Assignment 5: Data Visualization

Camila Zarate Ospina

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

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

```
#1 Set up
# Verify working directory and load packages
getwd()
```

```
## [1] "/Users/camilazarate/OneDrive - Duke University/2 Second semester/Data analytics/Environmental_Da
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

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

```
library(cowplot)

# Load data
PeterPaul <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", stri
PeterPaul.gathered <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.cs
Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                   stringsAsFactors = TRUE)

# Change dates
class(PeterPaul$sampledate)
```

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

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

class(PeterPaul.gathered$sampledate)
```

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

```
PeterPaul.gathered$sampledate <- as.Date(PeterPaul.gathered$sampledate, format = "%Y-%m-%d")

class(Litter$collectDate)
```

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

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

Define your theme

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

```
my.theme <- theme_grey(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(my.theme)
```

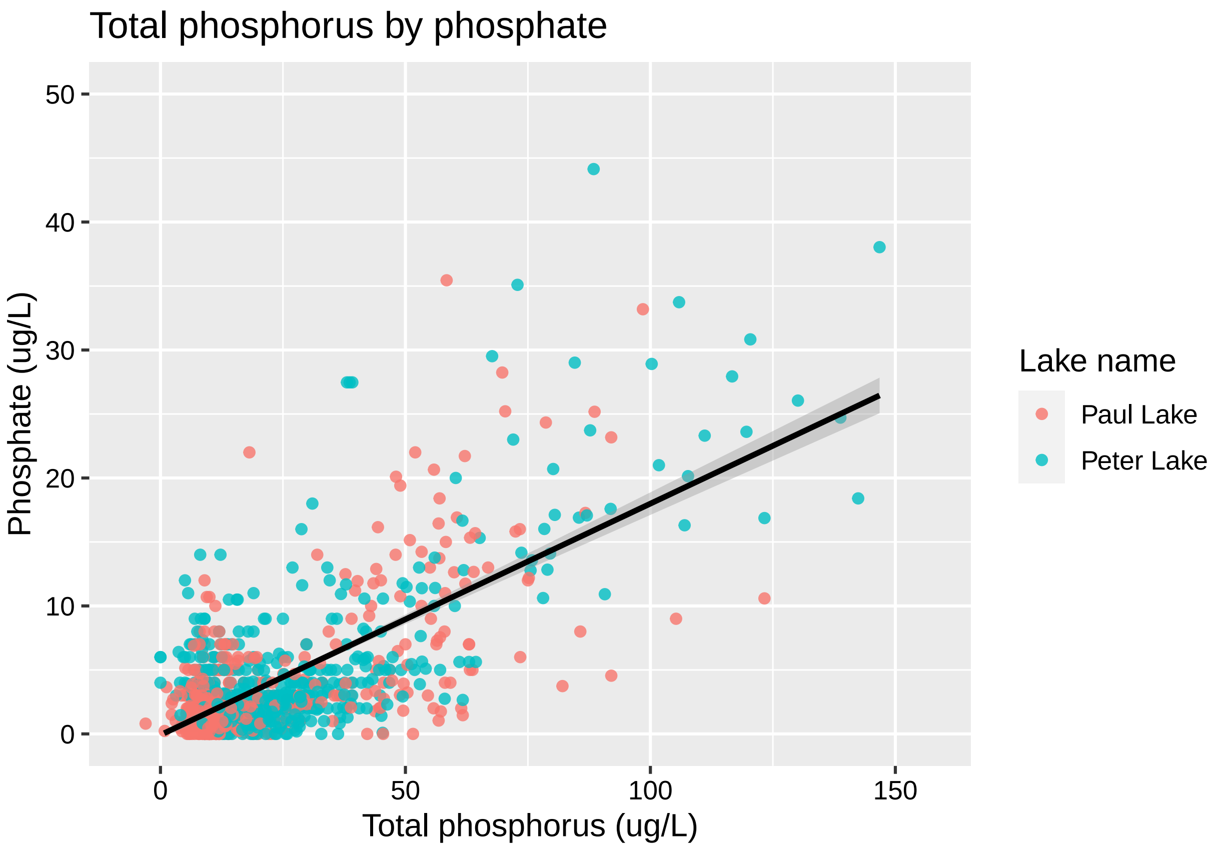
Create graphs

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.

```
tp_by_po4 <- ggplot(PeterPaul, aes(x = tp_ug, y = po4, color = lakename),
                    shape = lakename) +
  geom_point(alpha = 0.8, size = 1.5) +
  labs(title = "Total phosphorus by phosphate", color = "Lake name",
       y = "Phosphate (ug/L)", x = "Total phosphorus (ug/L)") +
  ylim(0, 50) +
  geom_smooth(method = lm, color = "black")
print(tp_by_po4)
```

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



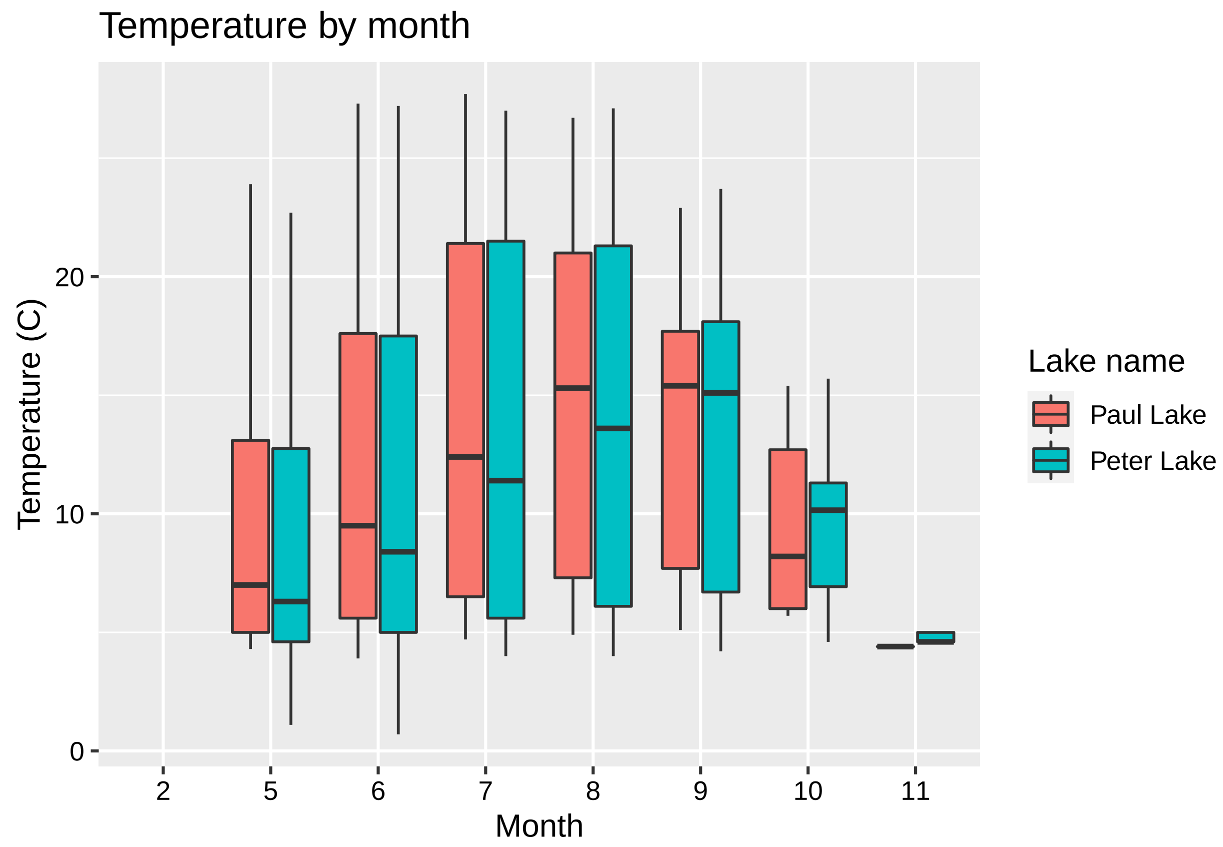
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

```
# Change month to factor so it spaces out the months per set of boxplots.
PeterPaul$month <- as.factor(PeterPaul$month)

# Temperature
# Fill inside aes to show the legend.
boxplot.temp <- ggplot(PeterPaul, aes(x = month, y = temperature_C,
                                  fill = lakename)) +
  geom_boxplot() +
  labs (title = "Temperature by month", fill = "Lake name",
        y = "Temperature (C)", x = "Month")
  scale_fill_brewer(palette = "Dark2")
```

```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: fill
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##         reset: function
##         train: function
##         super:  <ggproto object: Class RangeDiscrete, Range, gg>
##     rescale: function
##     reset: function
##     scale_name: brewer
##     train: function
##     train_df: function
##     transform: function
##     transform_df: function
##     super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

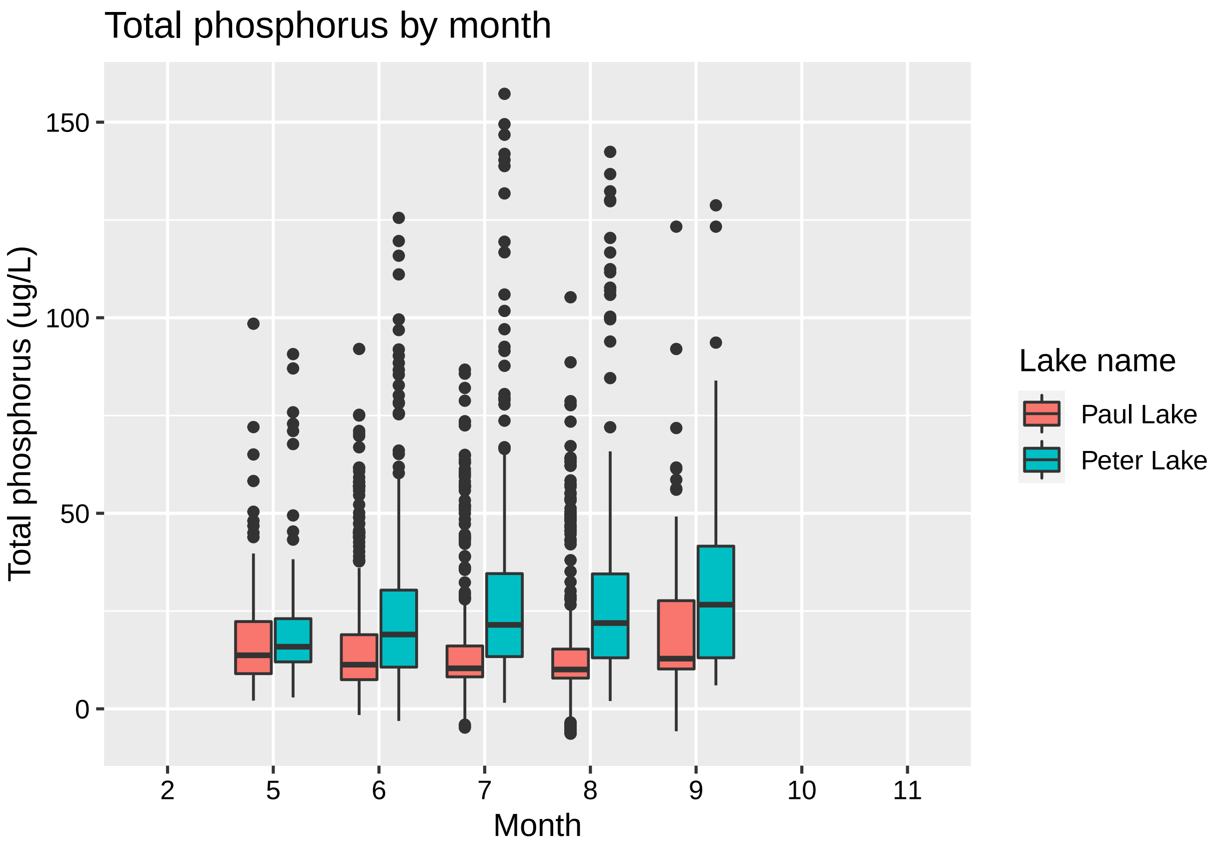
```r
print(boxplot.temp)
```



```
# TP
boxplot.tp <- ggplot(PeterPaul, aes(x = month, y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  labs (title = "Total phosphorus by month", fill = "Lake name",
        y = "Total phosphorus (ug/L)", x = "Month")
  scale_fill_brewer(palette = "Dark2")
```

```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: fill
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##         reset: function
##         train: function
##         super:  <ggproto object: Class RangeDiscrete, Range, gg>
##     rescale: function
##     reset: function
##     scale_name: brewer
##     train: function
##     train_df: function
##     transform: function
##     transform_df: function
##     super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

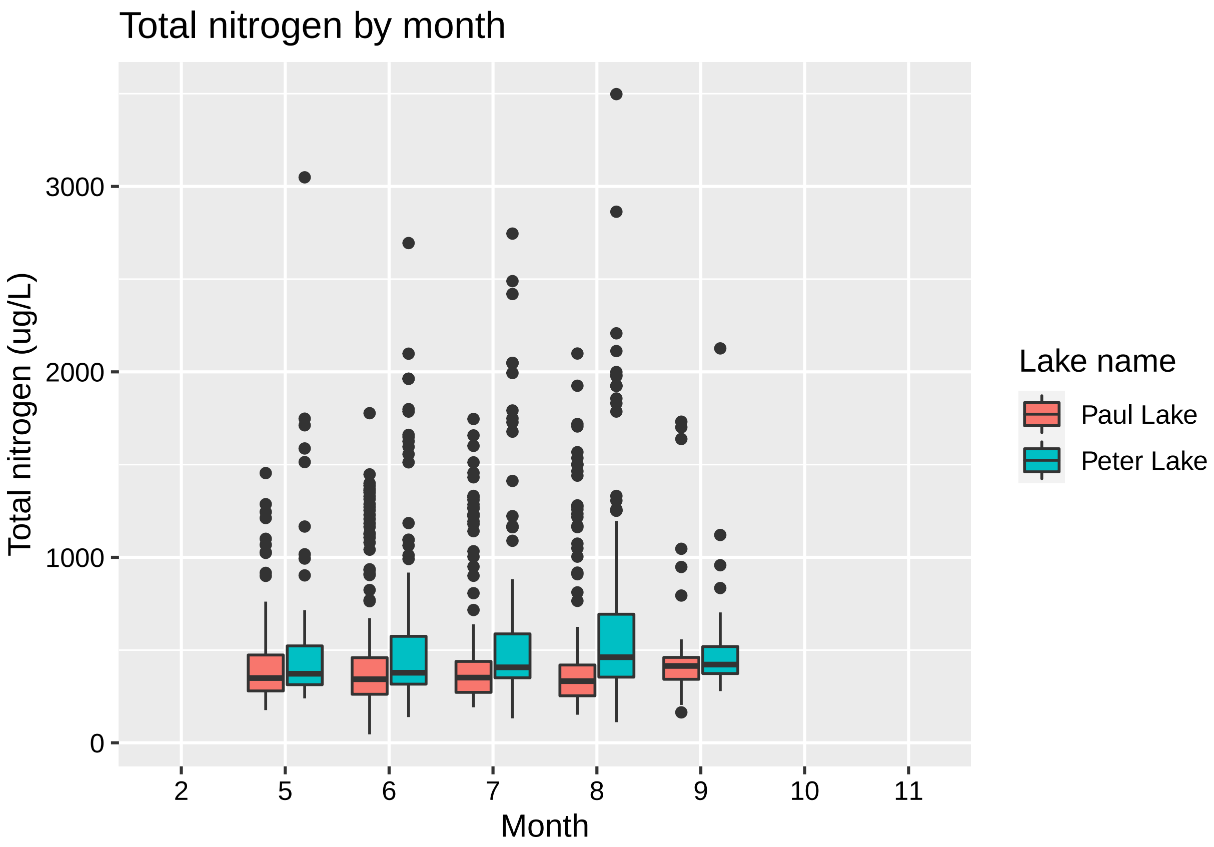
```
print(boxplot.tp)
```



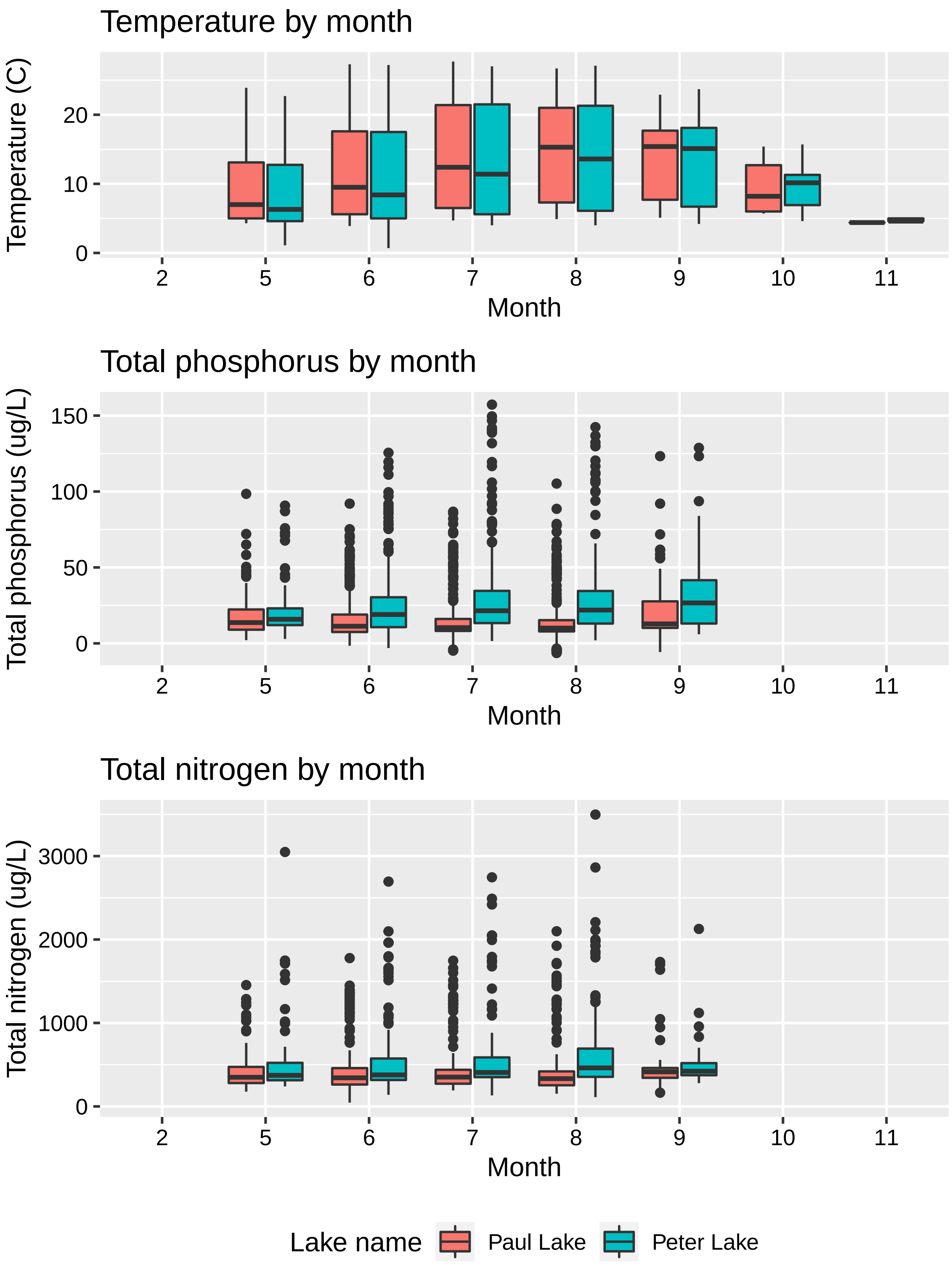
```
# TN
boxplot.tn <- ggplot(PeterPaul, aes(x = month, y = tn_ug, fill = lakename)) +
  geom_boxplot() +
  labs (title = "Total nitrogen by month", fill = "Lake name",
        y = "Total nitrogen (ug/L)", x = "Month")
  scale_fill_brewer(palette = "Dark2")
```

```
## <ggproto object: Class ScaleDiscrete, Scale, gg>
##     aesthetics: fill
##     axis_order: function
##     break_info: function
##     break_positions: function
##     breaks: waiver
##     call: call
##     clone: function
##     dimension: function
##     drop: TRUE
##     expand: waiver
##     get_breaks: function
##     get_breaks_minor: function
##     get_labels: function
##     get_limits: function
##     guide: legend
##     is_discrete: function
##     is_empty: function
##     labels: waiver
##     limits: NULL
##     make_sec_title: function
##     make_title: function
##     map: function
##     map_df: function
##     n.breaks.cache: NULL
##     na.translate: TRUE
##     na.value: NA
##     name: waiver
##     palette: function
##     palette.cache: NULL
##     position: left
##     range: <ggproto object: Class RangeDiscrete, Range, gg>
##         range: NULL
##         reset: function
##         train: function
##         super:  <ggproto object: Class RangeDiscrete, Range, gg>
##     rescale: function
##     reset: function
##     scale_name: brewer
##     train: function
##     train_df: function
##     transform: function
##     transform_df: function
##     super:  <ggproto object: Class ScaleDiscrete, Scale, gg>
```

```
print(boxplot.tn)
```



```
# Combination of plots
combined <- plot_grid(
  boxplot.temp + theme(legend.position = "none"),
  boxplot.tp + theme(legend.position = "none"),
  boxplot.tn + theme(legend.position = "bottom"),
  ncol = 1, align = 'hv', rel_widths = c(1.5,1.5,1.5), rel_heights = c(2.5,3,4))
print(combined)
```



Temperature by month

Total phosphorus by month

Total nitrogen by month

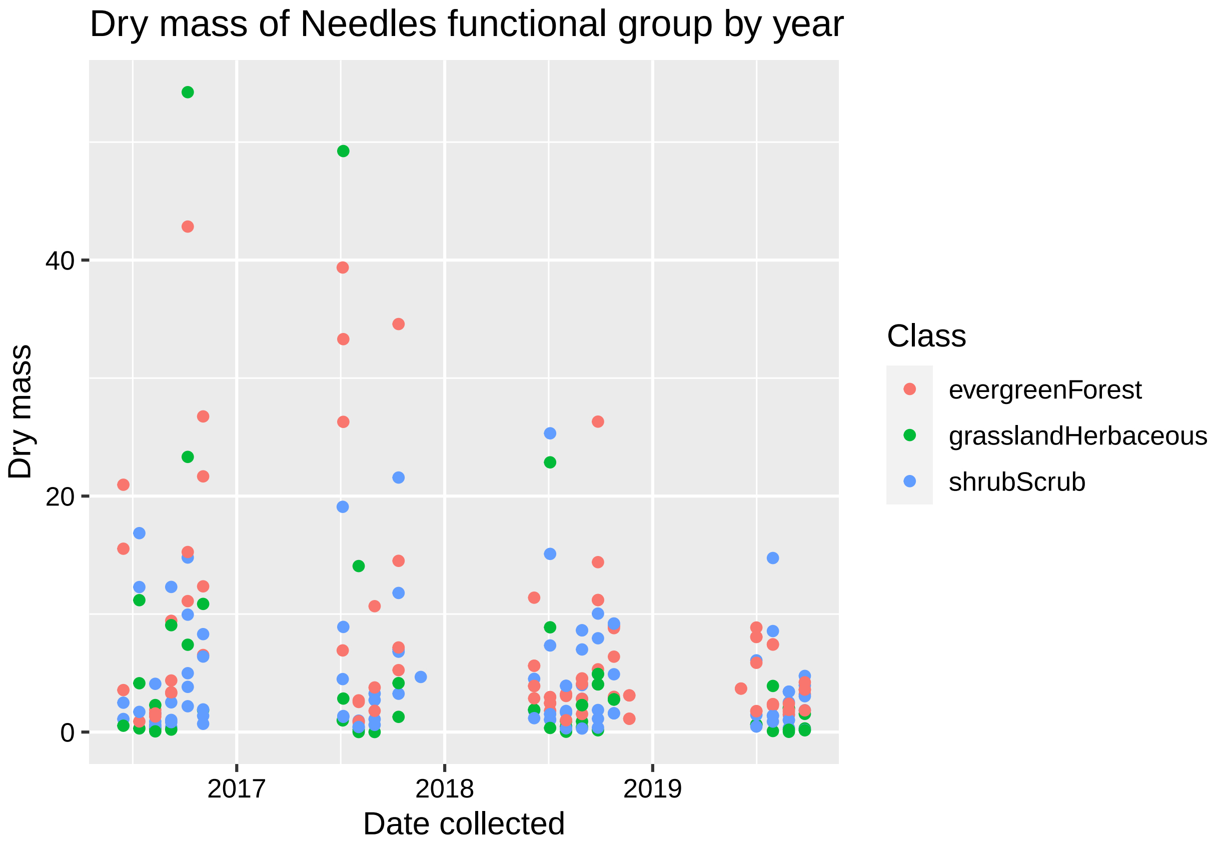
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperatures in both lakes are similar, presenting higher temperatures between months 7 and 8 (July and August), which correpond to the summer months. Overall, Peter Lake has higher values of total phosphorus and total nitrogen across the months. Regarding total phosphorus, Paul Lake levels tends to decrease over time, while Peter Lake levels increase. Even when all months present outliers with large concentration of total phosphorus, these outliers are larger

during the summer months. Likewise, regarding total nitrogen, Paul Lake levels tends to decrease over time, while Peter Lake levels increase.

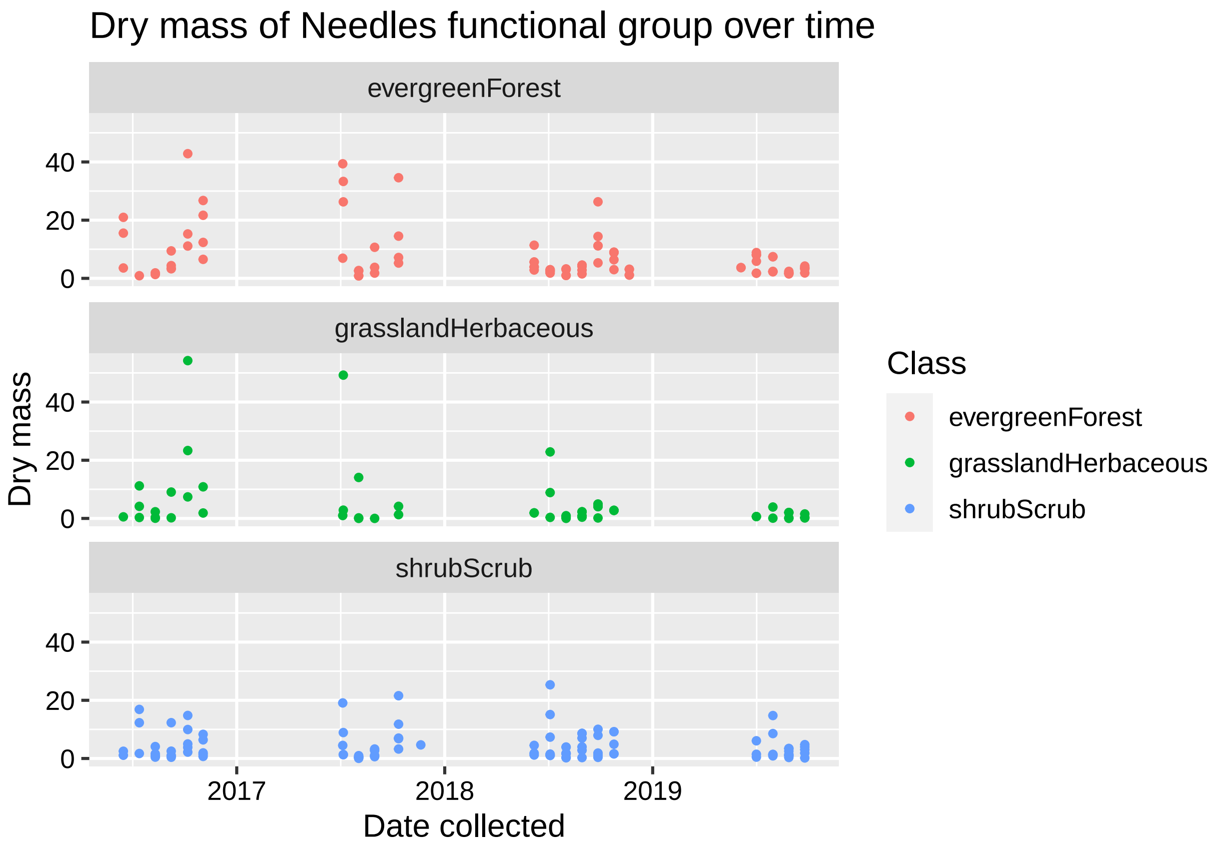
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

```r
# Subset of litter dataset
needles.drymass <- ggplot(subset(Litter, functionalGroup == "Needles"),
                  aes(y = dryMass, x = collectDate, color = nlcdClass)) +
  geom_point() +
  labs(title = "Dry mass of Needles functional group by year",
       x = "Date collected", y = "Dry mass", color = "Class")
print(needles.drymass)
```



7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```r
needles <- ggplot(subset(Litter, functionalGroup == "Needles"),
                  aes(y = dryMass, x = collectDate, color = nlcdClass)) +
  geom_point(size = 1) +
  labs(title = "Dry mass of Needles functional group over time",
       x = "Date collected", y = "Dry mass", color = "Class") +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(needles)
```



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

Answer: Plot 7 (facet) is more effective, because it allows us to clearly see the data by class over the months, while in Plot 6 many data points overlap with others and it's more difficult to find patterns.