

Assignment 5: Data Visualization

Sara Diamond

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 checking working directory and loading in packages and loading datasets
getwd()
```

```
## [1] "/Users/saradiamond/Documents/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5      v purrr  0.3.4
```

```
## v tibble  3.1.4      v dplyr  1.0.7
```

```
## v tidyr   1.1.4      v stringr 1.4.0
```

```
## v readr   2.0.2      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

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

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

```
library(cowplot)
```

```
PeterPaul.ChemNu.Processed <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                                         stringsAsFactors = TRUE)
```

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

#2 checking the date column

class(PeterPaul.ChemNu.Processed$sampldate)

## [1] "factor"

PeterPaul.ChemNu.Processed$sampldate <- as.Date(PeterPaul.ChemNu.Processed$sampldate,
                                                format = "%Y-%m-%d")

#changing to date
class(PeterPaul.ChemNu.Processed$sampldate) #checking the class

## [1] "Date"

class(Litter.Processed$collectDate)

## [1] "factor"

Litter.Processed$collectDate <- as.Date(Litter.Processed$collectDate,
                                        format = "%Y-%m-%d") #changing the date
class(Litter.Processed$collectDate) #checking to make sure it is a date

## [1] "Date"
```

Define your theme

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

```
#3 creating my theme and setting it to default

Sara.Theme <- theme_light(base_size = 14) +
  theme(axis.text = element_text(color = "blue"),
        legend.position = "right")

theme_set(Sara.Theme) #setting 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 ylim()).

```
#4 phosphorus by phosphate graph

PP.Phos.Graph <-
  ggplot(PeterPaul.ChemNu.Processed, aes(x = tp_ug, y = po4)) + #defining aesthetic
  geom_point() + #using scatter plot
  facet_wrap(vars(lakename)) + #allowing more than one plot
  xlim(0,150) + #hiding extreme values
  ylim(0,100) + #hiding extreme values
  ylab("Phosphate") + #changing y axis label
  xlab("Phosphorus") + #changing x axis label
```

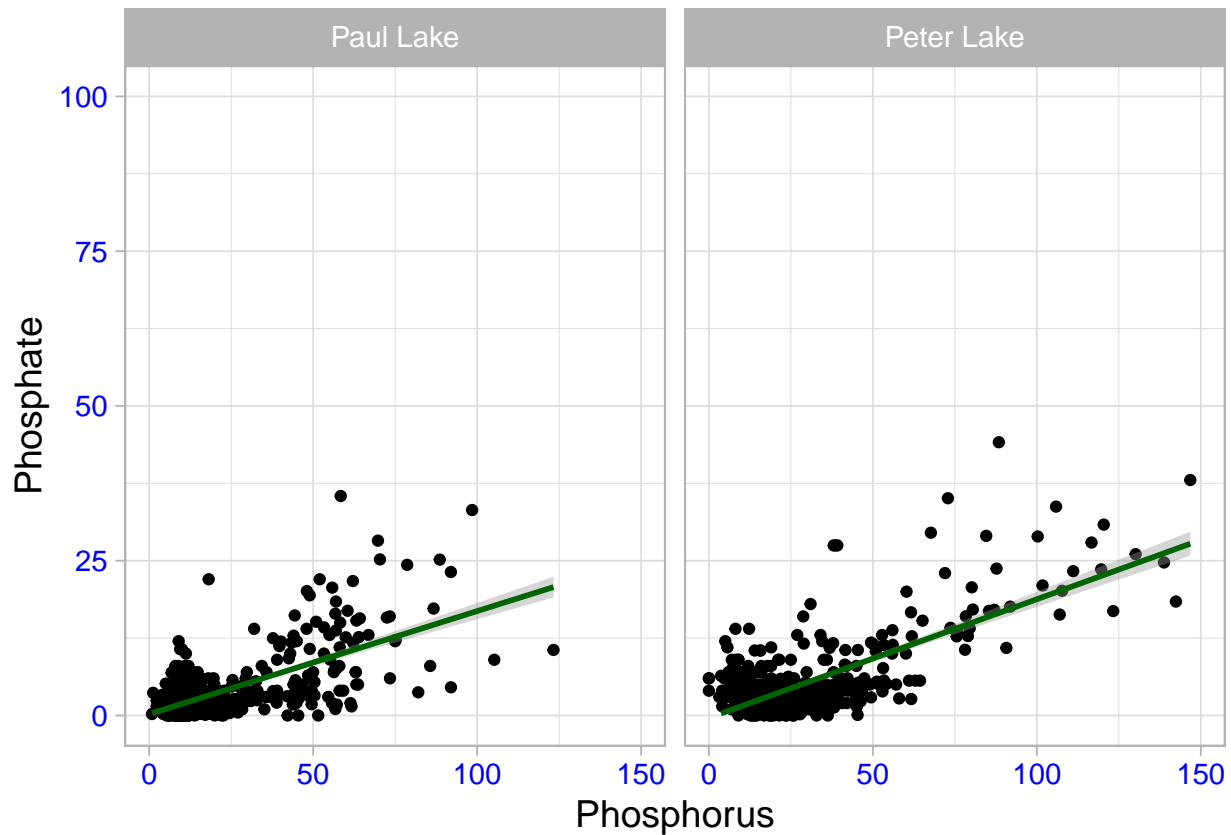
```
geom_smooth(method = "lm", color = "darkgreen") #adding line of best fit
print(PP.Phos.Graph) #showing graph
```

```
## `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 creating 3 boxplots
```

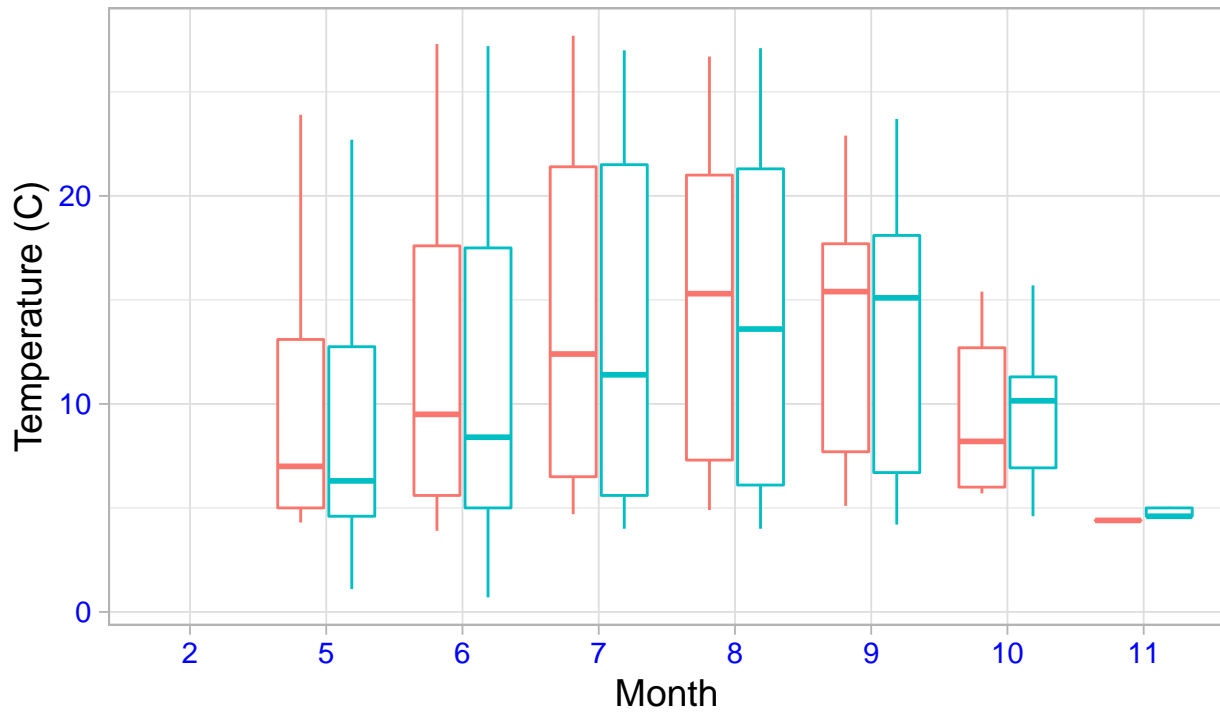
```
#changing month to category
```



```
PeterPaul.ChemNu.Processed$month <- as.factor(PeterPaul.ChemNu.Processed$month)
```

```
#temp plot using boxplot
```

```
Temperature.Plot <- ggplot(PeterPaul.ChemNu.Processed,
                           aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color=lakename)) +
  theme(legend.position = "bottom") +
  xlab("Month") +
  ylab("Temperature (C)")
print(Temperature.Plot)
```

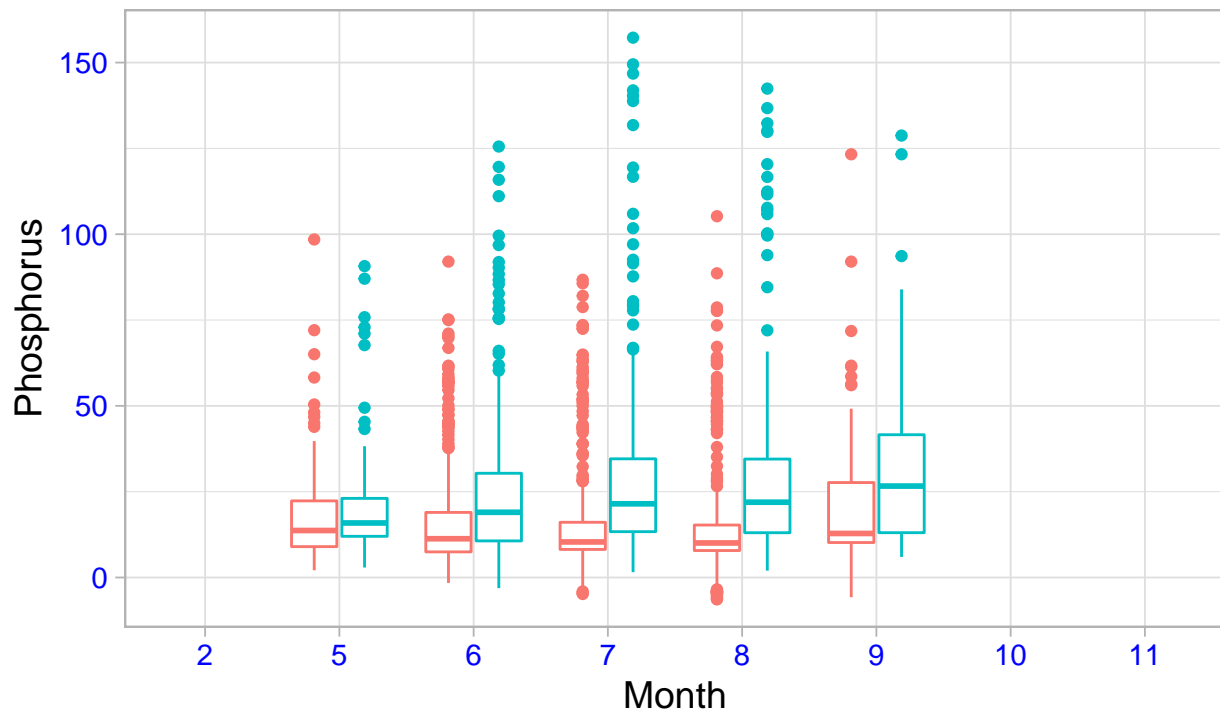
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```





lakename  Paul Lake  Peter Lake

```
#TP plot using boxplot
TP.Plot <- ggplot(PeterPaul.ChemNu.Processed,
                  aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color=lakename)) +
  theme(legend.position = "bottom") +
  xlab("Month") +
  ylab("Phosphorus")
print(TP.Plot)
```

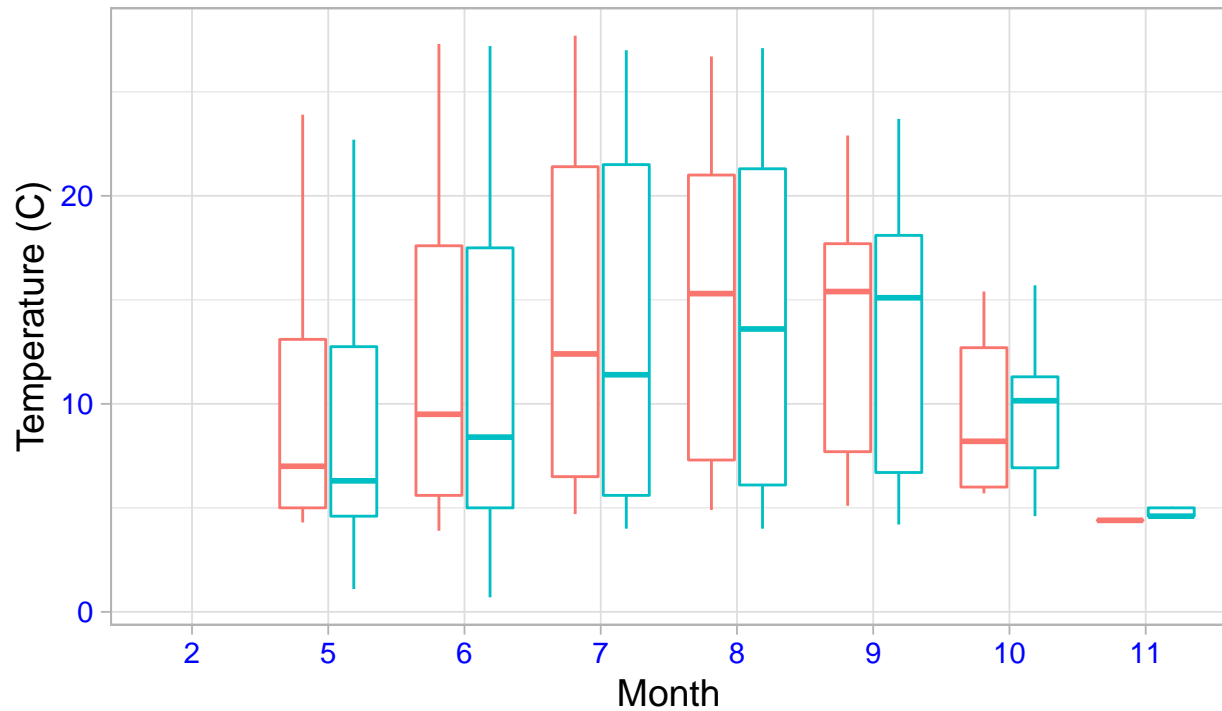
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```





lakename  Paul Lake  Peter Lake

```
#TN plot using boxplot
TN.Plot <- ggplot(PeterPaul.ChemNu.Processed,
                  aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color=lakename)) +
  theme(legend.position = "bottom") +
  xlab("Month") +
  ylab("Nitrogen")
print(Temperature.Plot)
```

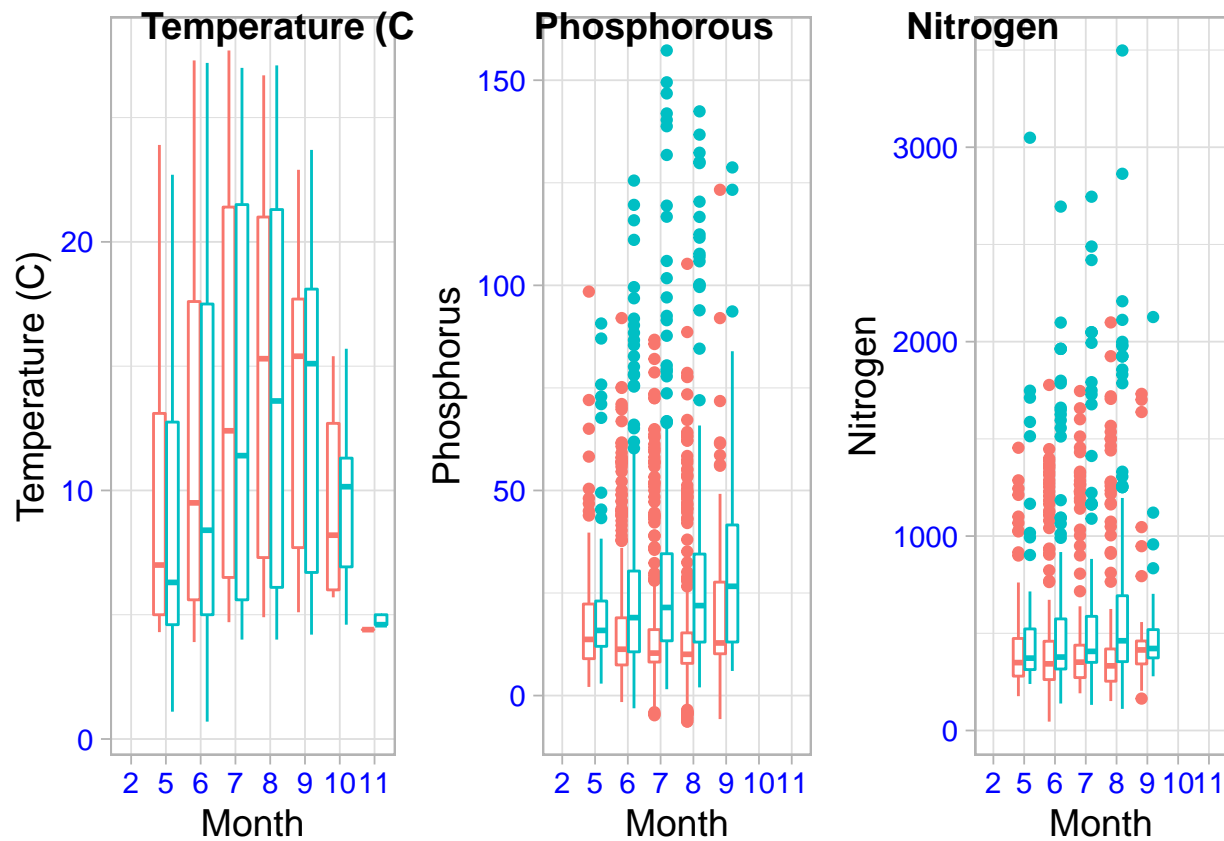
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```



lakename  Paul Lake  Peter Lake

```
#assigning all three plots to one object and in one row
Three.Plot <- plot_grid(Temperature.Plot + theme(legend.position = "none"),
  TP.Plot + theme(legend.position = "none"),
  TN.Plot + theme(legend.position = "none"),
  labels = c("Temperature (C)", "Phosphorous", "Nitrogen"),
  nrow = 1)
```

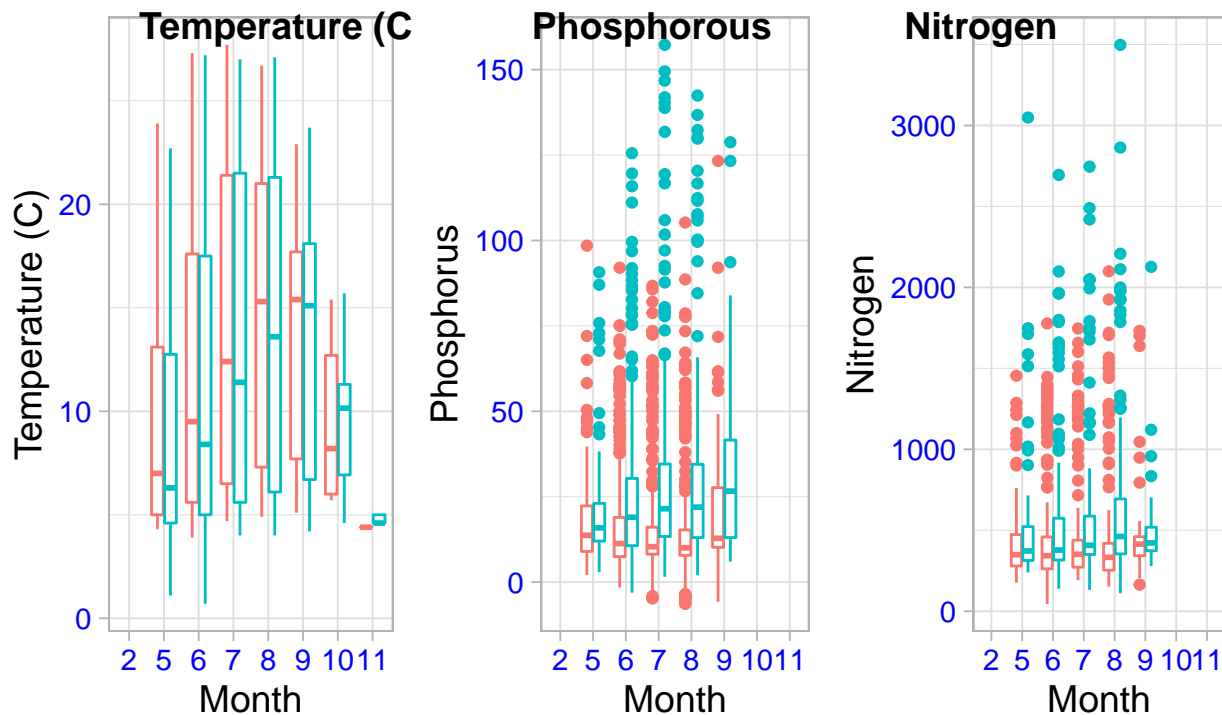
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
print(Three.Plot)
```



```
Three.Plot.Legend <- get_legend(TN.Plot) #pulling a legend
```

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

```
Final.Plot <- plot_grid((Three.Plot), (Three.Plot.Legend), nrow = 2, rel_heights = c(3, 0.5))
print(Final.Plot) #producing final plot
```



lakename ▢ Paul Lake ▢ Peter Lake

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

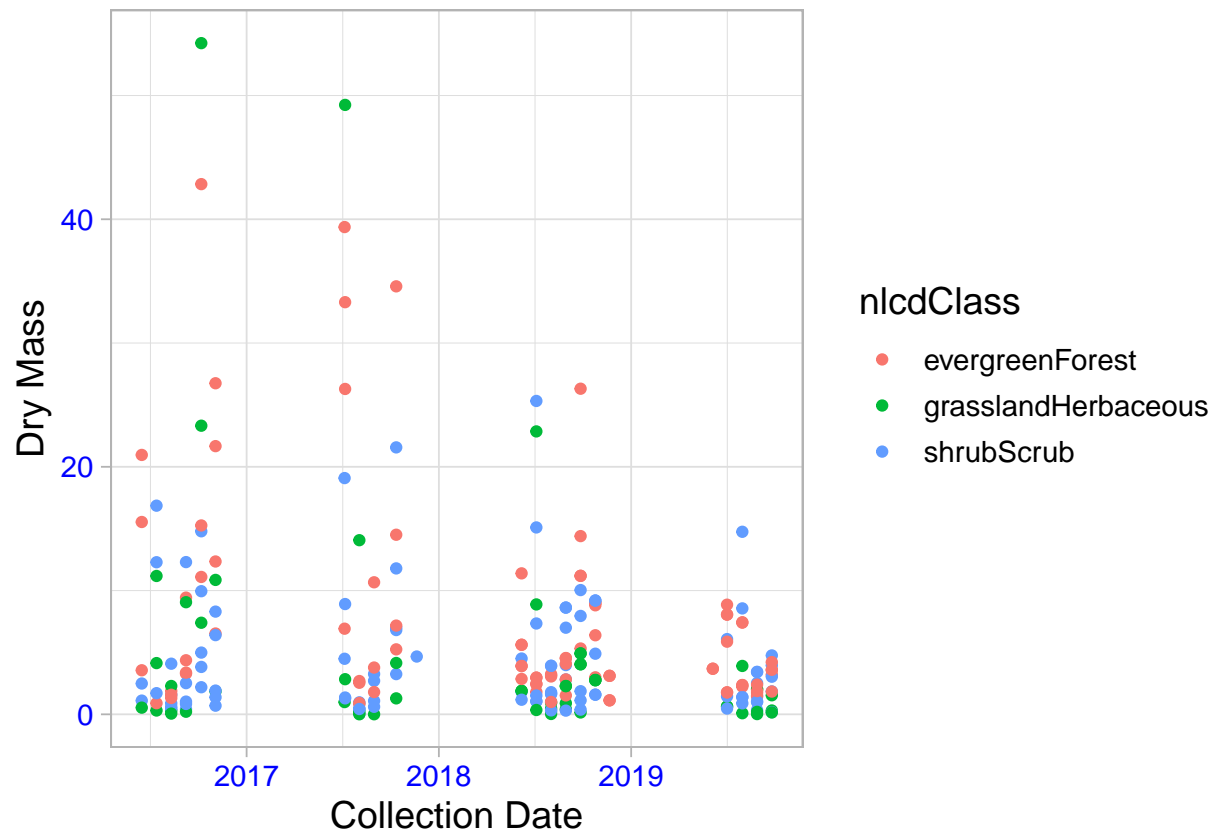
Answer: There are many potential outliers for nitrogen and phosphorous but generally, as the temperature rises so do nitrogen and phosphorous levels and they seem to be higher in Peter Lake than in Paul Lake.

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

Needle.Plot <- #creating first subset plot with needles functionalgroup and nlcd as the color aes
ggplot(subset(Litter.Processed, functionalGroup == "Needles"),
  aes(x=collectDate, y=dryMass, color =nlcdClass))+
  geom_point() +
  theme(legend.position = "right") +
  ylab("Dry Mass") +
  xlab("Collection Date")

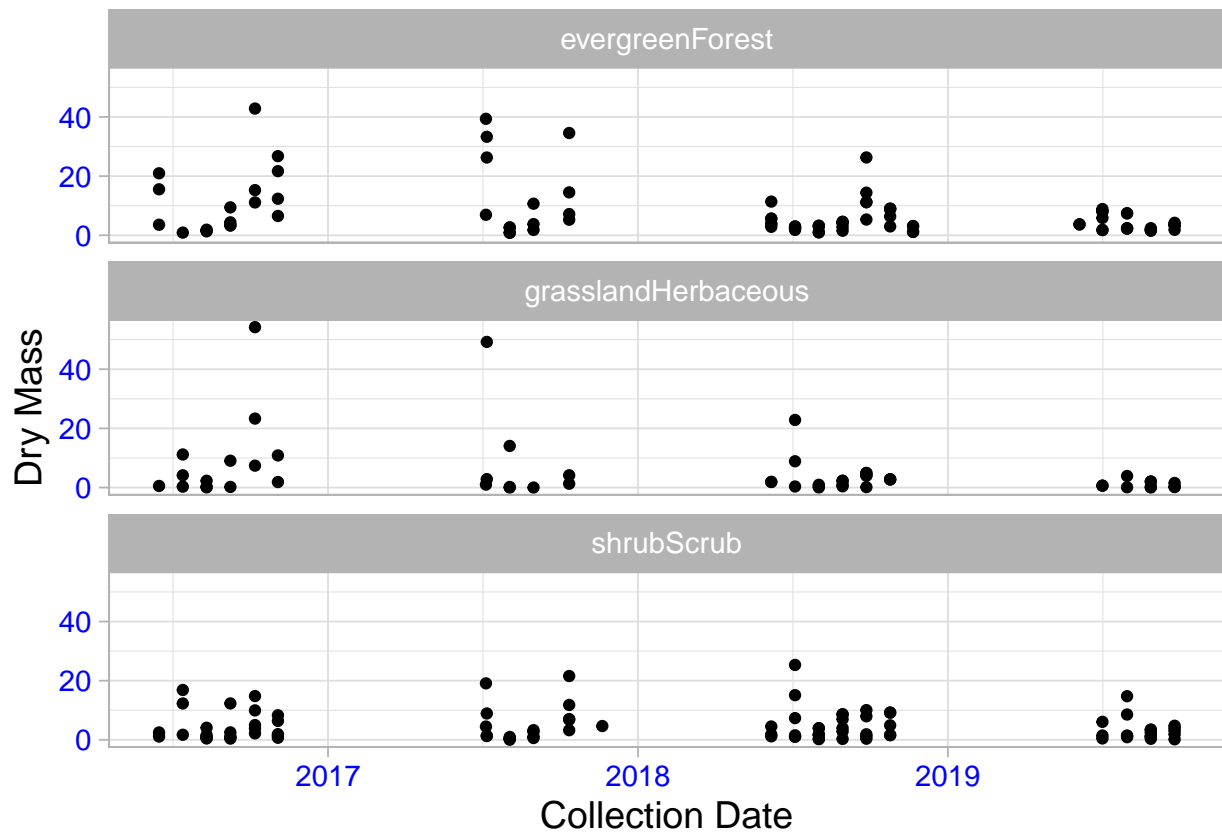
print(Needle.Plot)
```

#7

```
Second.Needle.Plot <- #adding in the facetwrap to same plot
  ggplot(subset(Litter.Processed, functionalGroup == "Needles"),
    aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  xlab("Collection Date") +
  ylab("Dry Mass")

print(Second.Needle.Plot)
```



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

Answer: I believe the facet wrap plot is better, because it separates the values more clearly and is easier to read and understand more so than the first plot in number 6.