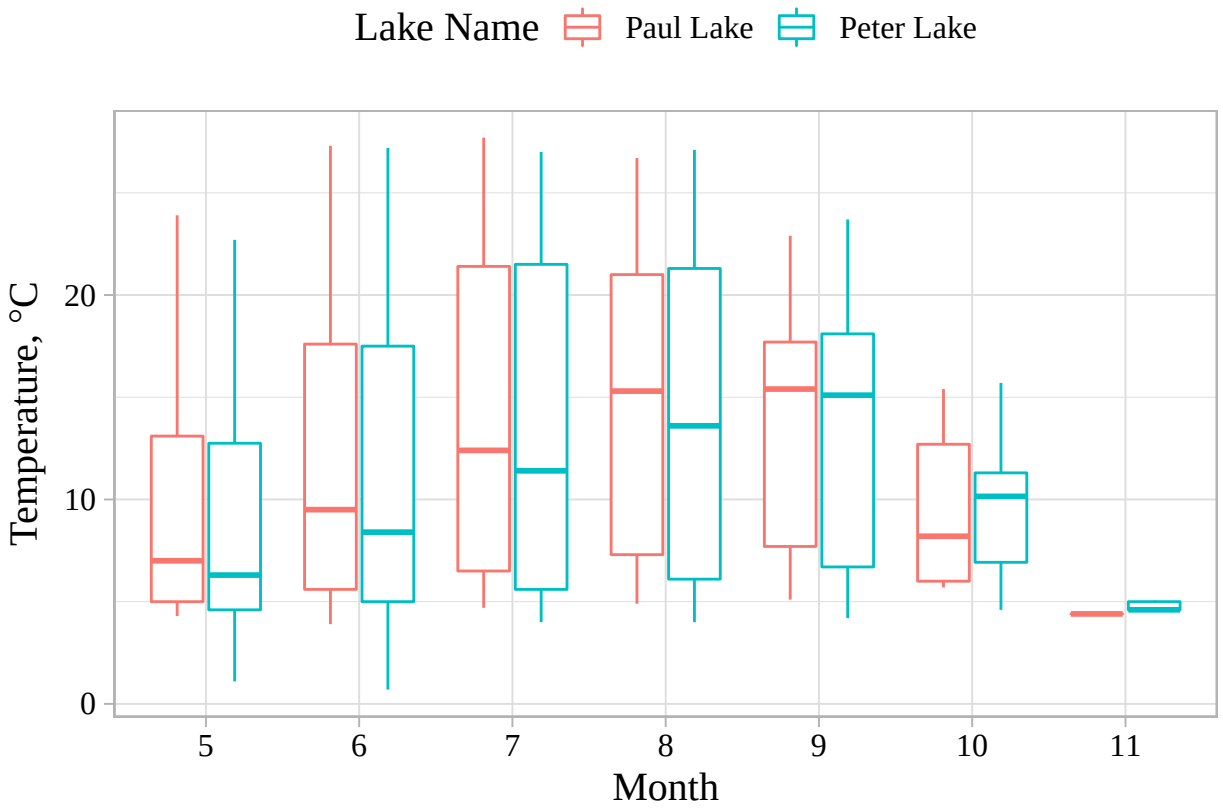
```
## [1] "integer"
```

```
P.P.Nutrients$month <- as.factor(P.P.Nutrients$month)

tempplot <- ggplot(P.P.Nutrients,
                   aes(x=month, y=temperature_C, color=lakename)) +
  geom_boxplot() + xlab("Month") +
  ylab(expression(paste("Temperature, ", degree, "C"))) +
  labs(color="Lake Name") +
  scale_x_discrete(limits=factor("5":"11"))
print(tempplot)
```

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

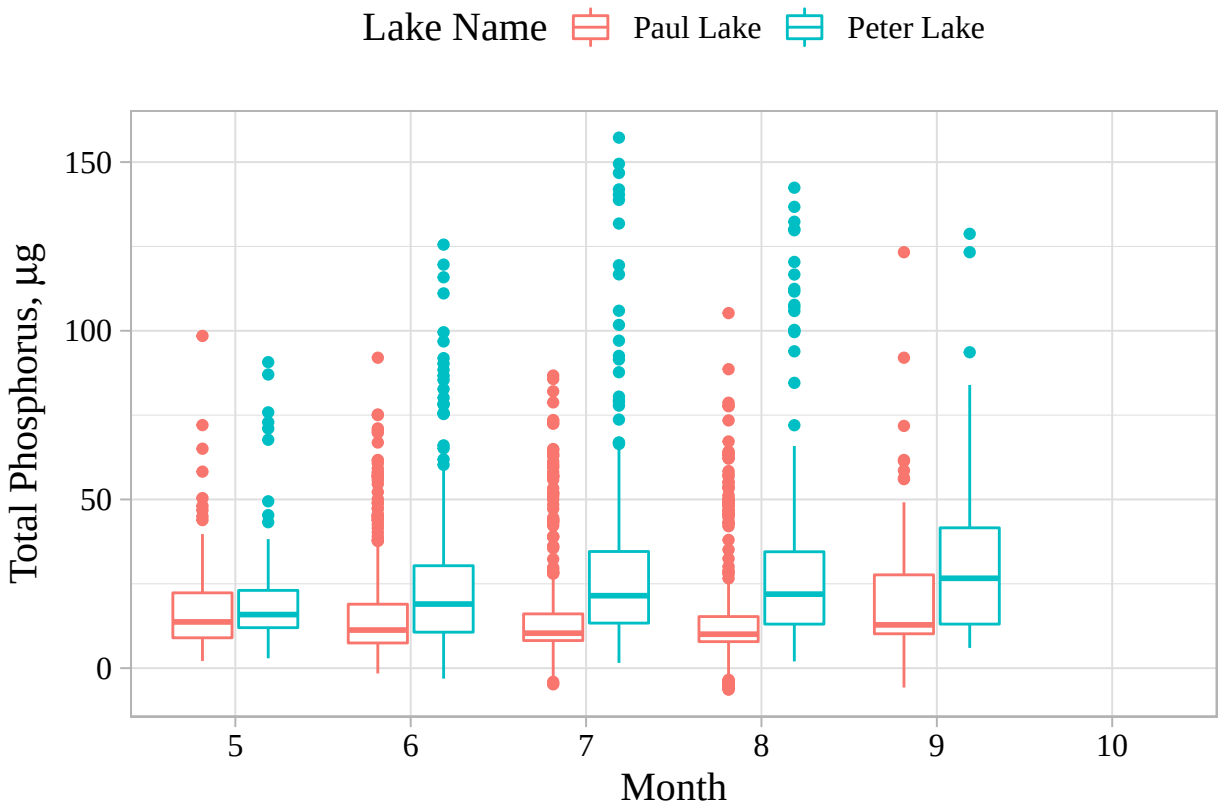
```
## Warning: Removed 3550 rows containing non-finite values (stat_boxplot).
```




```
TPplot <- ggplot(P.P.Nutrients,
                 aes(x=month, y=tp_ug, color=lakename)) +
  geom_boxplot() + xlab("Month") +
  ylab(expression(paste("Total Phosphorus, ", mu, "g"))) +
  labs(color="Lake Name") +
  scale_x_discrete(limits=factor("5":"10"))
print(TPplot)
```

```
## Warning: Removed 72 rows containing missing values (stat_boxplot).
```

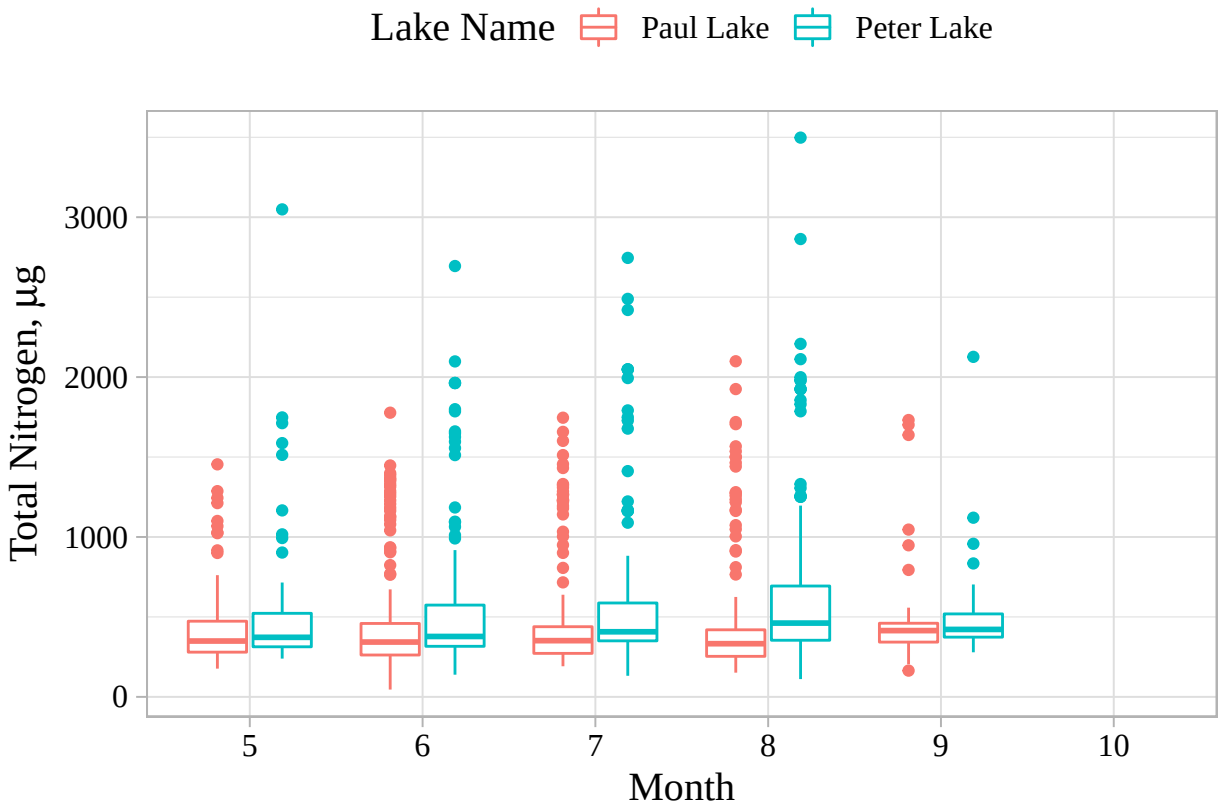
```
## Warning: Removed 20657 rows containing non-finite values (stat_boxplot).
```




```
TNplot <- ggplot(P.P.Nutrients, aes(x=month, y=tn_ug, color=lakename)) +
  geom_boxplot() +
  xlab("Month") +
  ylab(expression(paste("Total Nitrogen, ", mu, "g"))) +
  labs(color="Lake Name") +
  scale_x_discrete(limits=factor("5":"10"))
print(TNplot)
```

```
## Warning: Removed 72 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 21511 rows containing non-finite values (stat_boxplot).
```




```
legend <- get_legend(TNplot + theme(legend.position="bottom"))
```

```
## Warning: Removed 72 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 21511 rows containing non-finite values (stat_boxplot).
```

```
combo <- plot_grid(tempplot + theme(legend.position = "none"),
                   TPplot + theme(legend.position = "none"),
                   TNplot + theme(legend.position = "none"), nrow=1)
```

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

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

```
## Warning: Removed 72 rows containing missing values (stat_boxplot).
```

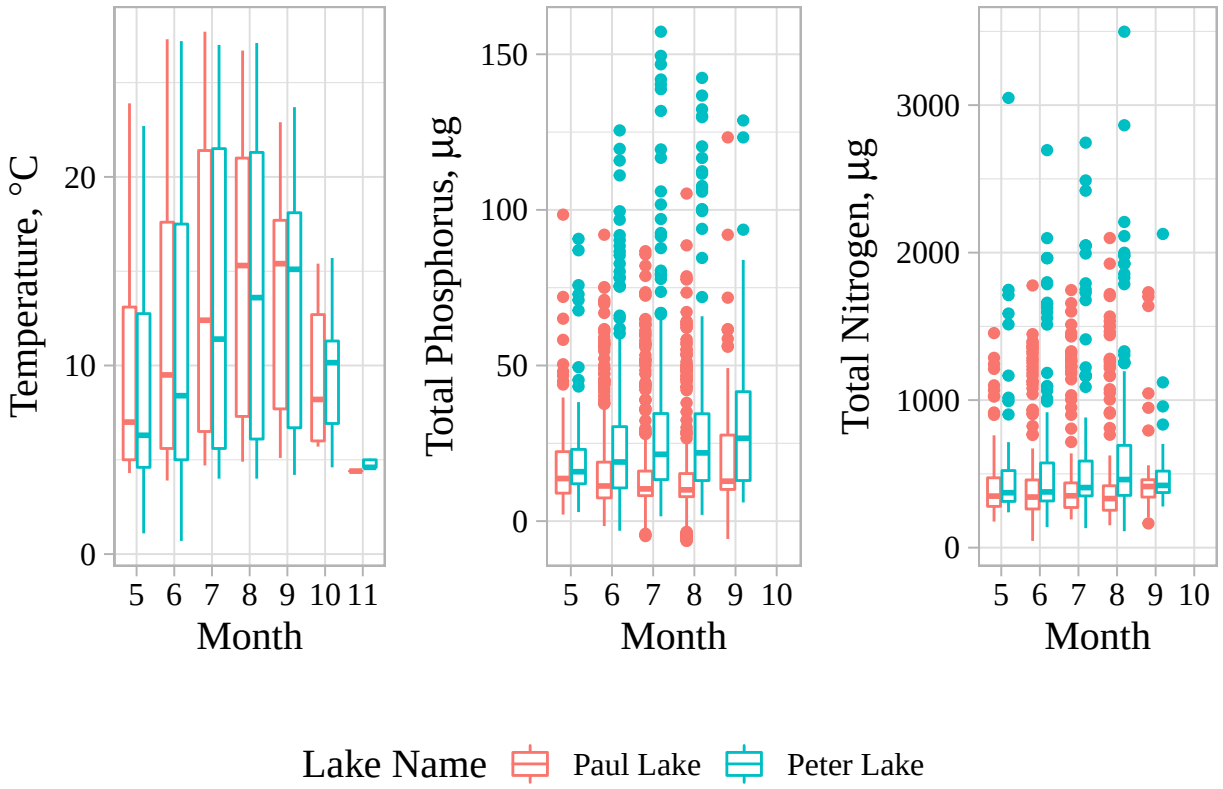
```
## Warning: Removed 20657 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 72 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 21511 rows containing non-finite values (stat_boxplot).
```

```
bottom <-  plot_grid(legend)

plot_grid(combo, bottom, nrow=2,
          rel_heights=(c(2,0.5)), rel_widths = (c(2,0.5)))
```



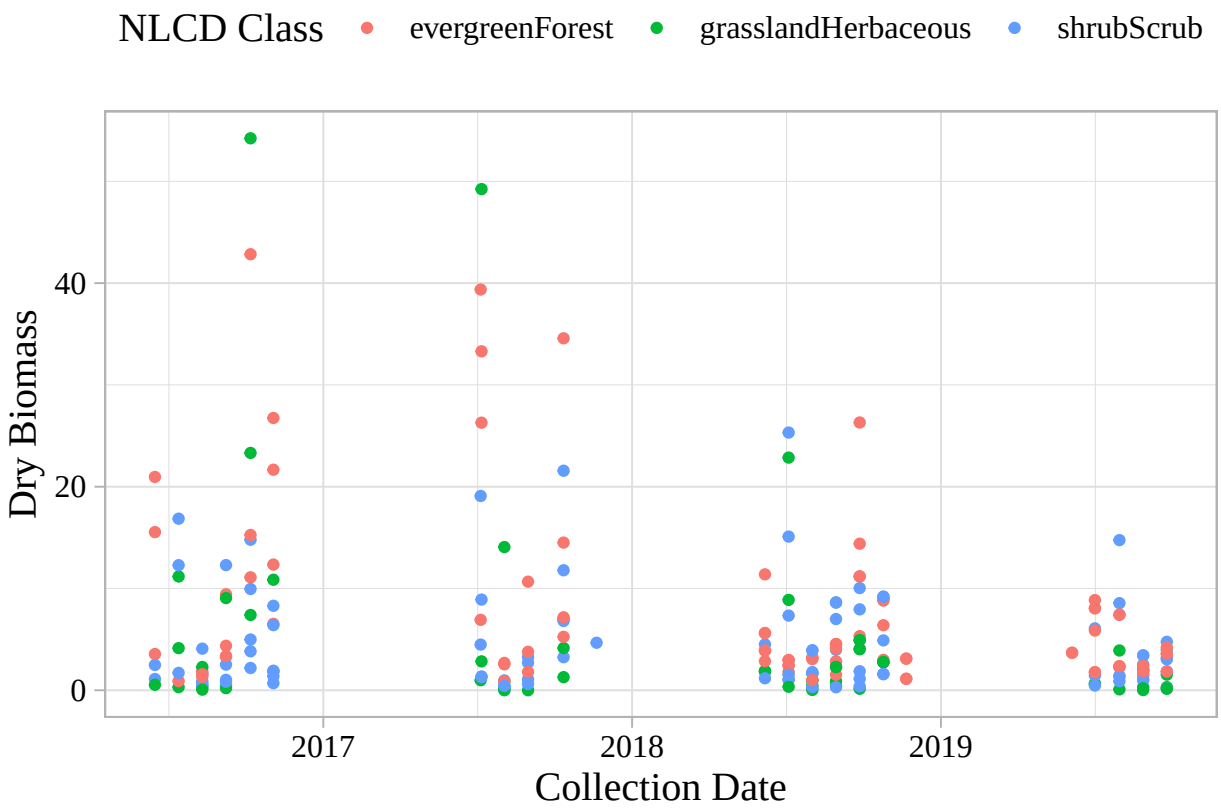
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Both lakes have higher temperatures during the summer months, though their average temperature is highest during September rather than a month that is truly during the summer. Generally, Paul Lake is warmer than Peter Lake. The amount of nitrogen in both lakes is much higher than phosphorus, and Peter Lake seems to have on average more of both nitrogen and phosphorus than Paul Lake. There is not much variation in total phosphorus throughout the year in either lake, though Peter Lake has slightly higher phosphorus in late summer/early fall than earlier in the years. Paul Lake's mean phosphorus levels are fairly consistent across months. Paul Lake's total nitrogen levels also lack variation across different months. Peter Lake's nitrogen levels are slightly higher at the end of the summer than earlier and later in the year.

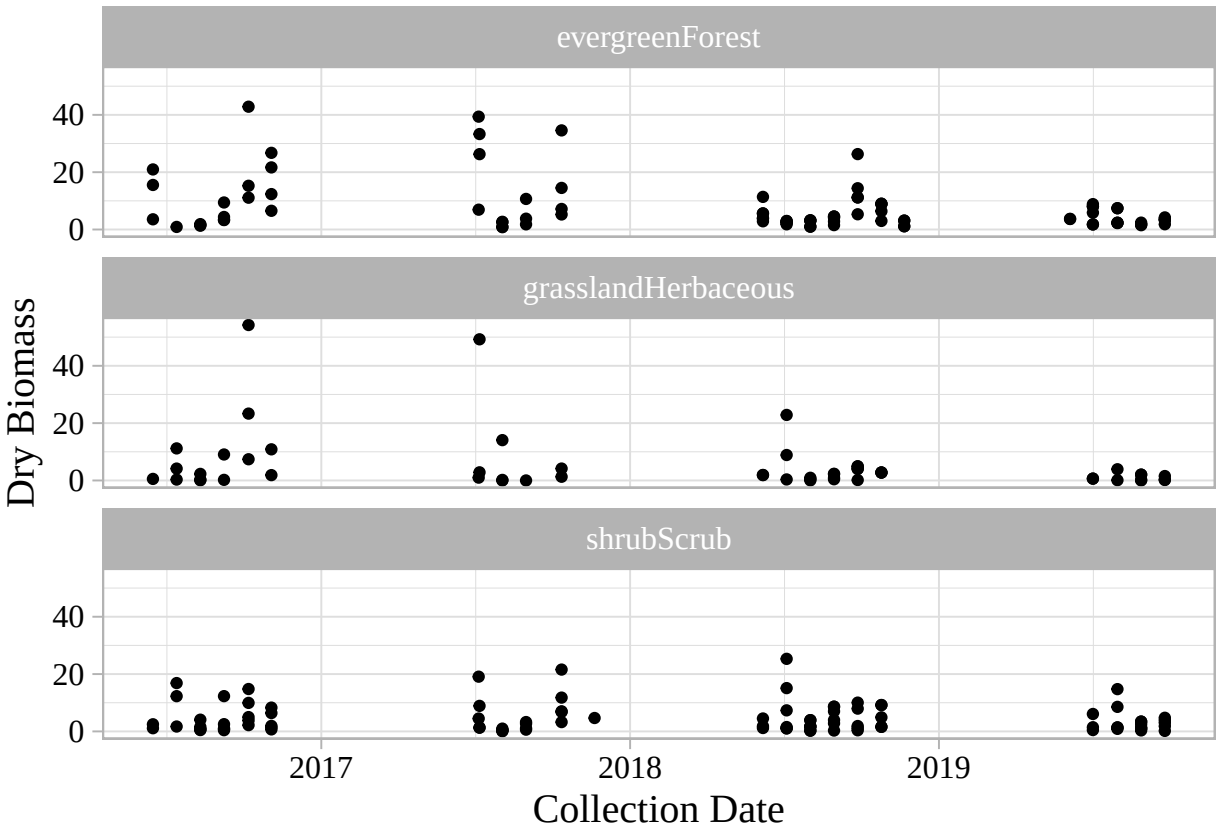
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
litterplot1 <-
  ggplot(subset(litter, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass, color=nlcdClass)) +
  geom_point() + xlab("Collection Date") +
  ylab("Dry Biomass") + labs(color = "NLCD Class")
print(litterplot1)
```




```
#7
litterplot2 <-
  ggplot(subset(litter, functionalGroup == "Needles"),
         aes(x = collectDate, y = dryMass)) +
  geom_point() +
   facet_wrap(vars(nlcdClass), nrow = 3) +
  xlab("Collection Date") + ylab("Dry Biomass")
print(litterplot2)
```



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

> Answer: The second, faceted plot seems to be more effective to me. While the first plot looks nice because it has colors, it is difficult to get a good idea of data differences between NLCD classes because so many of the points overlap - the plot ends up looking busy. Versus in the second plot, you can make a direct vertical comparison of drymass amounts in different years for each NLCD class - separating the different classes makes it easier to see subtle differences in the data points.