Assignment 5: Data Visualization

Rachel Gordon - Section 1

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 Monday, February 14 at 7:00 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 (use the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed.csv`] version).

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

```
#1 verifying working directory and loading tidyverse
getwd()
```

```
## [1] "/Users/rachelgordon/Documents/Duke Spring 2022/Environmental Data Analytics/Environmental_Data_A
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

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

```
#uploading NTL-LTER processed data file
PeterPaulChemNutrients.Processed <- read.csv(
  "./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)

#uploading Niwot Ridge Litter processed data
```

```r
NiwotLitter.Processed <- read.csv(
  "./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

#2
class(PeterPaulChemNutrients.Processed$sampledate) #checking format
```

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

```r
PeterPaulChemNutrients.Processed$sampledate <- as.Date(
  PeterPaulChemNutrients.Processed$sampledate, format = "%Y-%m-%d") #changing to date format

class(NiwotLitter.Processed$collectDate) #checking format
```

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

```r
NiwotLitter.Processed$collectDate <- as.Date(
  NiwotLitter.Processed$collectDate, format = "%Y-%m-%d")#changing to date format


class(NiwotLitter.Processed$collectDate) #rechecking format
```

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

Define your theme

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

```r
#3
#setting my theme
RGtheme <- theme_gray(base_size = 18) +
  theme(axis.text = element_text(color = "blue"))


theme_set(RGtheme)
```

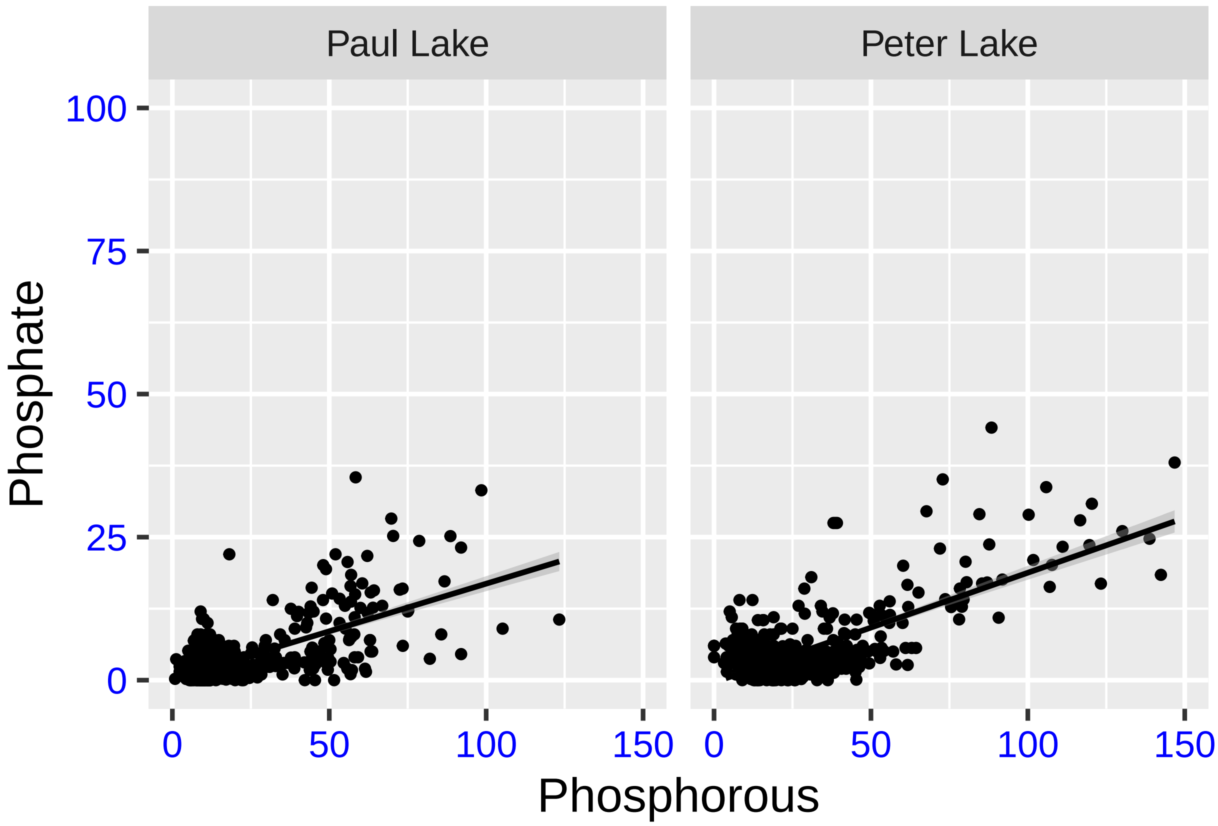
Create graphs

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 `ylim()`).

```r
#4
#plotting phosphorous by phosphate
PeterPaul.Phosporous.Phosphate <-
  ggplot(PeterPaulChemNutrients.Processed, aes(
    x= tp_ug, y = po4))+ #defining axes
          geom_point()+ #scatter plot
          facet_wrap(vars(lakename))+
  xlim(0,150)+ #hiding extreme values
  ylim(0,100)+ #hiding extreme values
  geom_smooth(method = 'lm', color="black")+ #line of best fit
  labs(x="Phosphorous", y="Phosphate")
print(PeterPaul.Phosporous.Phosphate)
```

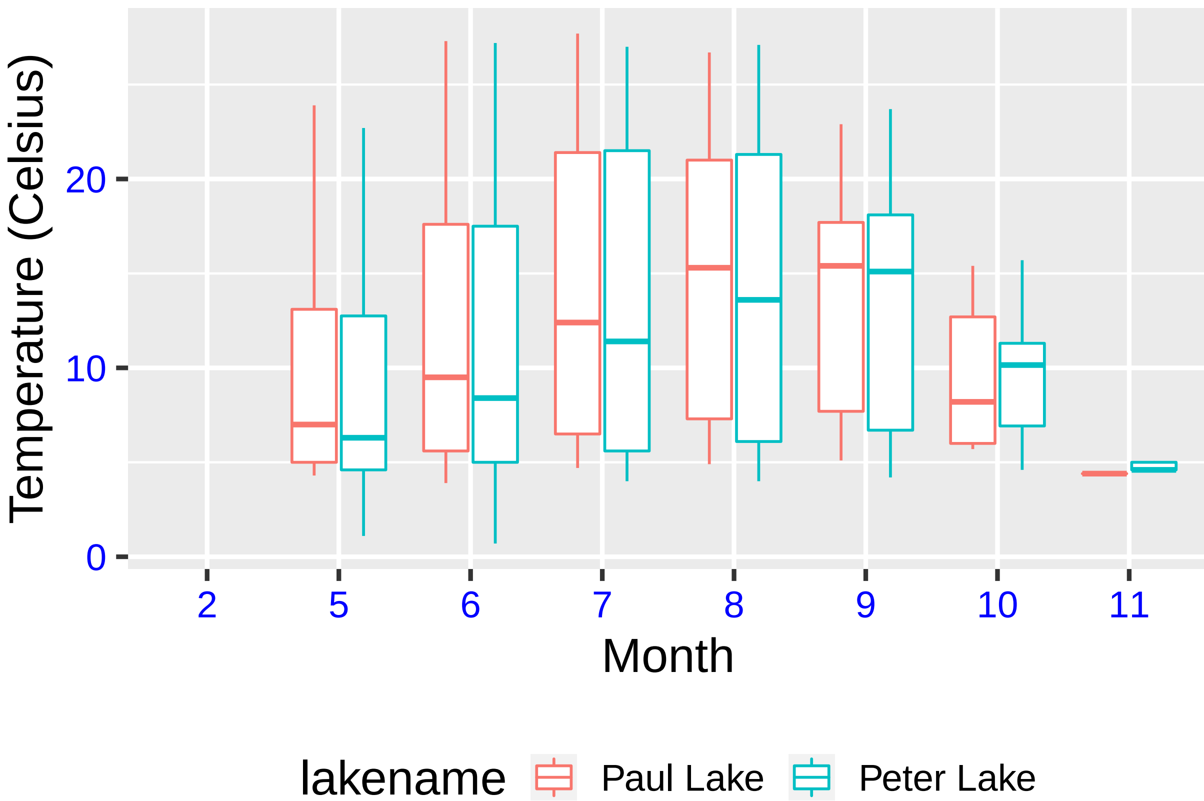
```
## `geom_smooth()` using formula 'y ~ x'

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

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

## Warning: Removed 2 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.

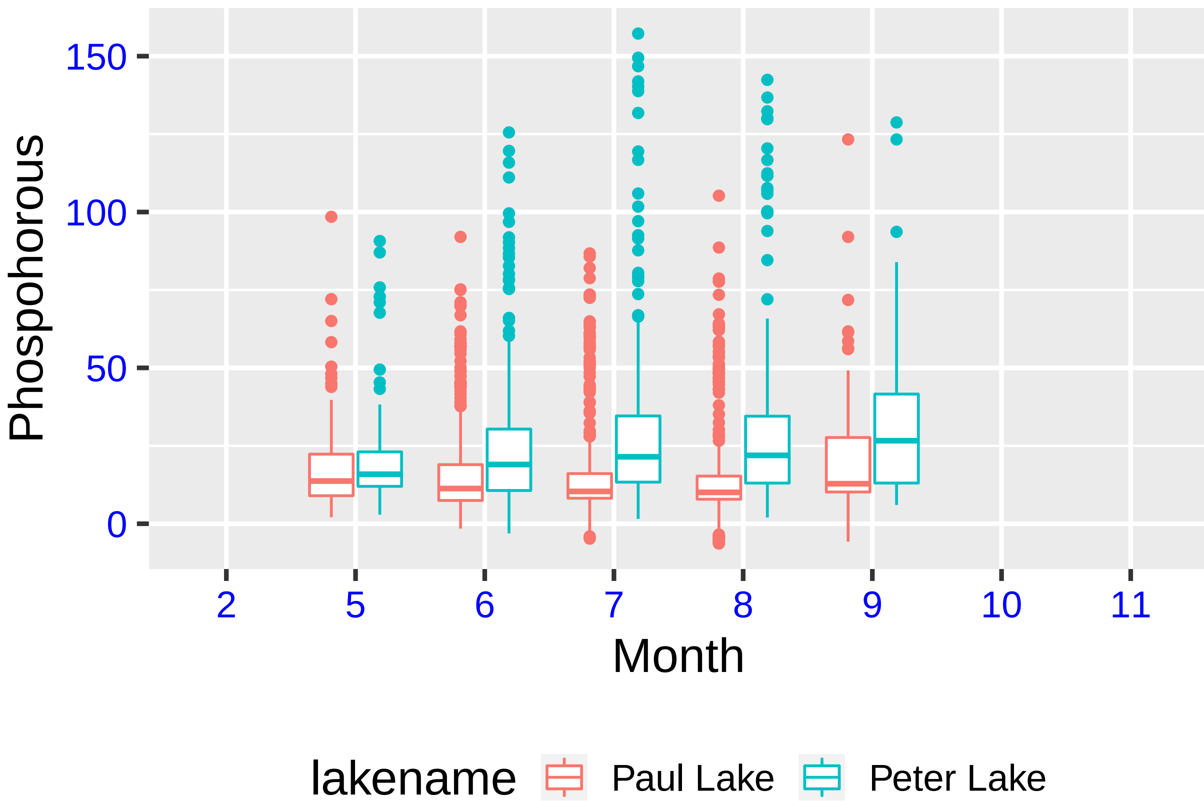
```
#5

PeterPaulChemNutrients.Processed$month <- as.factor(
  PeterPaulChemNutrients.Processed$month) #changing date to category

#temperature plot
PeterPaul.Temp <-
  ggplot(PeterPaulChemNutrients.Processed, aes(x = month, y = temperature_C,))+
  geom_boxplot(aes(color=lakename)) + theme(legend.position = "bottom")+
  labs(x="Month", y="Temperature (Celsius)")
print(PeterPaul.Temp) #boxplot for temperature
```

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



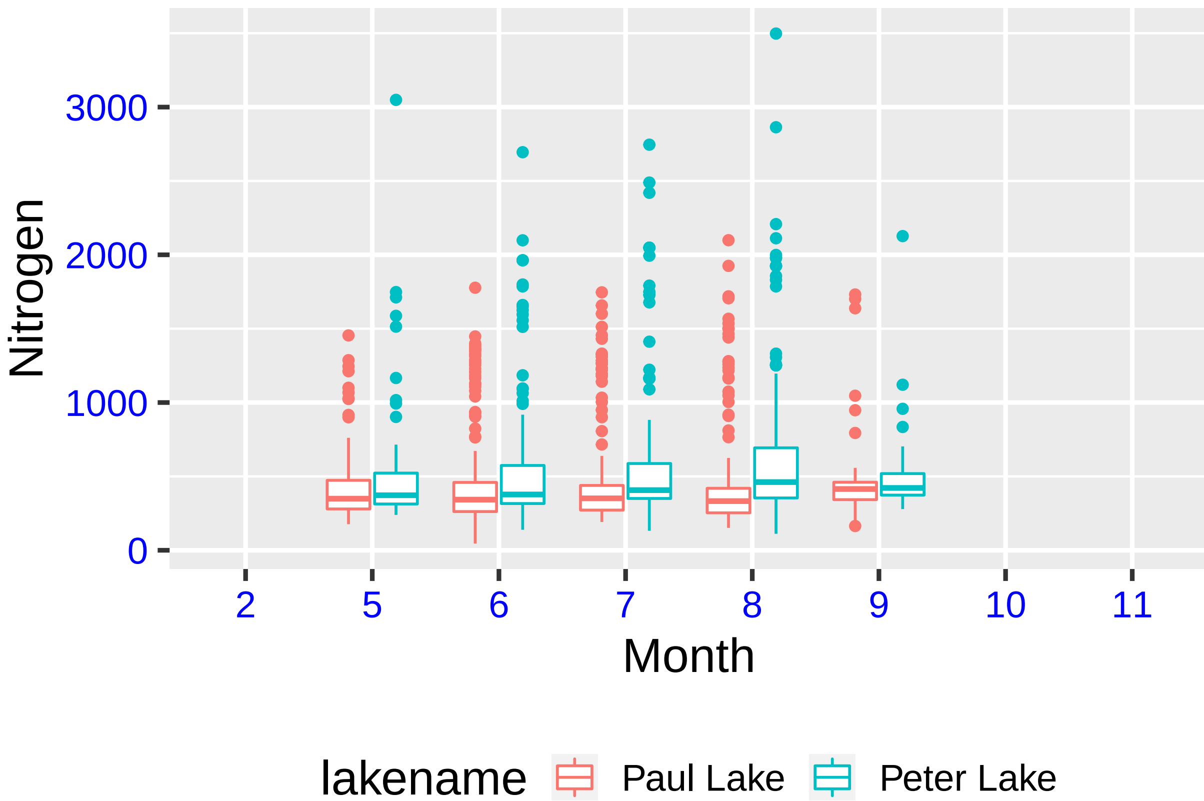

```
#tp plot
PeterPaul.TP <-
  ggplot(PeterPaulChemNutrients.Processed, aes(x= month, y = tp_ug))+
  geom_boxplot(aes(color=lakename)) + theme(legend.position = "bottom")+
  labs(x="Month", y="Phospohorous")
print(PeterPaul.TP) #boxplot for TP
```

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




```
#tn plot
PeterPaul.TN <-
  ggplot(PeterPaulChemNutrients.Processed, aes(x=month, y= tn_ug)) +
  geom_boxplot(aes(color=lakename)) + theme(legend.position = "bottom")+
  labs(x= "Month", y = "Nitrogen")
print(PeterPaul.TN) #boxplot for TN
```

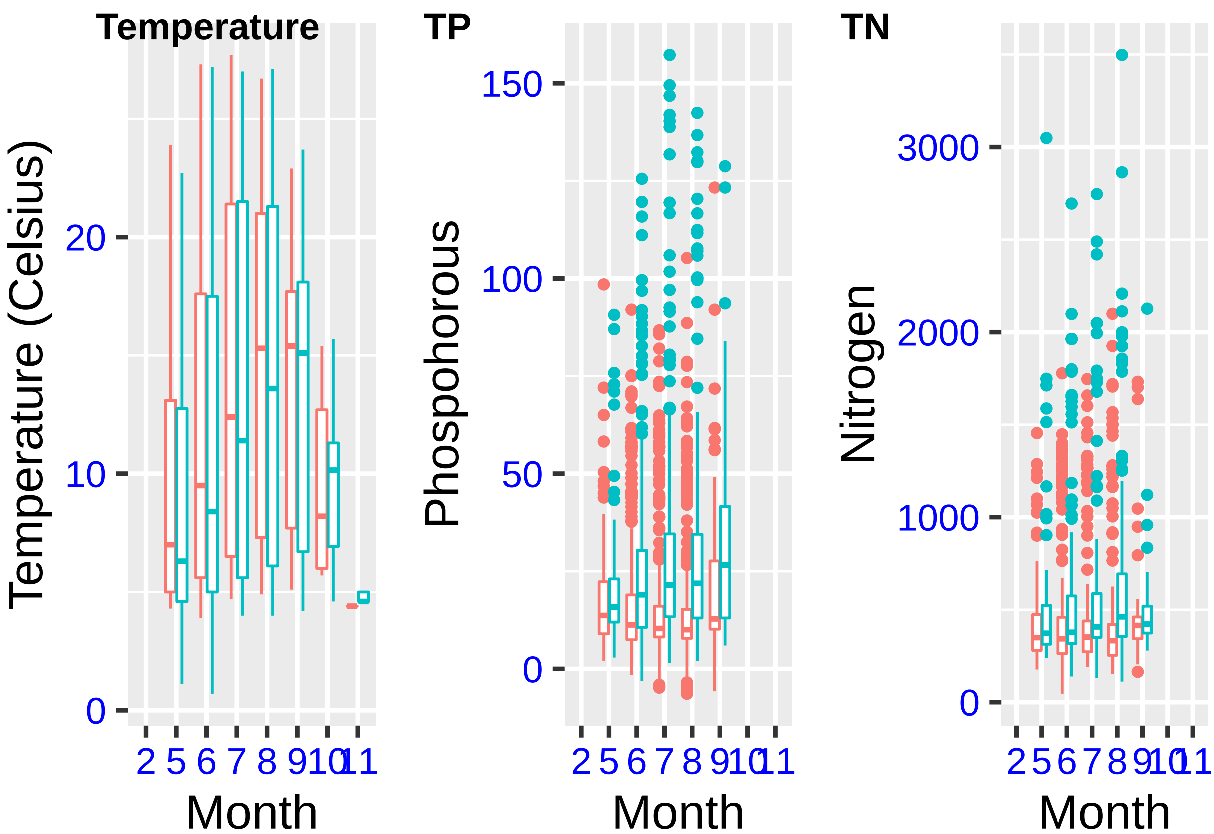
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```




```
#load cowplot
library(cowplot)

#combining plots without legends
PeterPaul.Plots <- plot_grid(PeterPaul.Temp +theme(
  legend.position = "none"), PeterPaul.TP + theme(
    legend.position = "none"), PeterPaul.TN + theme(
      legend.position = "none"),
        labels= c('Temperature', 'TP', 'TN'), nrow = 1)
```

```
## 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).
```

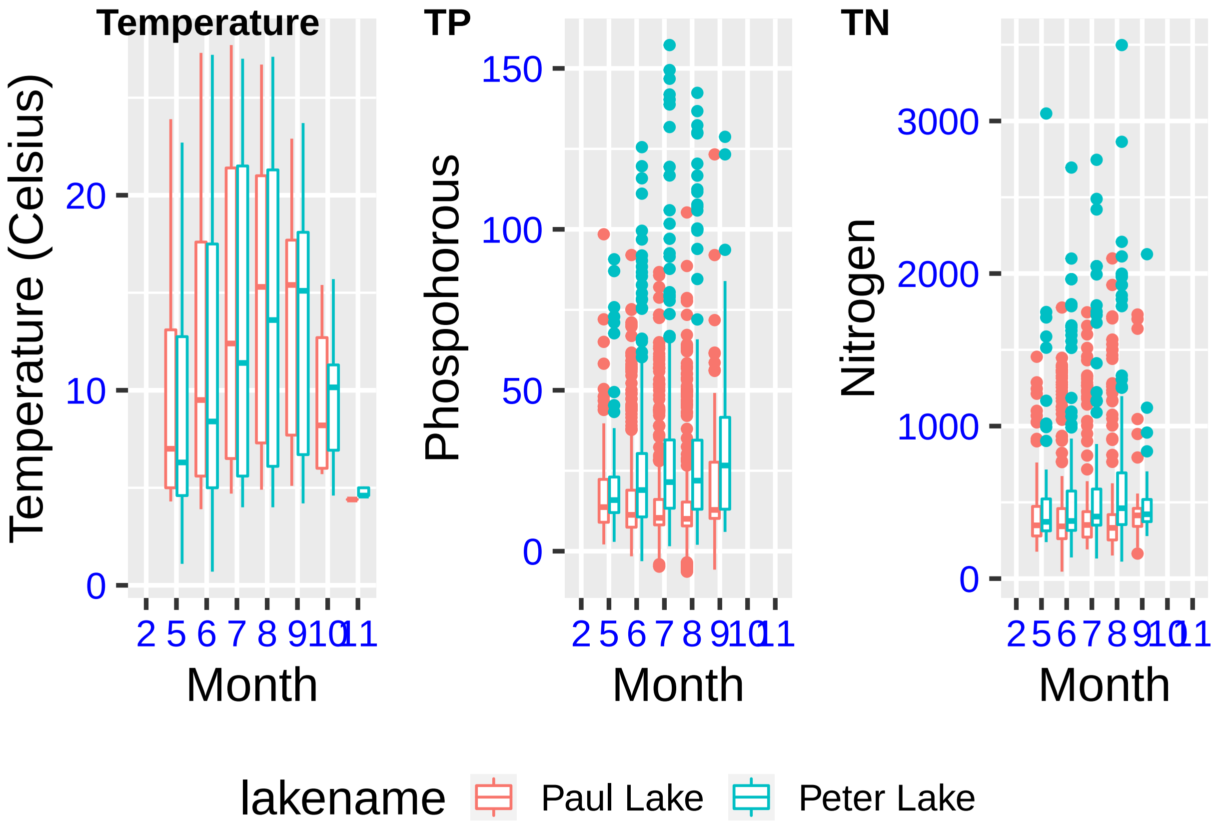
```
print(PeterPaul.Plots)
```




```r
#getting legend from TN plot
PeterPaul.legend <- get_legend(PeterPaul.TN)
```

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

```r
#creating final cowplot
PeterPaul.Plots.Final <- plot_grid((PeterPaul.Plots), (
  PeterPaul.legend), nrow = 2, rel_heights = c(3,.5))
print(PeterPaul.Plots.Final)
```

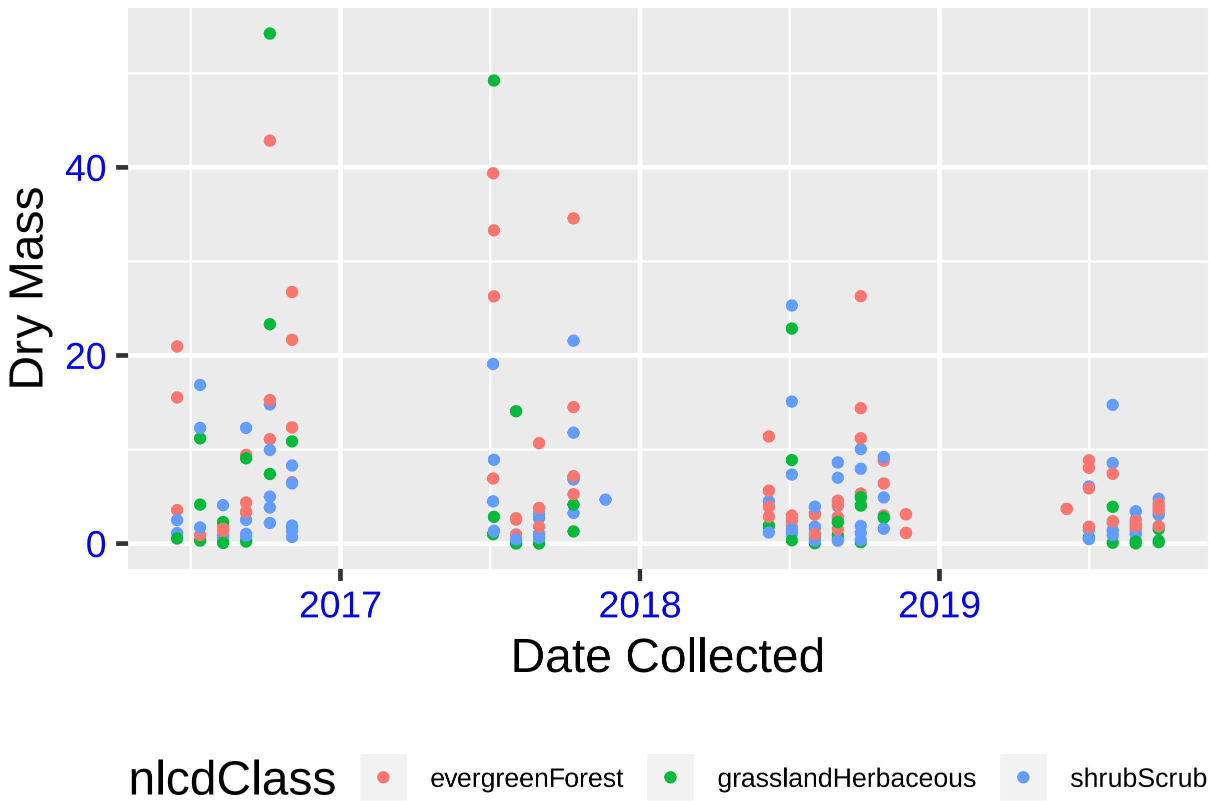


Question: What do you observe about the variables of interest over seasons and between lakes?

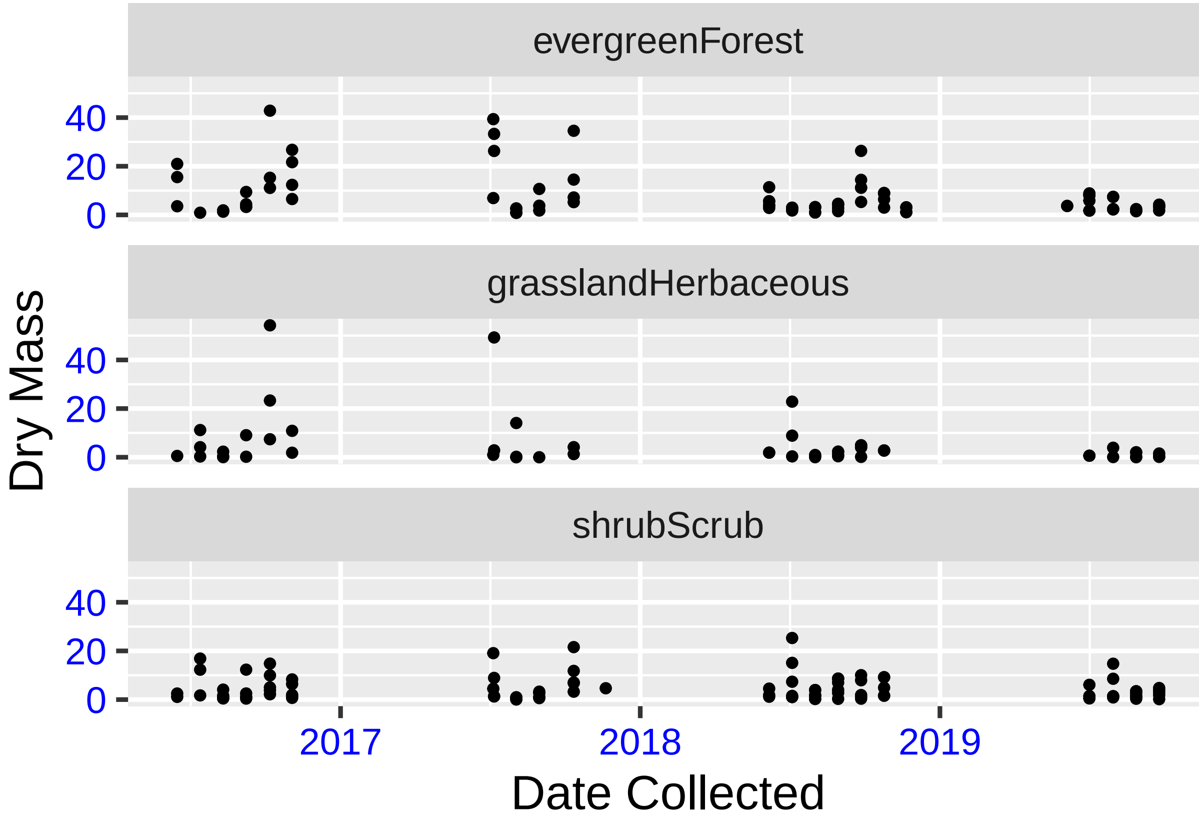
> Answer: There are many outliers across Peter and Paul lakes for both nitrogen and phosphorous, which may make it more difficult to determine certain observations. However, it is apparent that as the temperatures increase, the levels of tp and tn increase. Additionally, the temperature is generally greater in Peter Lake than Paul Lake, which leads to overall higher levels of nitrogen and phosphorous over time in 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
#plotting "Needles" subset with color aesthetic
Niwot.Needles1 <-ggplot(
  subset(NiwotLitter.Processed, functionalGroup == "Needles"),
         aes(x=collectDate, y = dryMass, color=nlcdClass))+
  geom_point() + theme(legend.position = "bottom",legend.text=element_text(size=10)) +
  ylab("Dry Mass") + xlab("Date Collected")
print(Niwot.Needles1)
```




```
#7
#faceting Niwot.Needles1 plot
Niwot.Needles2 <-ggplot(
  subset(NiwotLitter.Processed, functionalGroup== "Needles"),
  aes(x=collectDate, y= dryMass))+
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow=3)+
  xlab("Date Collected")+
  ylab("Dry Mass")
print(Niwot.Needles2)
```



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

> Answer: I believe the faceted plot (#7) is more effective. Although it is helpful to see the data all in one plot by color, having it faceted out makes it easier to see the mass levels of the different NLCD class types broken out and to compare to each other over time.