Assignment 5: Data Visualization

Dori Rathmell

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.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 14th @ 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_PeterPau`
version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed`
version).

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

```
# 1
getwd()
```

```
## [1] "/home/guest/R/EDA Fall/Assignments"
```

```
library(tidyverse)
```

```
## -- 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   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- 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
```

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

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

```
Litter_Mass_data <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
    stringsAsFactors = TRUE)
Chem_nutrients <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
    stringsAsFactors = TRUE)
# opening and naming datasets 2
class(Litter_Mass_data$collectdate)
```

```
## [1] "NULL"
```

```
class(Chem_nutrients$sampledate)
```

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

```
# checking the initial class of date columns in both datasets
Litter_Mass_data$collectDate <- as.Date(Litter_Mass_data$collectDate, format = "%Y-%m-%d")
Chem_nutrients$sampledate <- as.Date(Chem_nutrients$sampledate, format = "%Y-%m-%d")
# reclassifying them as 'date' classes
class(Litter_Mass_data$collectDate)
```

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

```
class(Chem_nutrients$sampledate)
```

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

```
# double checking that they are now in the 'date' class
```

Define your theme

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

```
# 3
plot_theme <- theme_gray(base_size = 12) + theme(axis.text = element_text(color = "Gray"),
    legend.position = "right")
theme_set(plot_theme)
# setting my default theme to the gray theme
```

Create graphs

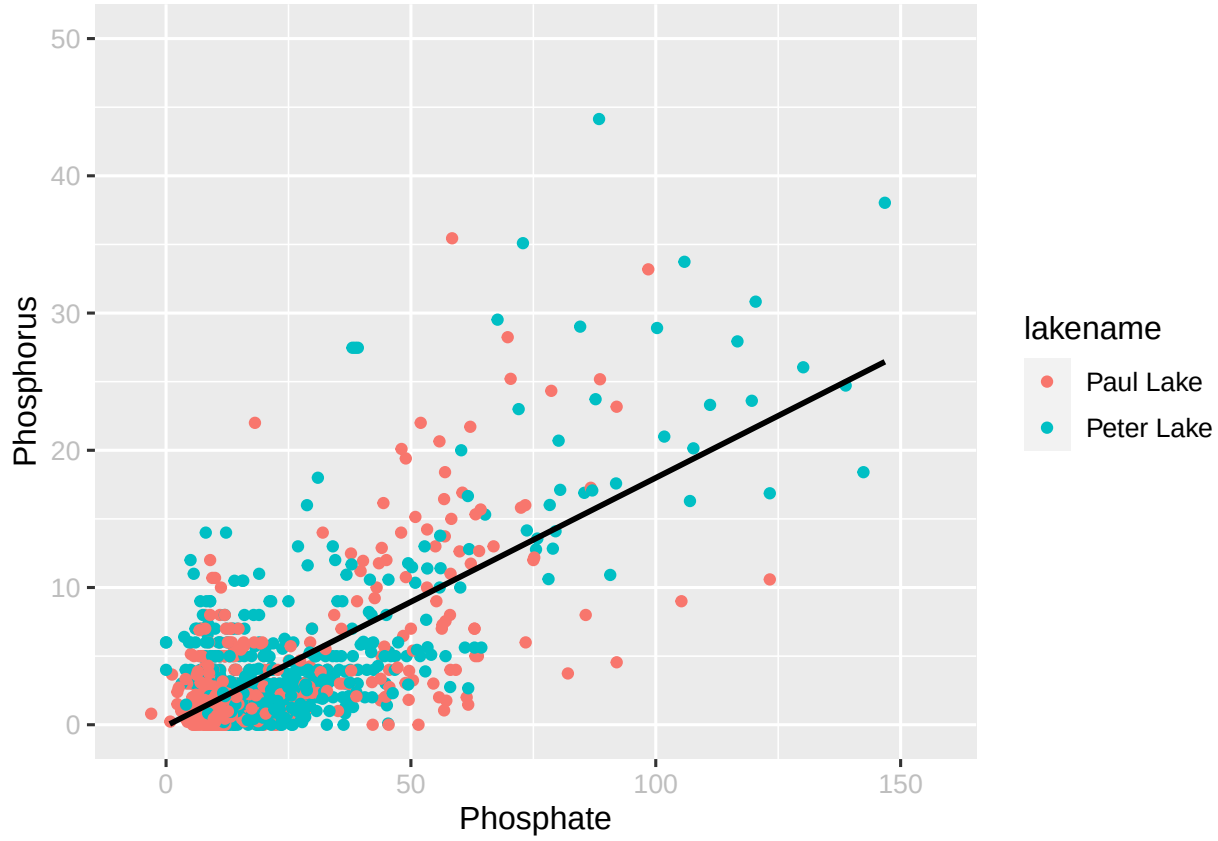
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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 (hint: change the limits using `xlim()` and/or `ylim()`).

```
# 4
plot1 <- ggplot(Chem_nutrients, aes(y = po4, x = tp_ug)) + geom_point(aes(color = lakename)) +
```

```
    ylim(0, 50) + plot_theme + geom_smooth(method = lm, se = FALSE, color = "black") +
    xlab("Phosphate") + ylab("Phosphorus")
# making my first scatter plot, setting the colors and theme + line of best fit
print(plot1)
```

```
## `geom_smooth()` using formula 'y ~ x'
## Warning: Removed 21947 rows containing non-finite values (stat_smooth).
## Warning: Removed 21947 rows containing missing values (geom_point).
## Warning: Removed 2 rows containing missing values (geom_smooth).
```




```
# displaying scatter plot
```

5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

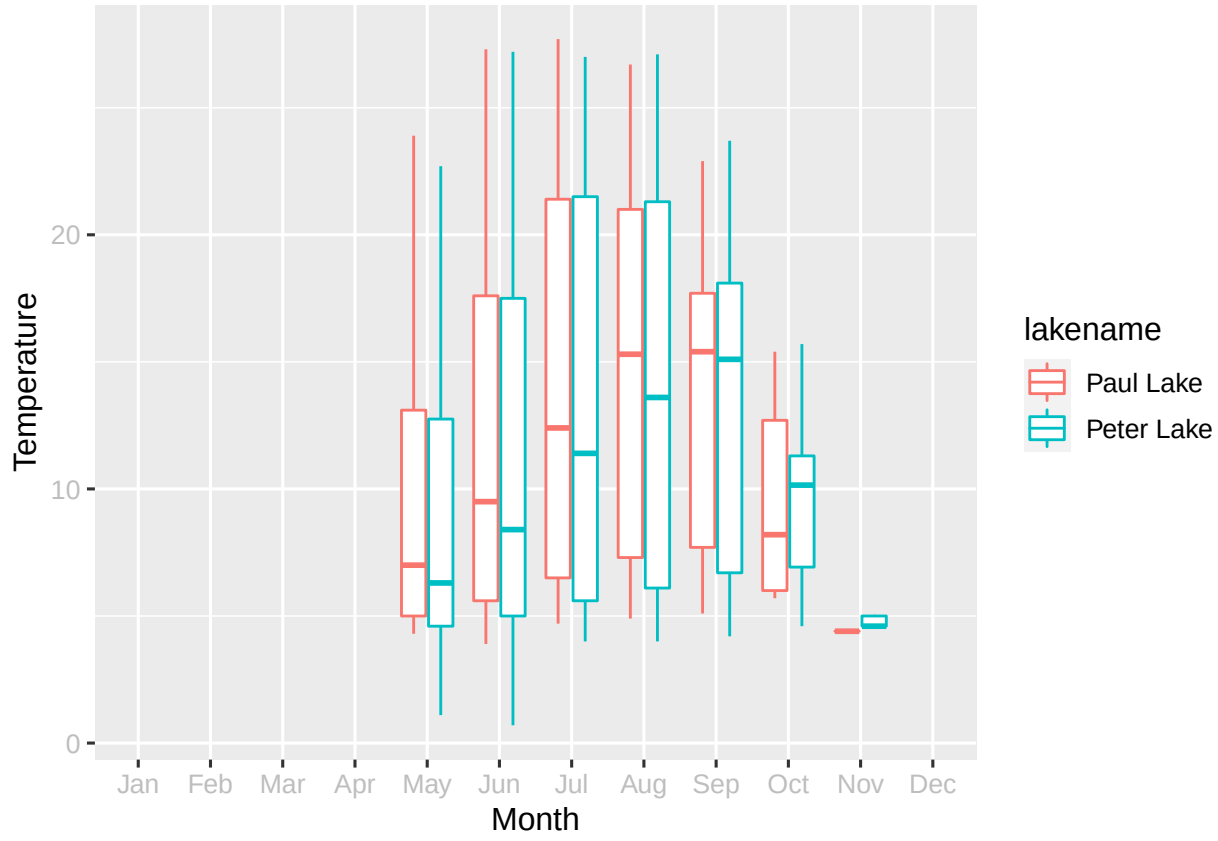
(c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
#a
boxplot1 <-
  ggplot(Chem_nutrients, aes(x = factor(month, levels=c(1:12)), y = temperature_C)) +
  geom_boxplot(aes(color = lakename))+
  plot_theme +
  ylab('Temperature')+
```

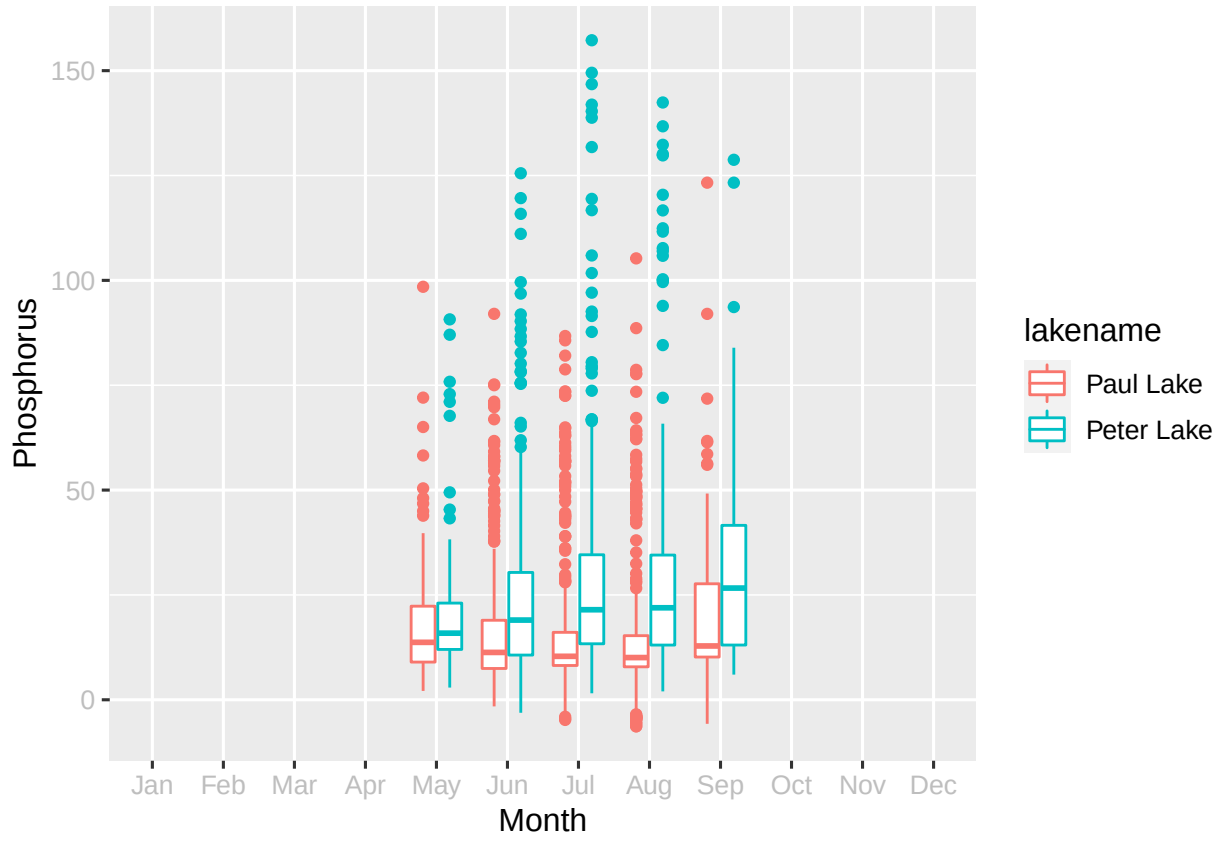
```
  xlab('Month')+
  scale_x_discrete(labels=month.abb[],drop=F) #converting months from numbers to names
#making the first boxplot
print(boxplot1)
```

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




```
#displaying the first boxplot
#b
boxplot2 <-
  ggplot(Chem_nutrients, aes(x = factor(month, levels=c(1:12)), y = tp_ug)) +
  geom_boxplot(aes(color = lakename))+
  plot_theme+
  ylab('Phosphorus')+
  xlab('Month')+
  scale_x_discrete(labels=month.abb[],drop=F)
#making the second box plot
print(boxplot2)
```

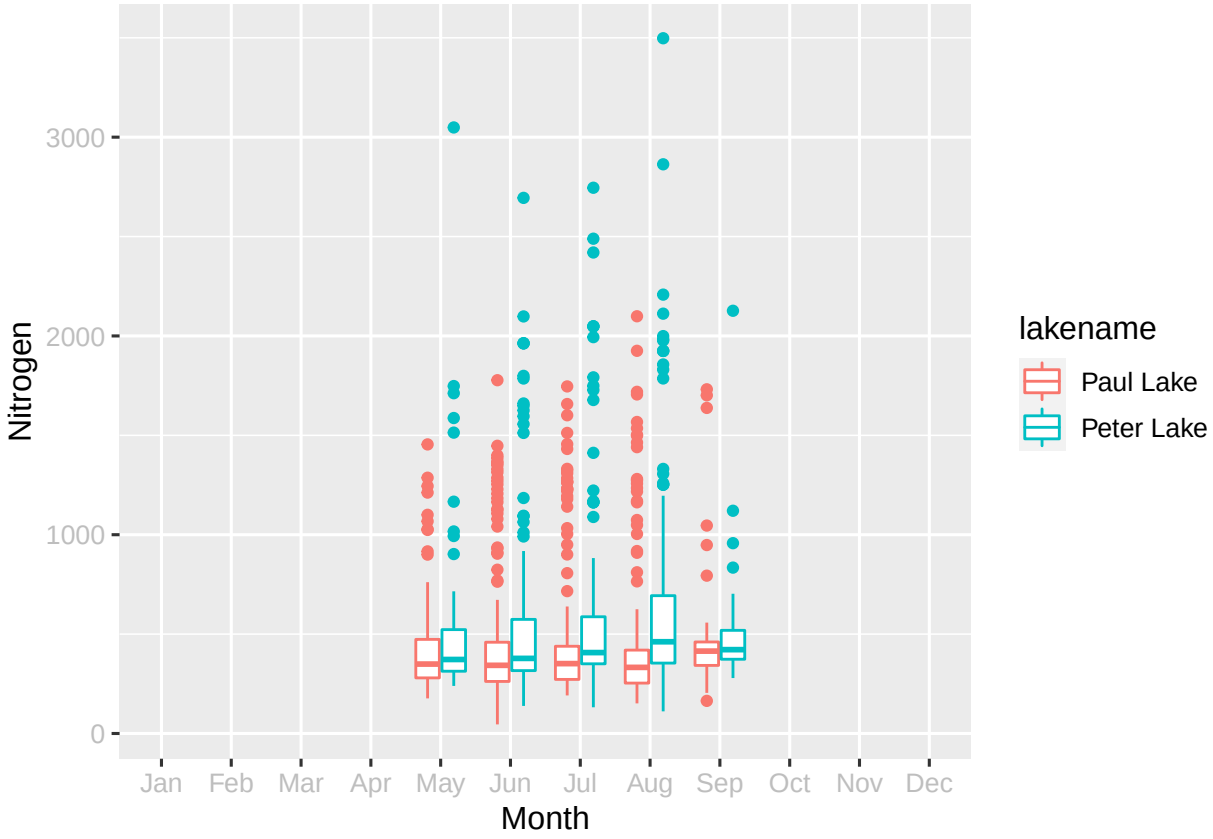
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```




```
#displaying the second box plot

#c
boxplot3 <-
  ggplot(Chem_nutrients, aes(x = factor(month, levels=c(1:12)), y = tn_ug)) +
  geom_boxplot(aes(color = lakename))+
  plot_theme+
  xlab('Month')+
  ylab('Nitrogen')+
  scale_x_discrete(labels=month.abb[],drop=F)
#making the third box plot
print(boxplot3)
```

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




```
#displaying the third box plot

#d

library(cowplot)
cowplot1 <- plot_grid(
  boxplot1 + theme(legend.position="none", axis.title.x = element_blank()),
  boxplot2 + theme(legend.position="none", axis.title.x = element_blank()),
  boxplot3 + theme(legend.position='bottom'), #making only one legend present
  align = 'vh',
  ncol = 1,nrow = 3, rel_heights = c(1.5, 1.5, 2)#setting the heights and rows
  #combining all the box plots into one cow plot
)
```

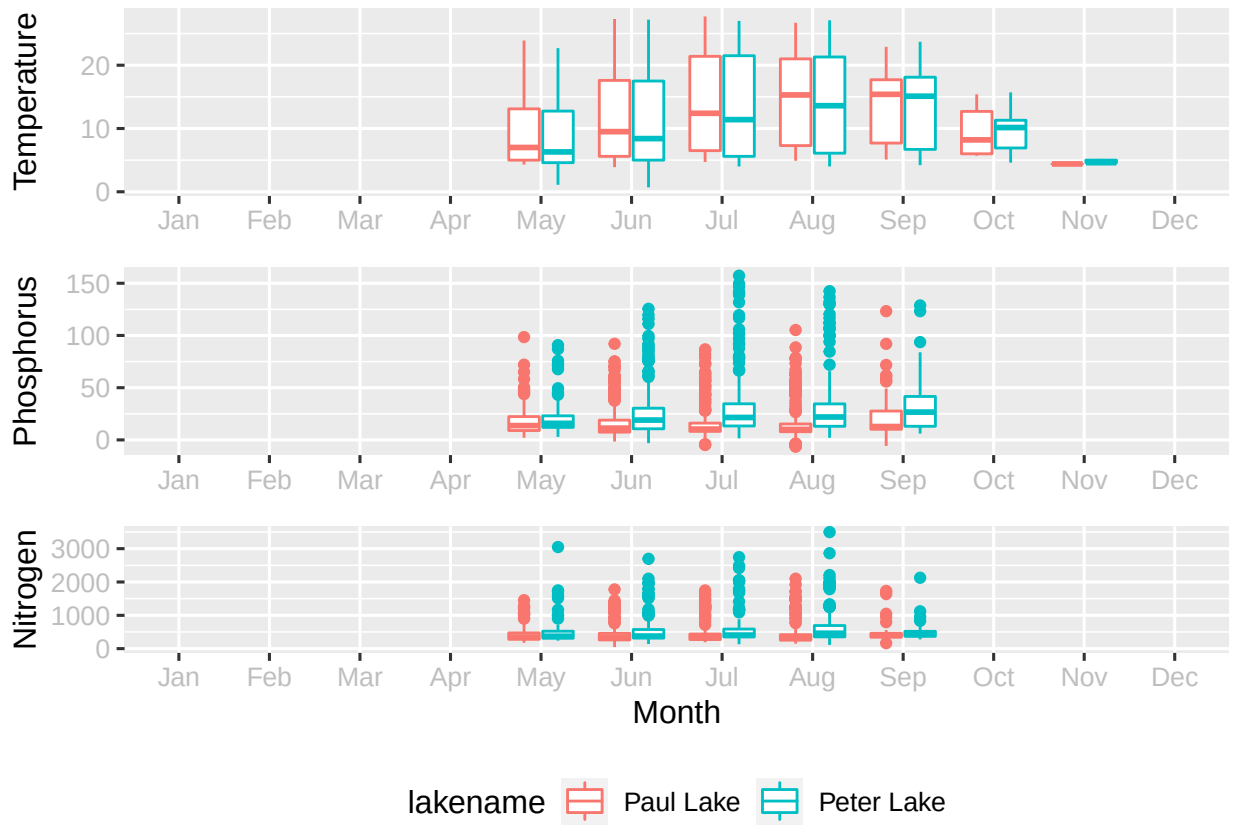
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).

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

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

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
print(cowplot1)
```




```
#displaying cowplot
```

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

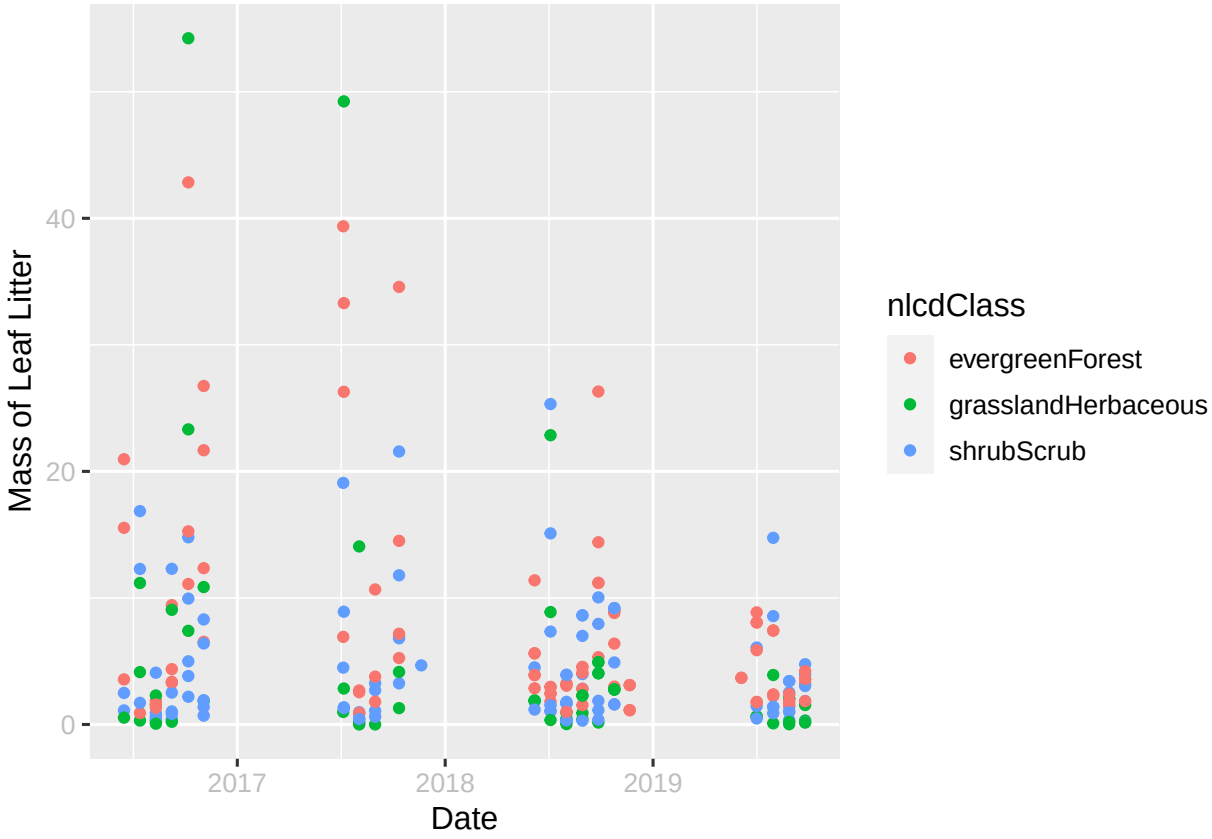
> Answer: While temperature is consistent between the two lakes, Peter Lake has a higher Nitrogen and Phosphorus content throughout the year.

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.

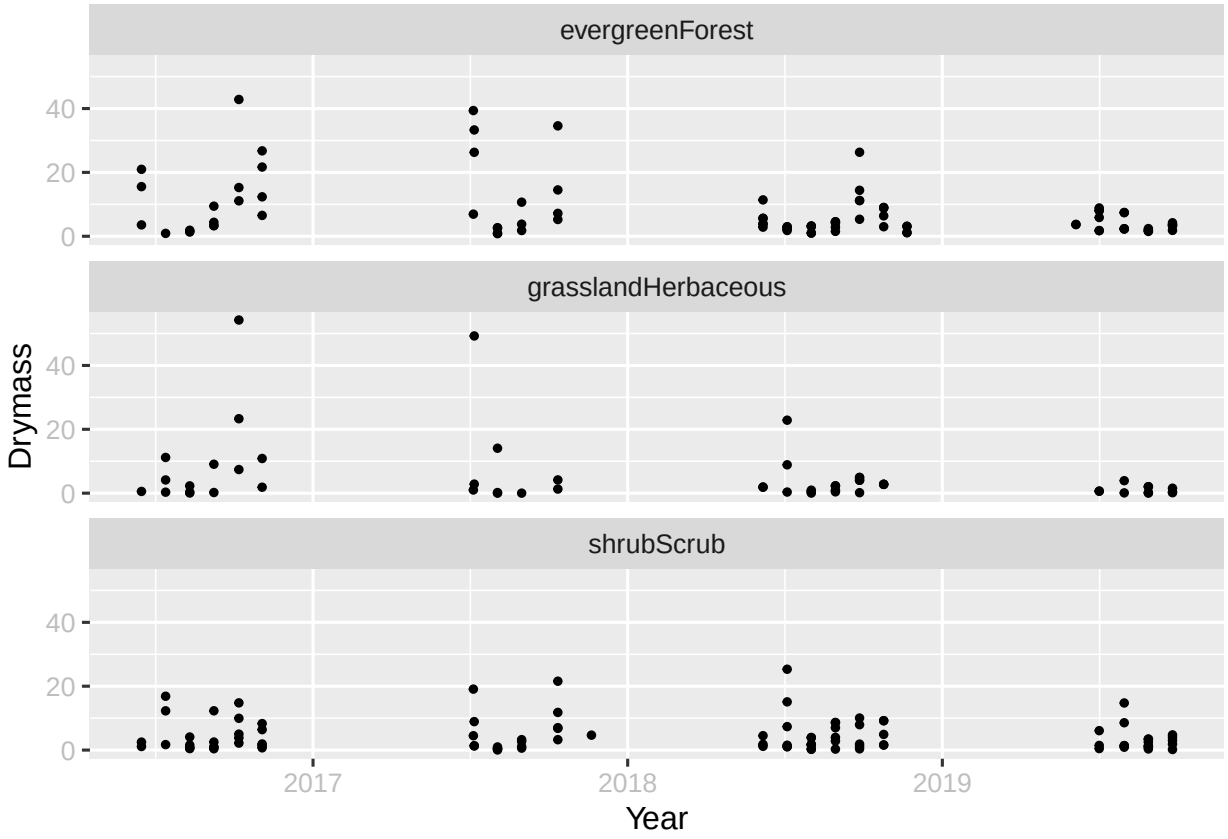
```
# 6
Litter_Mass_dataNEW <- Litter_Mass_data %>%
    filter(functionalGroup == "Needles") %>%
    group_by(dryMass, collectDate, nlcdClass)
# filtering and grouping the data needed for the plot

litter_plot1 <- ggplot(Litter_Mass_dataNEW, aes(x = collectDate, y = dryMass)) +
    geom_point(aes(color = nlcdClass)) + xlab("Date") + ylab("Mass of Leaf Litter") +
    plot_theme
print(litter_plot1)
```




```
# creating the plot grouped by color 7

litter_plot2 <- ggplot(Litter_Mass_dataNEW, aes(x = collectDate, y = dryMass)) +
    geom_point(size = 1) + plot_theme + facet_wrap(vars(nlcdClass), nrow = 3) + xlab("Year") +
    ylab("Drymass")
print(litter_plot2)
```




```
# creating the plot separated by category
```

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

> Answer: I think plot number 7 is more effective, having the dry mass already separated by category even without color is more clear than plot 6 where the mass is separated by color but all stacked on top of eachother.