Assignment 5: Data Visualization

Megan McClaugherty

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 Getting working directory, loading necessary packages
# and datasets

getwd()
```

```
## [1] "/home/guest/EDA-Fall2022/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
```

```
library(cowplot)
```

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

```
library(RColorBrewer)

LakeChemNutrient <- read.csv("/home/guest/EDA-Fall2022/Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_
    stringsAsFactors = TRUE)

LitterMassTrap <- read.csv("/home/guest/EDA-Fall2022/Data/Processed/NEON_NIWO_Litter_mass_trap_Processe
    stringsAsFactors = TRUE)

# 2 Formatting date

LakeChemNutrient$sampledate <- as.Date(LakeChemNutrient$sampledate,
    format = "%Y-%m-%d")
LitterMassTrap$collectDate <- as.Date(LitterMassTrap$collectDate,
    format = "%Y-%m-%d")
```

Define your theme

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

```
# 3 Building a theme and setting it as default theme to be
# applied across all plots

A5theme <- theme_light(base_size = 12) + theme(axis.text = element_text(color = "black"),
    axis.title.x = element_text(color = "black"), legend.position = "bottom")

theme_set(A5theme)
```

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 Using a scatterplot with a trend line to visualize the
# relationship between total phosphorus and phosphate
# concentrations, using different colors for the two lakes.
# Adjusted y-axis to remove an outlier.
PhosphatePhosphorus <- ggplot(LakeChemNutrient, aes(x = tp_ug,
    y = po4, color = lakename)) + geom_point() + geom_smooth(method = lm,
    color = "black") + ylim(0, 45) + xlim(0, 150) + xlab("Total Phosphorus Concentration (ug/L)") +
    ylab("Phosphate (ug/L)")
```

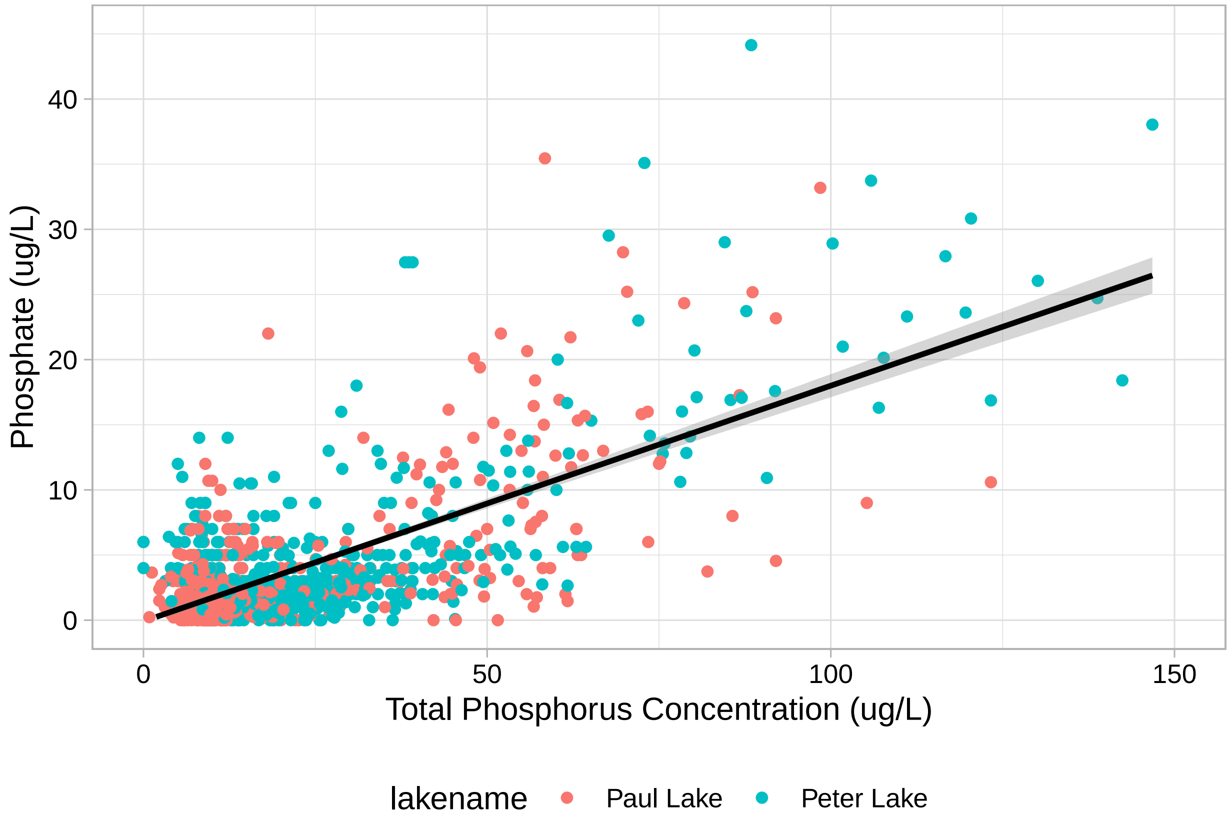
```
print(PhosphatePhosphorus)
```

```
## `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 1 rows containing missing values (geom_smooth).
```



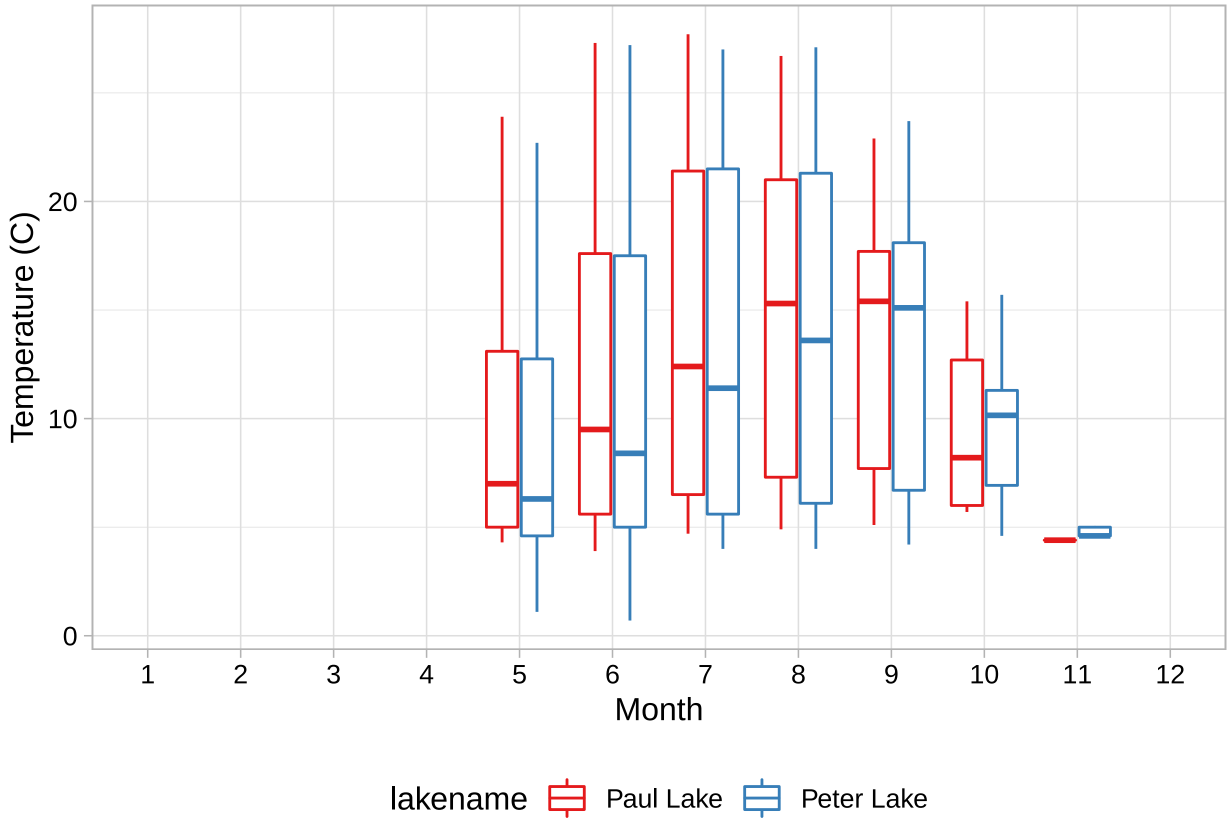
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

(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.

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

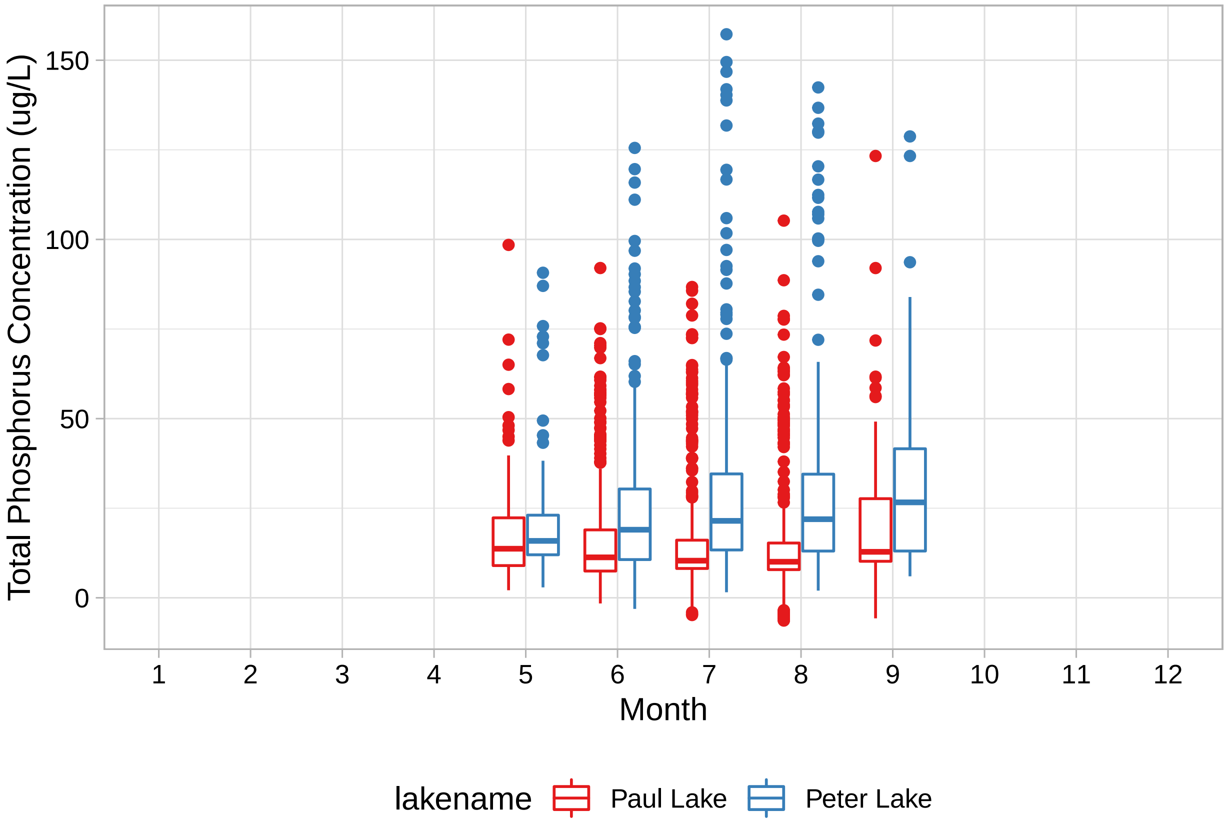
```
# 5 Creating separate boxplots for temperature, total
# phosphorus, and total nitrogen by month for each lake.

LakeTempbyMonth <- ggplot(LakeChemNutrient, aes(x = factor(month,
    levels = c(1:12)), y = temperature_C, color = lakename)) +
    geom_boxplot() + xlab("Month") + ylab("Temperature (C)") +
    scale_x_discrete(drop = FALSE) + scale_color_brewer(palette = "Set1")
print(LakeTempbyMonth)
```

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



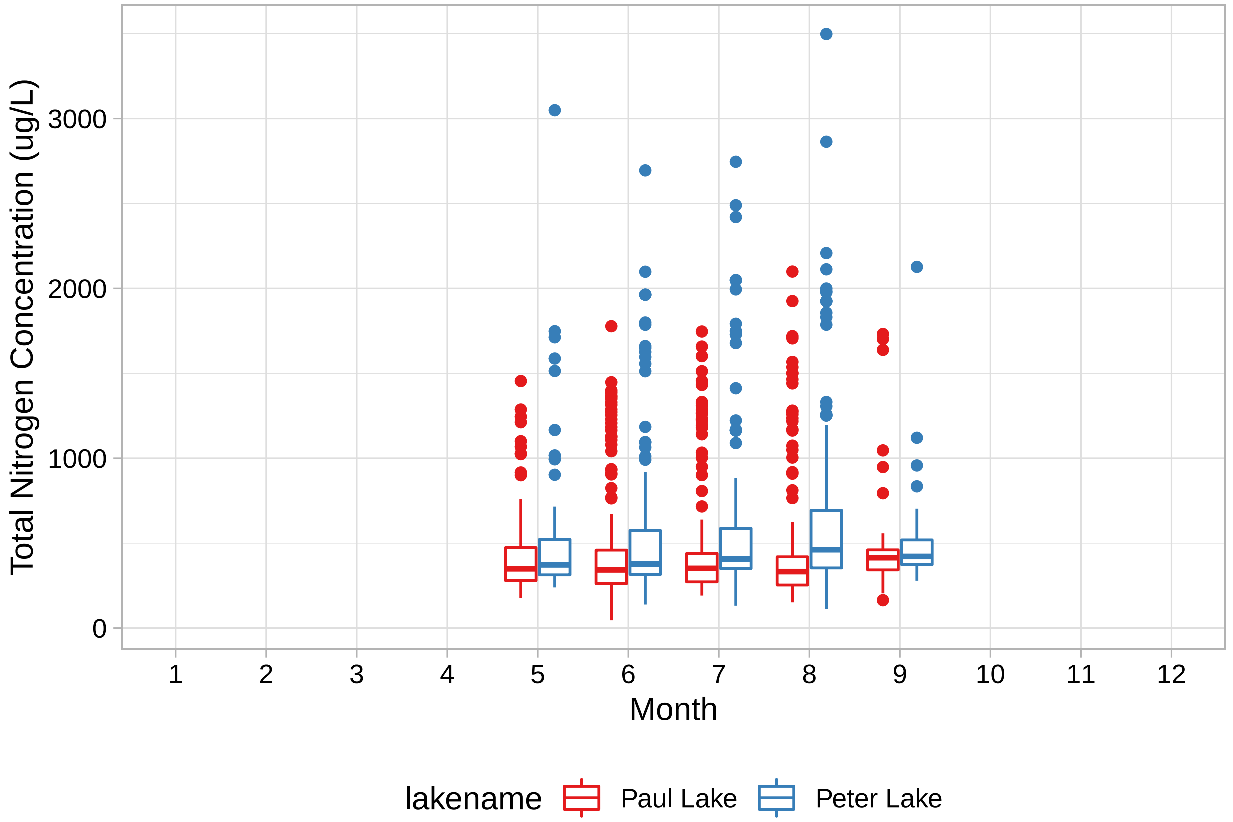
```
LakeTPbyMonth <- ggplot(LakeChemNutrient, aes(x = factor(month,
    c(1:12)), y = tp_ug, color = lakename)) + geom_boxplot() +
    xlab("Month") + ylab("Total Phosphorus Concentration (ug/L)") +
    scale_color_brewer(palette = "Set1") + scale_x_discrete(drop = FALSE)

print(LakeTPbyMonth)
```

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



```r
LakeTNbyMonth <- ggplot(LakeChemNutrient, aes(x = factor(month,
    c(1:12)), y = tn_ug, color = lakename)) + geom_boxplot() +
    xlab("Month") + ylab("Total Nitrogen Concentration (ug/L)") +
    scale_color_brewer(palette = "Set1") + scale_x_discrete(drop = FALSE)

print(LakeTNbyMonth)
```

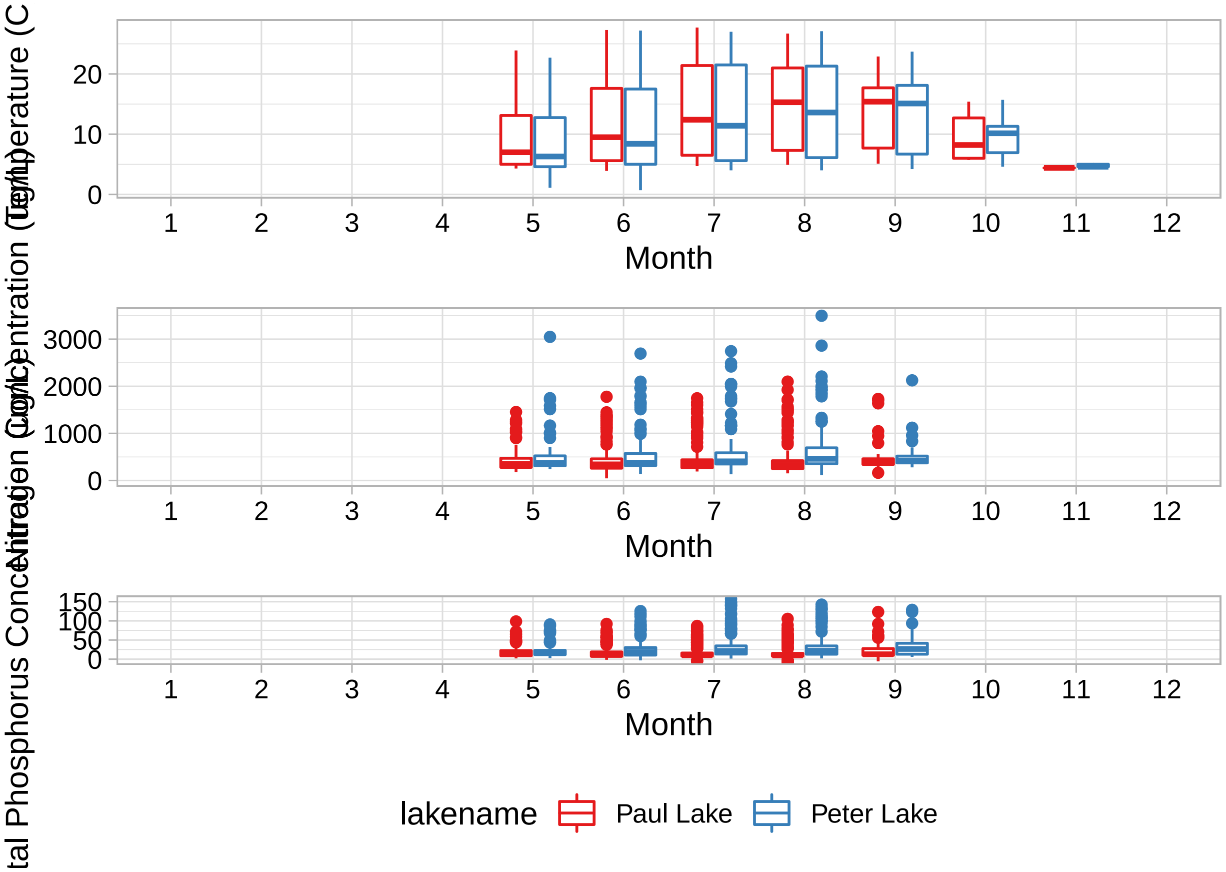
```
## Warning: Removed 21583 rows containing non-finite values (stat_boxplot).
```



```
# cow plot to combine all three graphs into one output

LakeTempTPTNbyMonth <- plot_grid(LakeTempbyMonth + theme(legend.position = "none"),
    LakeTNbyMonth + theme(legend.position = "none"), LakeTPbyMonth +
        theme(legend.position = "bottom"), nrow = 3, align = "v",
    rel_heights = c(2, 2, 2))
```

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

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

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

```
print(LakeTempTPTNbyMonth)
```

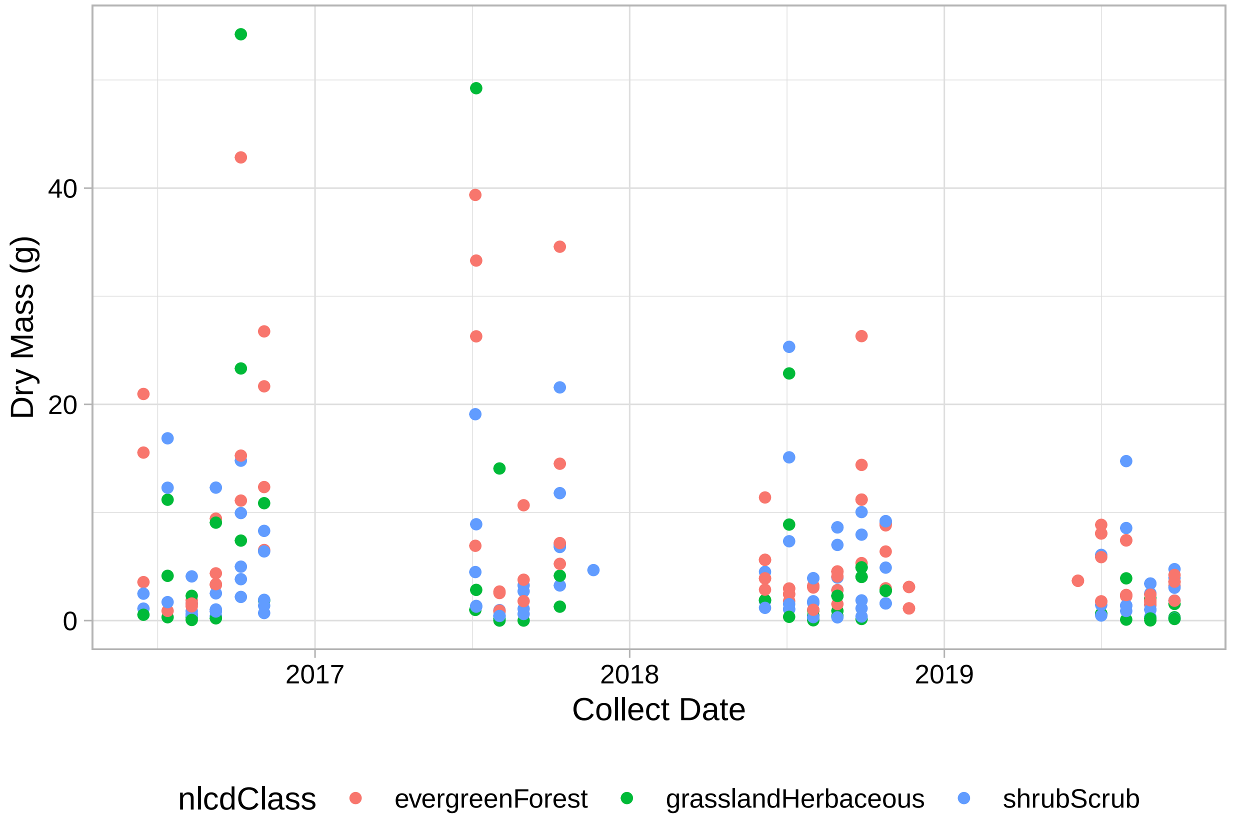


Question: What do you observe about the variables of interest over seasons and between lakes?

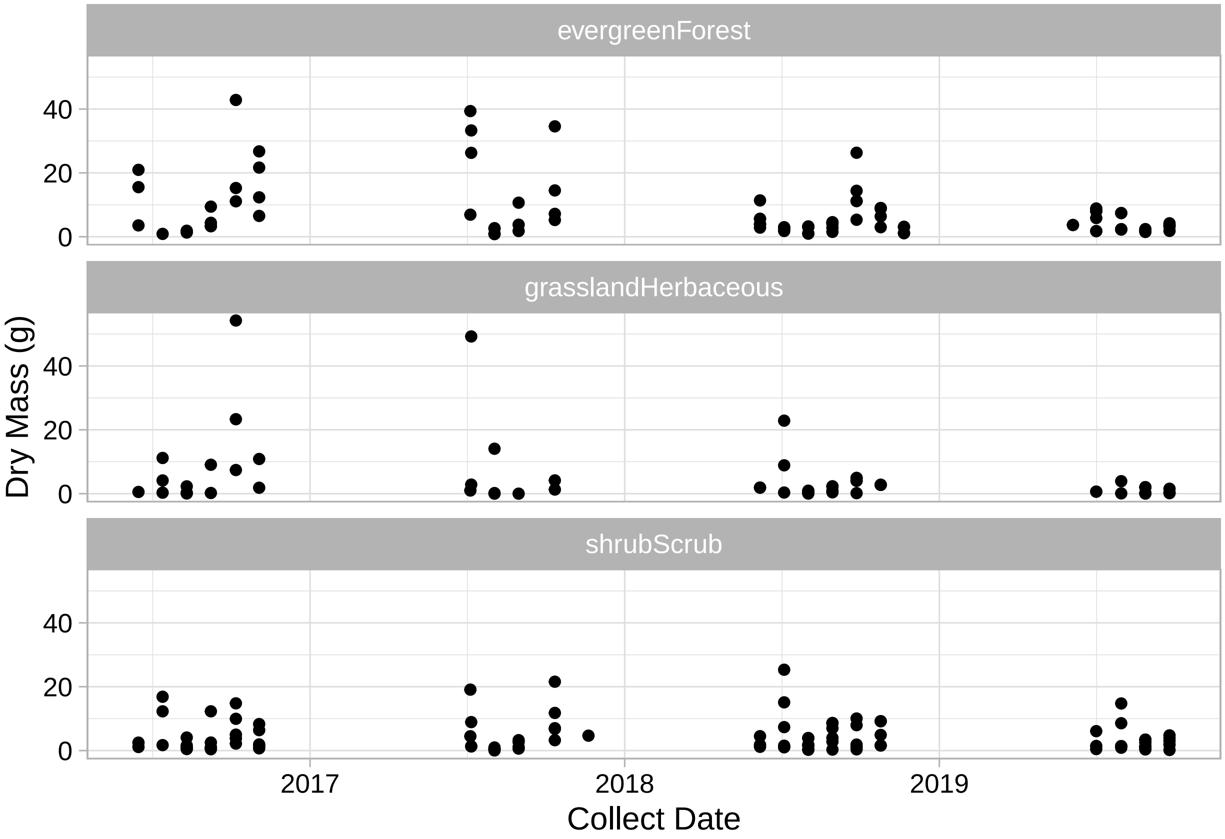
> Answer: In the warmer months, especially July and August, the range and median temperature increases in both lakes compared to cooler months, with the highest median temperature occuring in September. For months with available data, the median temperature in Paul Lake is always slightly higher than Peter Lake except in October and November. For total phosphorus, the range and median increases relatively gradually from May through September in Peter Lake. However, in Paul Lake, the range and median decrease from May through July, then appears to stabilize before increeasing again in September. The median total phosphorus in Paul Lake is consistently lower than that in Peter Lake. While both have numerous high value, there is a broader distribution of outliers 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
NeedleMass <- ggplot(subset(LitterMassTrap, functionalGroup ==
    "Needles"), aes(x = collectDate, y = dryMass, color = nlcdClass)) +
    xlab("Collect Date") + ylab("Dry Mass (g)") + geom_point()

print(NeedleMass)
```



```
# 7
NeedleMassFacet <- ggplot(subset(LitterMassTrap, functionalGroup ==
    "Needles"), aes(x = collectDate, y = dryMass)) + xlab("Collect Date") +
    ylab("Dry Mass (g)") + geom_point() + facet_wrap(vars(nlcdClass),
    nrow = 3)
print(NeedleMassFacet)
```



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

Answer: Plot 7 seems more effective because it allows you to see a more comprehensive distribution of each NLCD class over time and easily make comparisons among the classes. In plot 6, there is a lot of overlap in the data, so even with color it is difficult to observe patterns or trends within or between the classes.