Assignment 5: Data Visualization

Enikoe Bihari

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

# check working directory
getwd()
```

```
## [1] "C:/Users/eniko/Documents/Duuuuuke/2021-22/Data Analytics/Environmental_Data_Analytics_2022/Assig
```

```
# load packages
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- 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

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5
## Warning: package ’readr’ was built under R version 4.0.5

## Warning: package ’dplyr’ was built under R version 4.0.5

## Warning: package ’forcats’ was built under R version 4.0.5

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

```r
library(cowplot)
```

```
## Warning: package ’cowplot’ was built under R version 4.0.5
```

```r
library(ggplot2)
# install.packages("gridExtra")
library("gridExtra")
```

```
## Warning: package ’gridExtra’ was built under R version 4.0.5

##
## Attaching package: ’gridExtra’

## The following object is masked from ’package:dplyr’:
##
##     combine
```

```r
# install.packages("grid")
library("grid")
# install.packages("viridisLite")
library(viridis)
```

```
## Loading required package: viridisLite

## Warning: package ’viridisLite’ was built under R version 4.0.5

##
## Attaching package: ’viridis’

## The following object is masked from ’package:viridisLite’:
##
##     viridis.map
```

```r
library(RColorBrewer)

# read in csv files
lake <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                 stringsAsFactors = T)
litter <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                   stringsAsFactors = T)

#2

# change the dates to date object
lake$date <- as.Date(lake$sampledate,
                            format = "%Y-%m-%d")
litter$date <- as.Date(litter$collectDate,
                              format = "%Y-%m-%d")
```

Define your theme

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

```r
#3

# create a theme with gray defaults
theme1 <- theme_gray(base_size = 10) +
  theme(axis.text = element_text(color = "grey50"),
        legend.position = "top")

# set it as the default theme
theme_set(theme1)
```

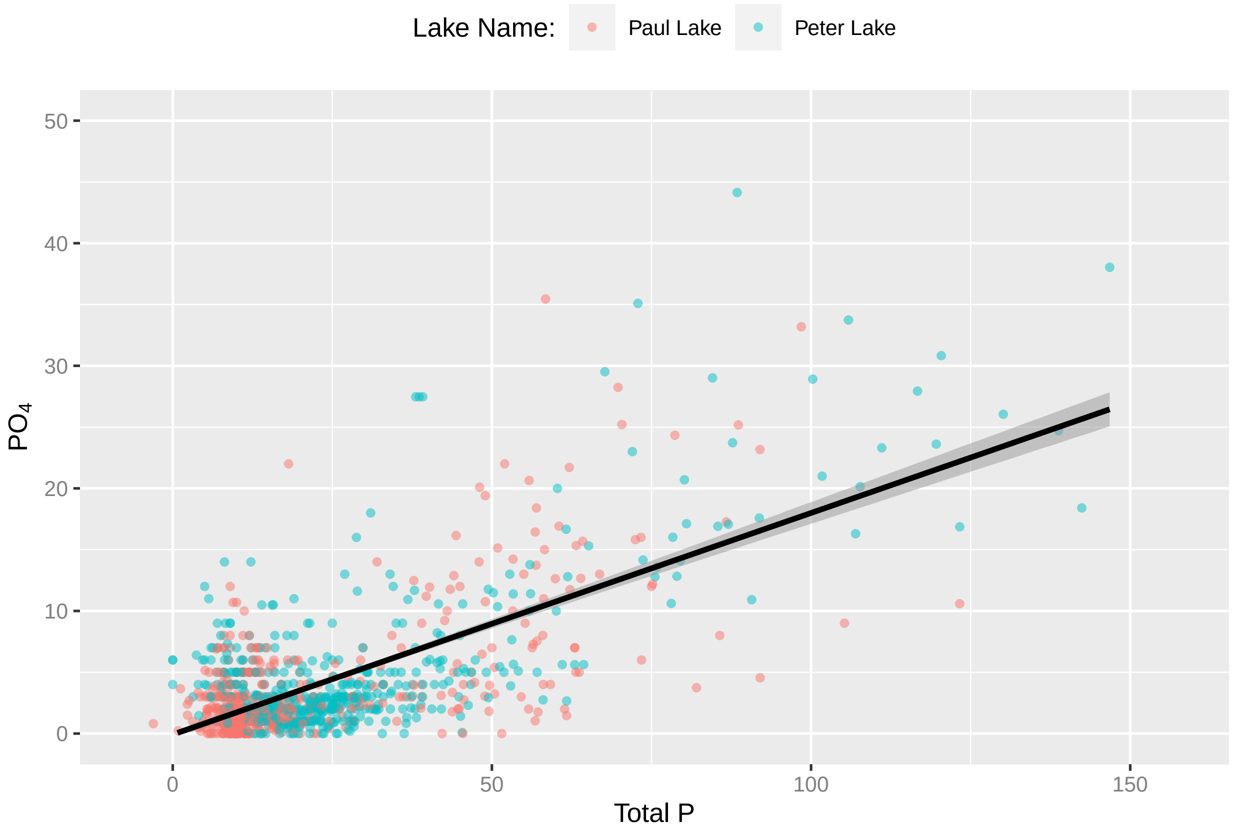
Create graphs

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

# make plot
plot1 <-
ggplot(lake) +
  geom_point(aes(x=tp_ug, y=po4, color=lakename), size = 1, alpha = 0.5) +
  geom_smooth(aes(x=tp_ug, y=po4), method = lm, color = ’black’, alpha = 0.5) +
  # xlim() +
  ylim(0, 50) +
  ylab(expression("PO"[4])) +
  xlab(expression(paste("Total P"))) +
  labs(color = "Lake Name:")

print(plot1)
```

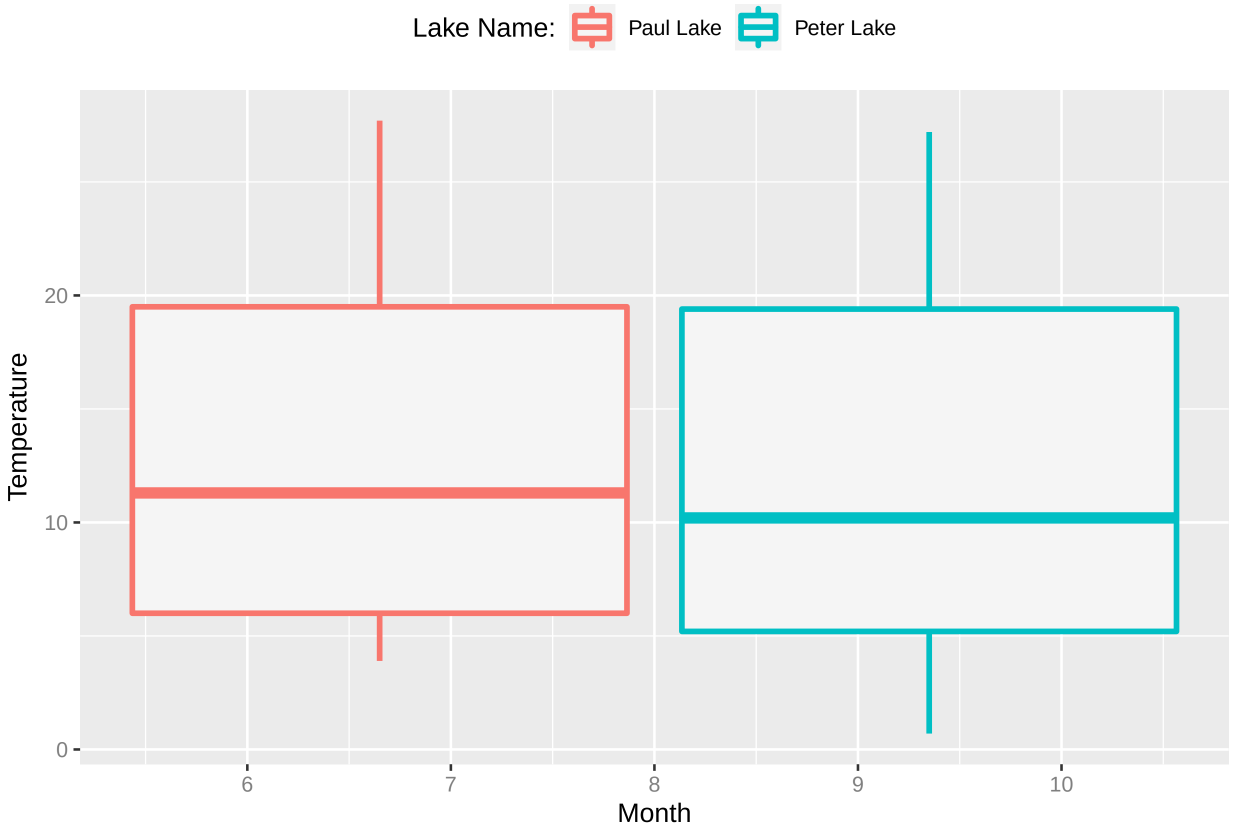
```
## ‘geom_smooth()‘ using formula ’y ~ x’

## 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).
```



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.

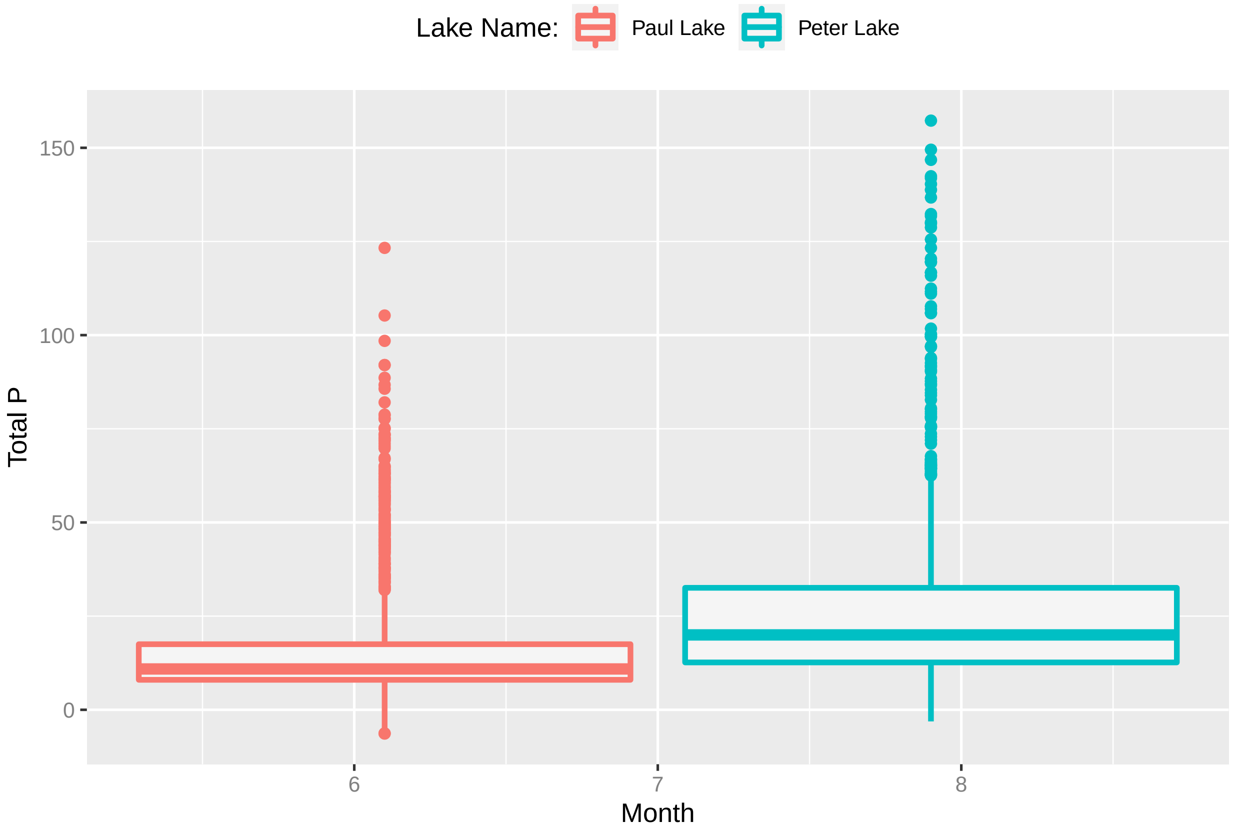
```
#5

# temp plot
plot_temp <-
ggplot(lake) +
  geom_boxplot(aes(x=month, y=temperature_C, color=lakename), fill='gray96', size = 1) +
  ylab(expression("Temperature")) +
  xlab(expression("Month")) +
  labs(color = "Lake Name:")

print(plot_temp)
```

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



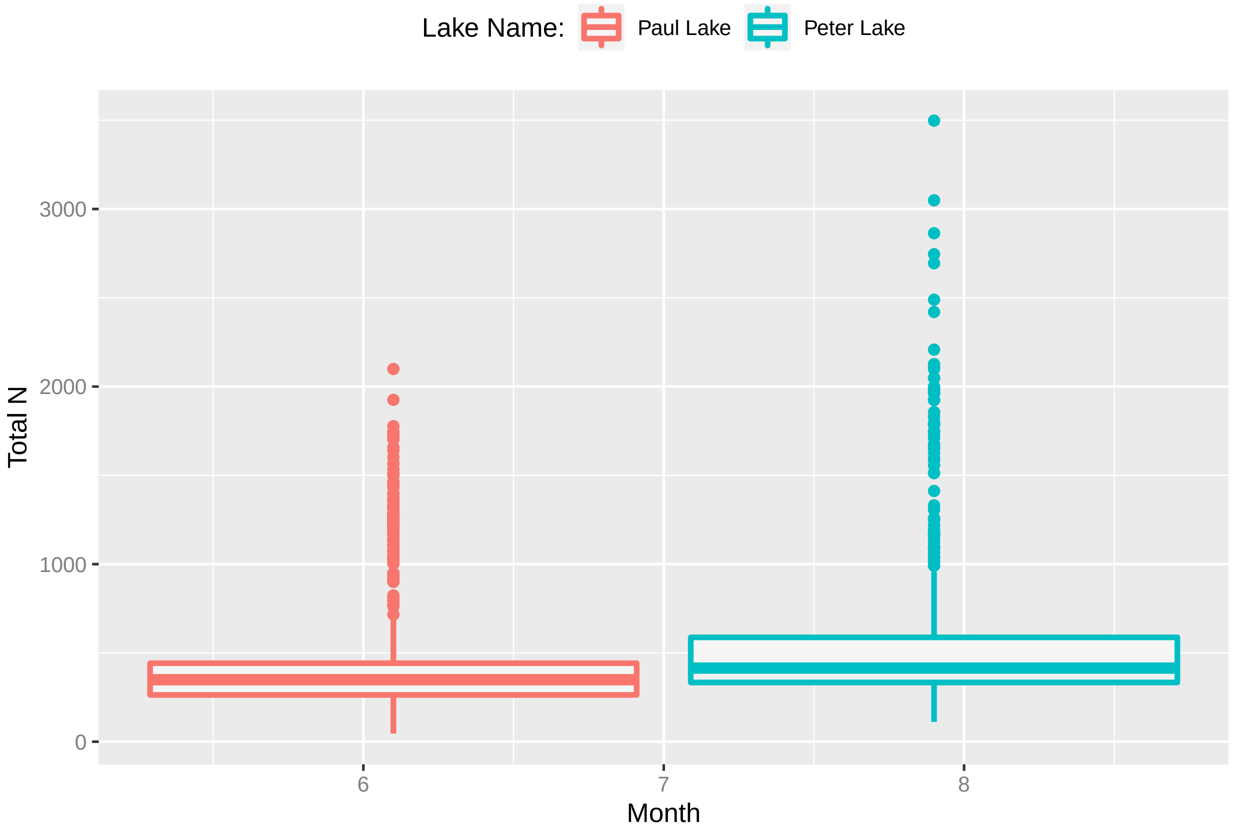
```
# TP plot
plot_TP <-
ggplot(lake) +
  geom_boxplot(aes(x=month, y=tp_ug, color=lakename), fill='gray96', size = 1) +
  xlab(expression("Month")) +
  ylab(expression("Total P")) +
  labs(color = "Lake Name:")

print(plot_TP)
```

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



```
# TN plot
plot_TN <-
ggplot(lake) +
  geom_boxplot(aes(x=month, y=tn_ug, color=lakename), fill='gray96', size = 1) +
  xlab(expression("Month")) +
  ylab(expression("Total N")) +
  labs(color = "Lake Name:")

print(plot_TN)
```

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



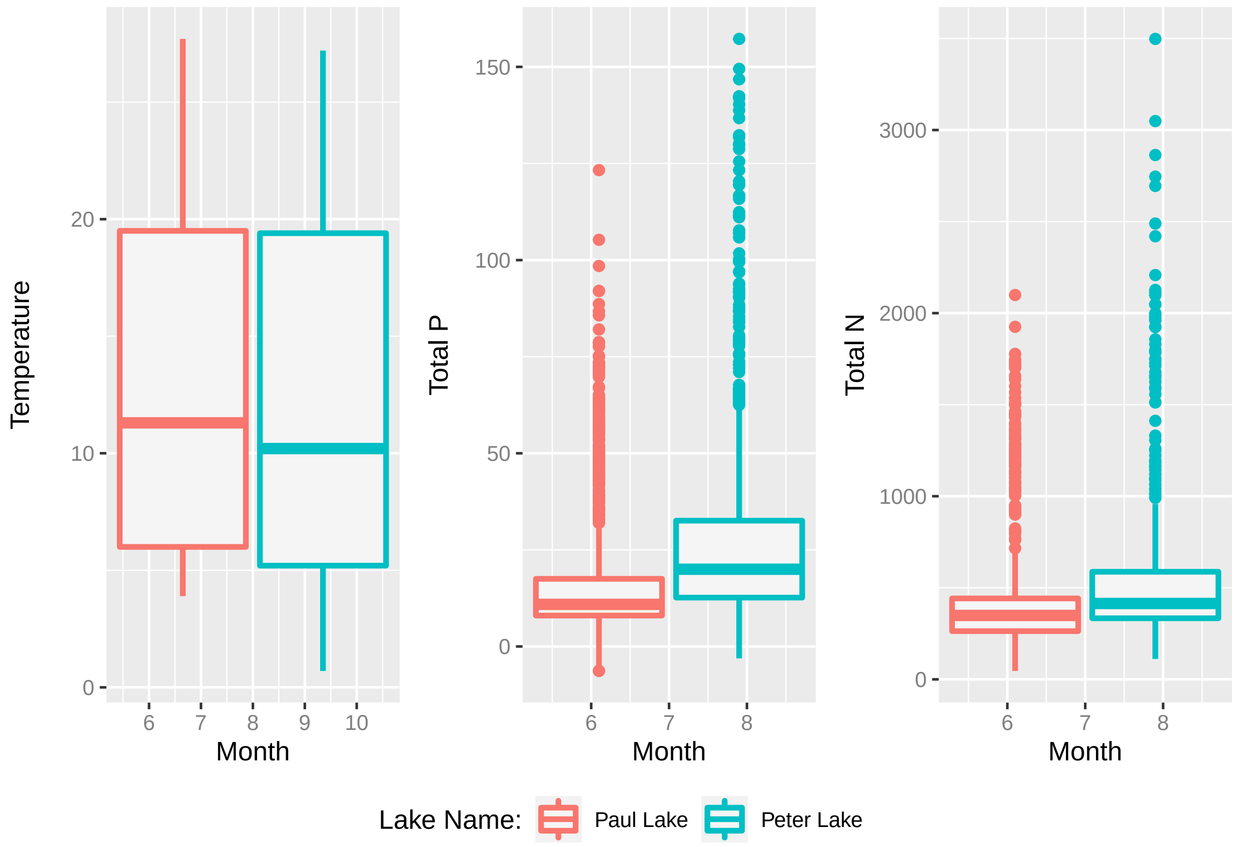
```
# cowplot of all 3
plots_grid <-
  plot_grid(plot_temp + theme(legend.position = 'none'),
            plot_TP + theme(legend.position = 'none'),
            plot_TN + theme(legend.position = 'none'),
            nrow = 1,
            align = 'hv')
```

```
## 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).
```

```
legend_grid = get_legend(plot_TN +
                           guides(color = guide_legend(nrow = 1)) +
                           theme(legend.position = "bottom"))
```

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

```
plot_grid(plots_grid, legend_grid, ncol = 1, rel_heights = c(1, .1))
```

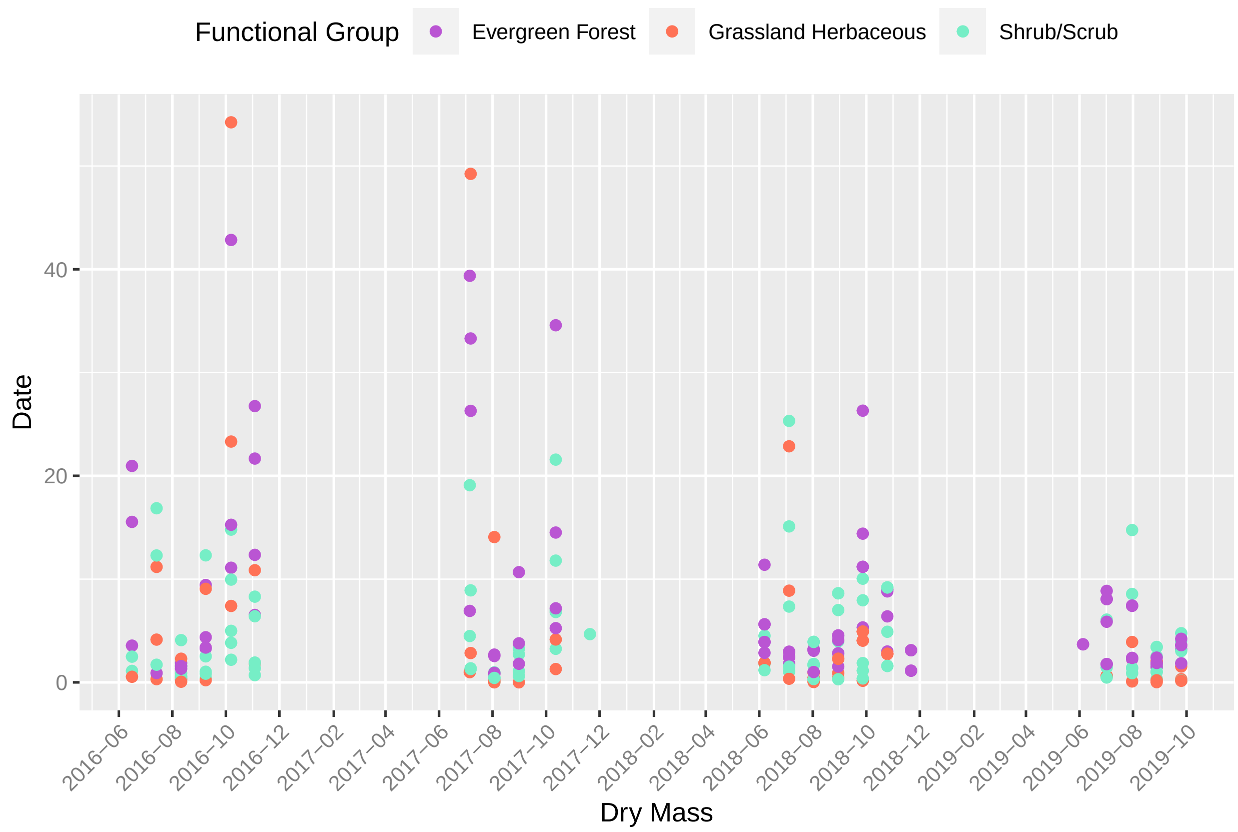


Question: What do you observe about the variables of interest over seasons and between lakes?

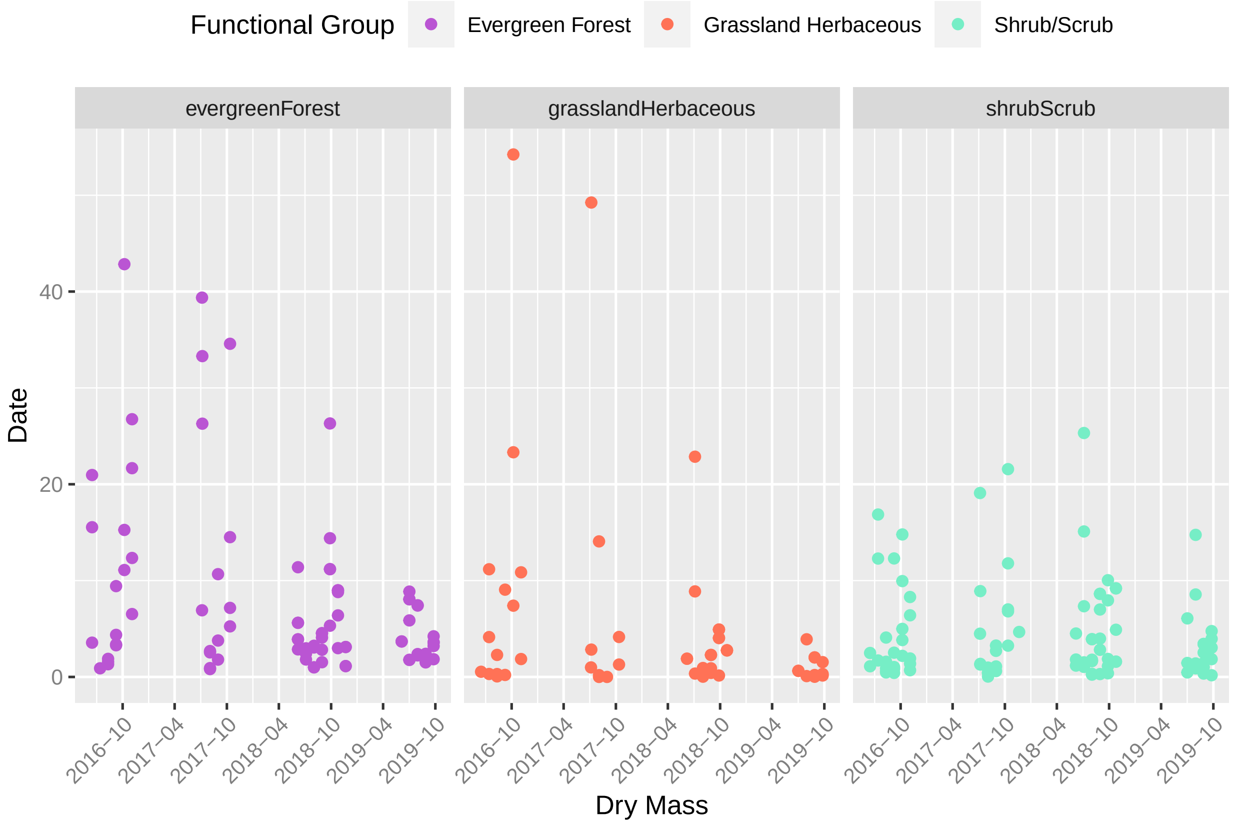
> Answer: Nitrogen and phosphorous were only sampled in 3 months over the summer, and they are both higher in Peter Lake (both the range and the median increases). Temperature is higher in Peter Lake (range and median), and it was sampled throughout the year. Phosphorous and nitrogen both have very wide ranges and narrow IQRs leading to lots of individual outliers, especially in Peter Lake, and temperature also ranges widely as well but seems to have a more normal distribution.

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.

```r
#6

# make plot
plot2 <-
ggplot(subset(litter, functionalGroup == "Needles")) +
  geom_point(aes(x=date, y=dryMass, color=nlcdClass), size = 1.5) +
  scale_x_date(date_breaks = "2 months", date_labels = "%Y-%m") +
  ylab(expression("Date")) +
  xlab(expression("Dry Mass")) +
  labs(color = "Functional Group") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  scale_colour_manual(values = c("mediumorchid", "coral1", "aquamarine2"),
                                 labels = c("Evergreen Forest", "Grassland Herbaceous", "Shrub/Scrub"))

print(plot2)
```



```
#7

# make plot
plot3 <-
ggplot(subset(litter, functionalGroup == "Needles")) +
  geom_point(aes(x=date, y=dryMass, color=nlcdClass), size = 1.5) +
  scale_x_date(date_breaks = "6 months", date_labels = "%Y-%m") +
  facet_wrap(vars(nlcdClass), nrow = 1) +
  ylab(expression("Date")) +
  xlab(expression("Dry Mass")) +
  labs(color = "Functional Group") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
    scale_colour_manual(values = c("mediumorchid", "coral1", "aquamarine2"),
                                 labels = c("Evergreen Forest", "Grassland Herbaceous", "Shrub/Scrub"))

print(plot3)
```



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

Answer: The faceted plot is definitely more effective. You can't see any clear patterns between the different functional groups when all three are plotted on the same axis with different colors (there aren't clear clusters or range differences, just a mess of colors that is very confusing to the eye). However, when they are plotted side by side, you can see differences in range, clustering, and change over time.