SimonHeinberg_A05_Data Visualization

Simon Heinberg

Fall 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.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 your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

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

```
#1
library(tidyverse);library(lubridate);library(here);library(ggthemes)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
## here() starts at /Users/simonheinberg/Desktop/Duke_Coursework/Envirnmental_Data_Exploration/EDE_Fall
```

```r
library(cowplot)
```

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

```r
getwd()
```

```
## [1] "/Users/simonheinberg/Desktop/Duke_Coursework/Envirnmental_Data_Exploration/EDE_Fall2023"
```

```r
#2
 PeterPaul <-
   read.csv("./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", stringsAsF
 NeonLitter <-
   read.csv("./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = T)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```r
#3
Simon_theme <- theme_base() +
  theme(
    legend.key = element_rect(
      color='purple',

    ),
    plot.background = element_rect(
      color='blue',
      fill = 'grey'
    ),
    plot.title = element_text(
      color='blue'
    ),
    panel.grid.major = element_line(color="grey44")
    )
```

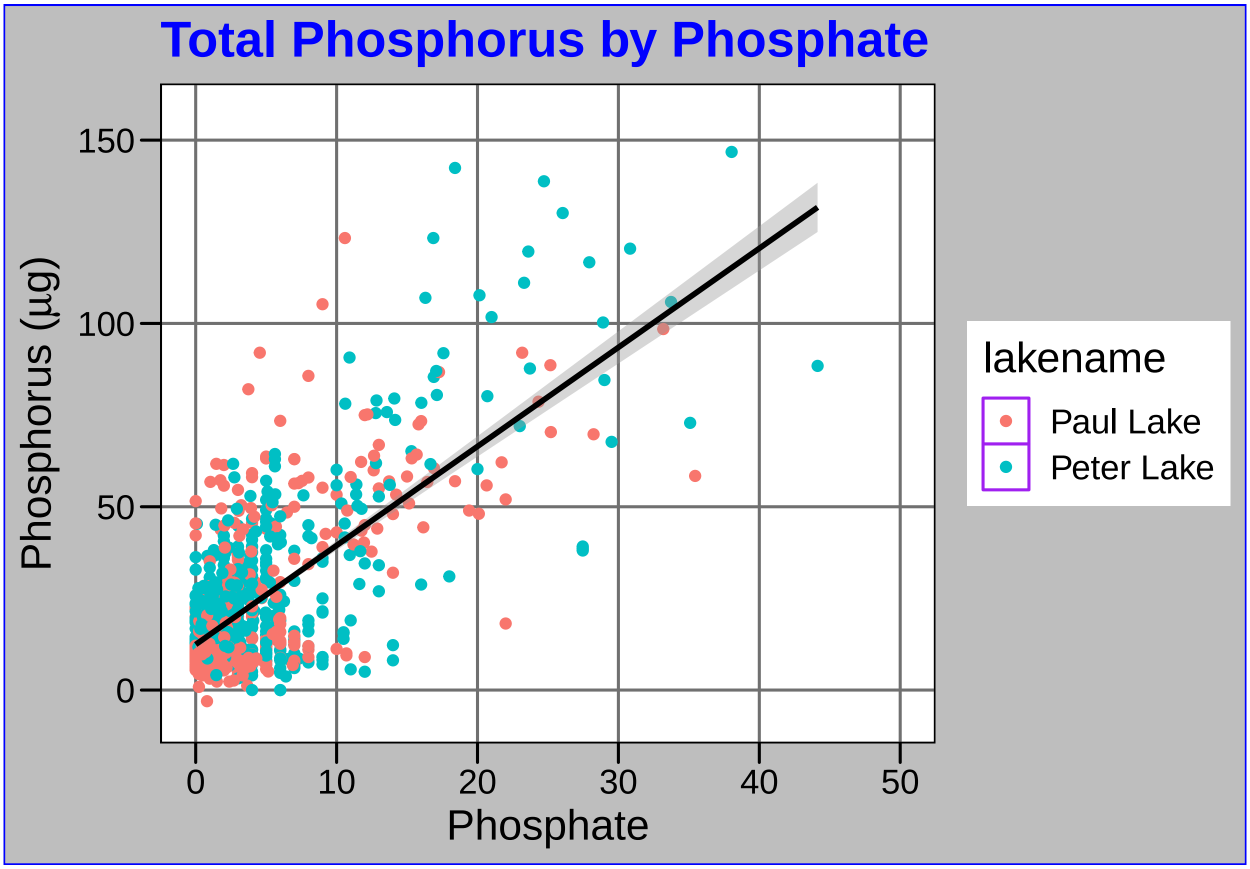
Create graphs

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

Phosphorus_phosphate <-ggplot(PeterPaul, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", color="black") +
  xlim(0, 50) +#this cuts out one outlier
xlab("Phosphate") +
ylab(expression(paste("Phosphorus (", mu, "g)"))) +
ggtitle("Total Phosphorus by Phosphate") +
  Simon_theme

print(Phosphorus_phosphate)
```

```
## `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()`).
```

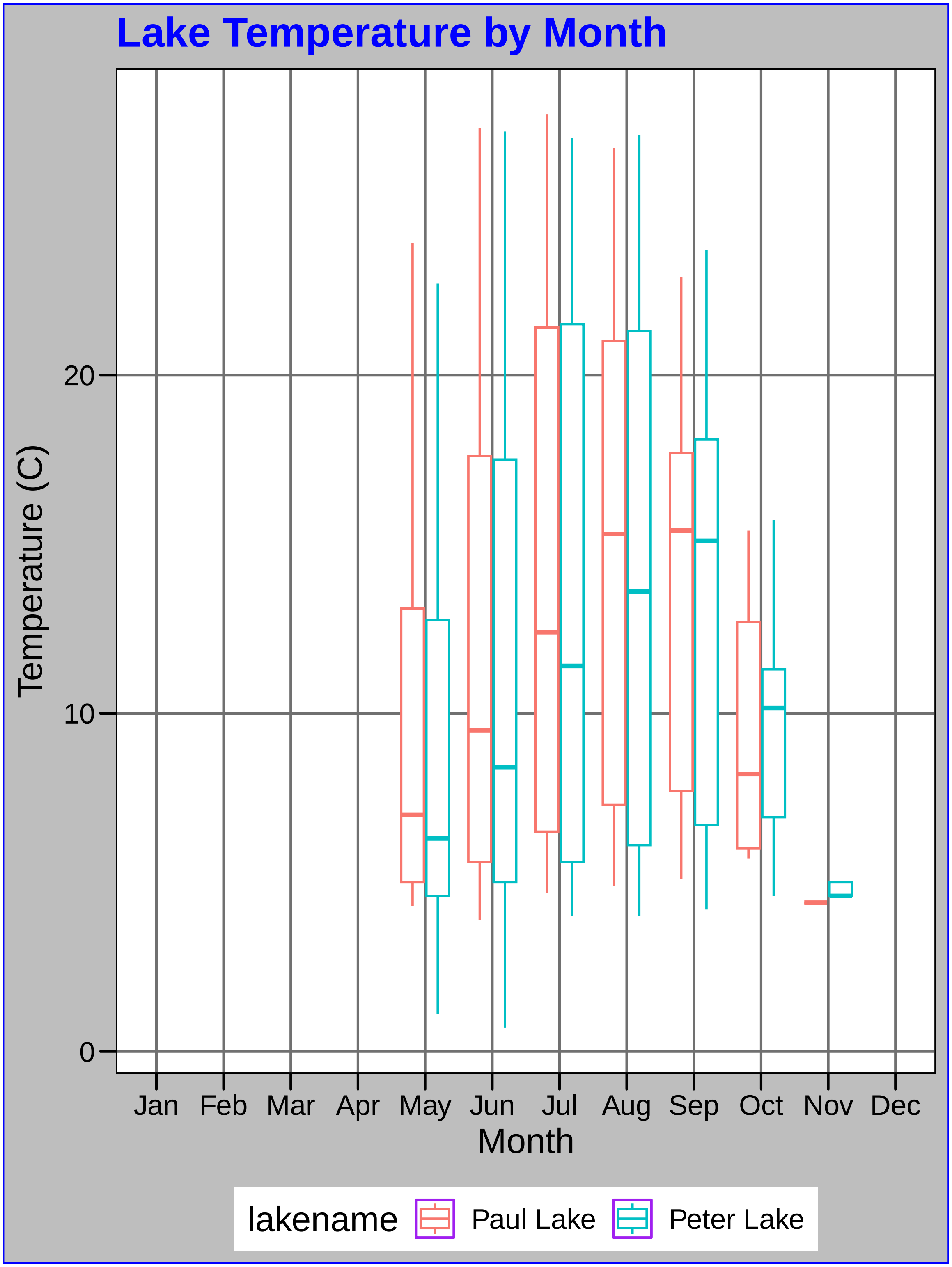


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
temp_Boxplot <-
  ggplot(PeterPaul, aes(x=factor(month,
                                   levels=1:12,
                                   labels=month.abb),
                         y = temperature_C, color = lakename)) +
  geom_boxplot()+
  scale_x_discrete(limits = month.abb, name = "Month")+
  ylab(expression(paste("Temperature (C)")))+
ggtitle("Lake Temperature by Month") +
  Simon_theme+
  theme(legend.position="bottom")


  temp_Boxplot
```

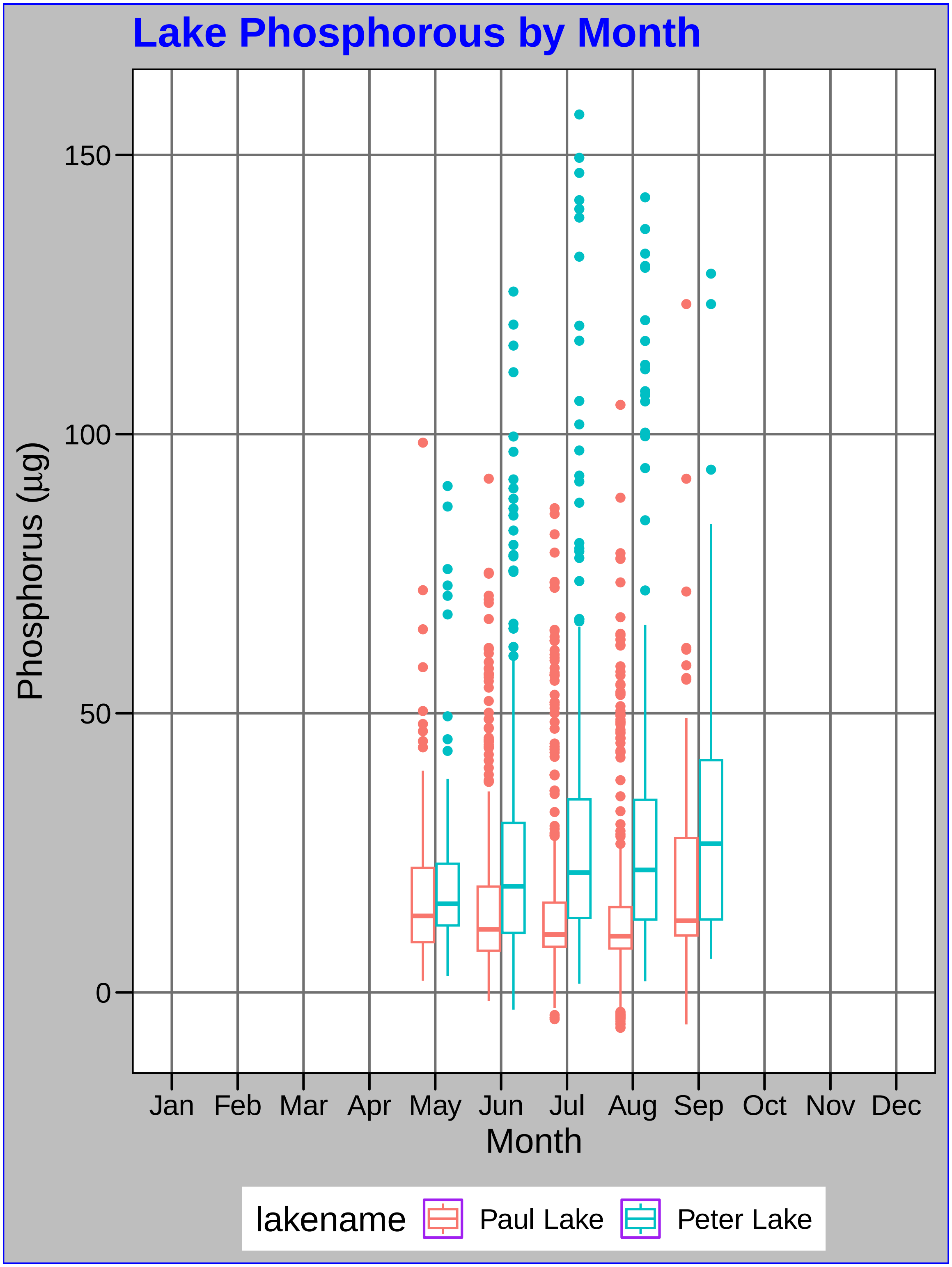
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```




```
TP_BoxPlot <-
  ggplot(PeterPaul, aes(x=factor(month,
                                   levels=1:12,
                                   labels=month.abb),
                          y = tp_ug, color = lakename)) +
```

```
  geom_boxplot()+
  scale_x_discrete(limits = month.abb, name = "Month")+
  ylab(expression(paste("Phosphorus (", mu, "g)")))+
ggtitle("Lake Phosphorous by Month") +
    Simon_theme+
  theme(legend.position="bottom")

TP_BoxPlot
```

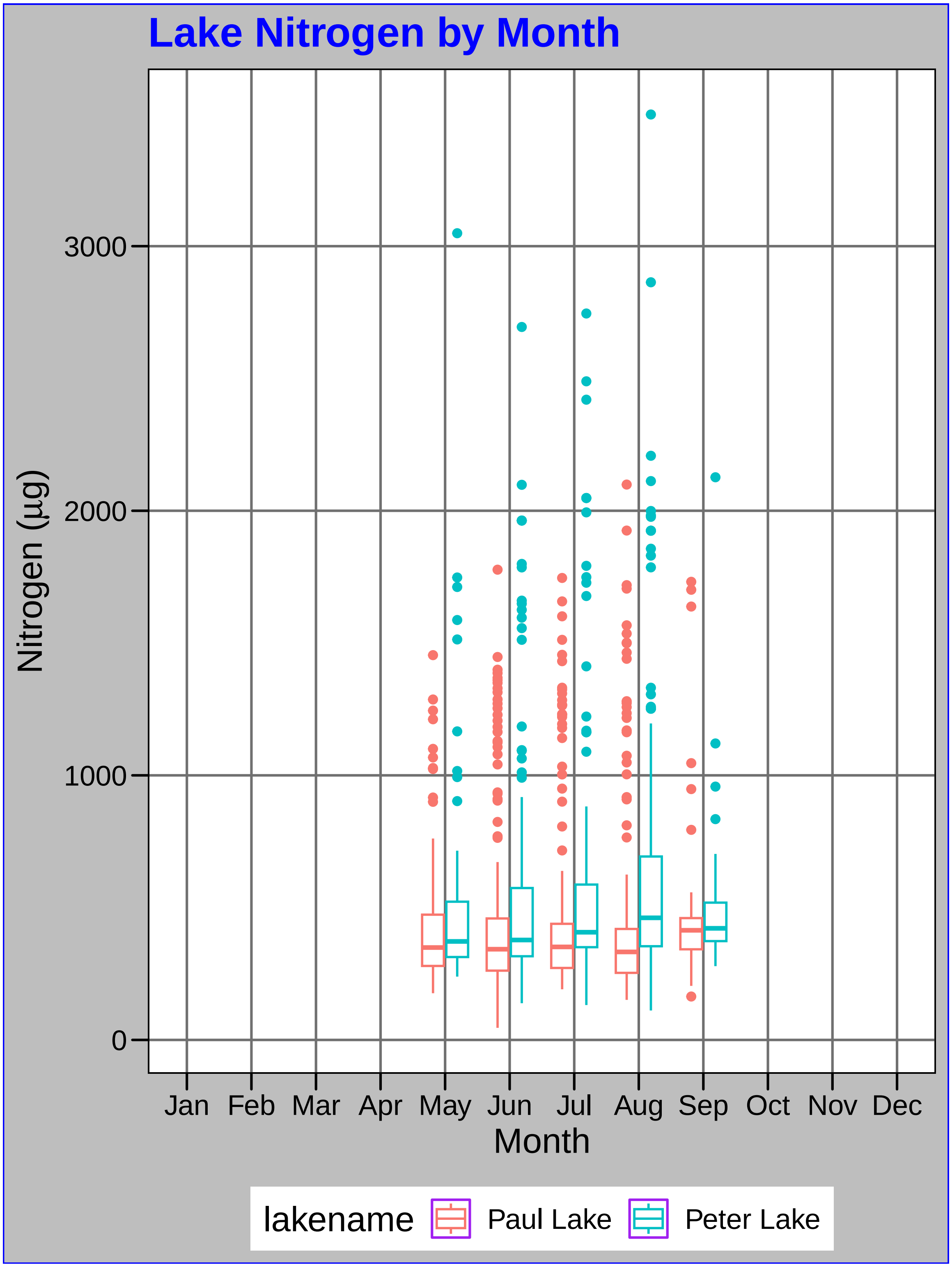
```
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```




```
TN_Boxplot <-
ggplot(PeterPaul, aes(x=factor(month,
                                levels=1:12,
                                labels=month.abb),
                      y = tn_ug, color = lakename)) +
```

```r
  geom_boxplot()+
  scale_x_discrete(limits = month.abb, name = "Month")+
  ylab(expression(paste("Nitrogen (", mu, "g)")))+
ggtitle("Lake Nitrogen by Month") +
 Simon_theme+
  theme(legend.position="bottom")

TN_Boxplot
```

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

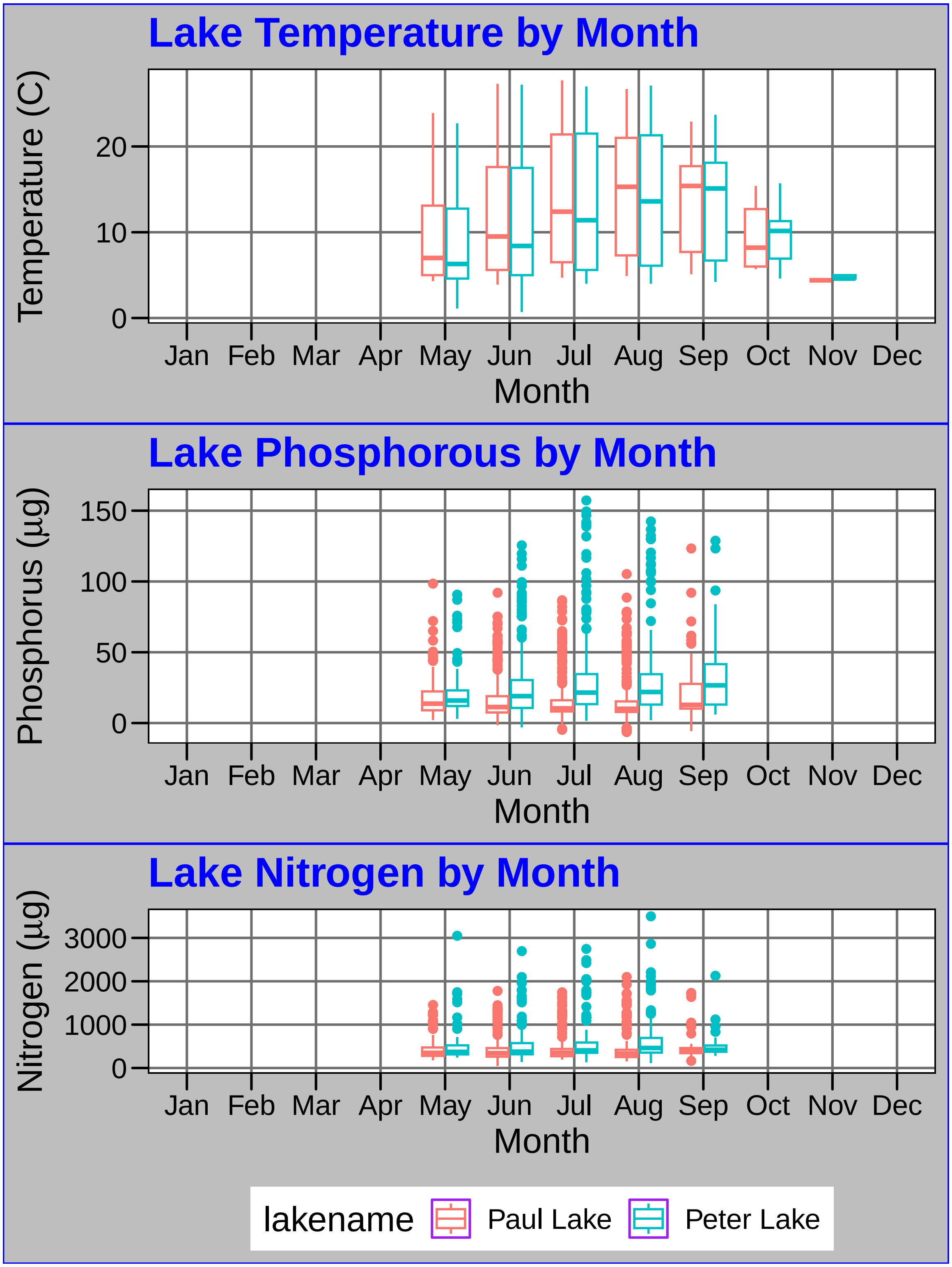



```
#Combining the three box plots using a cowplot

combined_plot<-plot_grid(temp_Boxplot+theme(legend.position="none"),
                         TP_BoxPlot+theme(legend.position="none"),
                         TN_Boxplot+theme(legend.position="bottom"),
```

```
                         ncol=1,
                         align="vh")
```

```
## 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()‘).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
print(combined_plot)
```



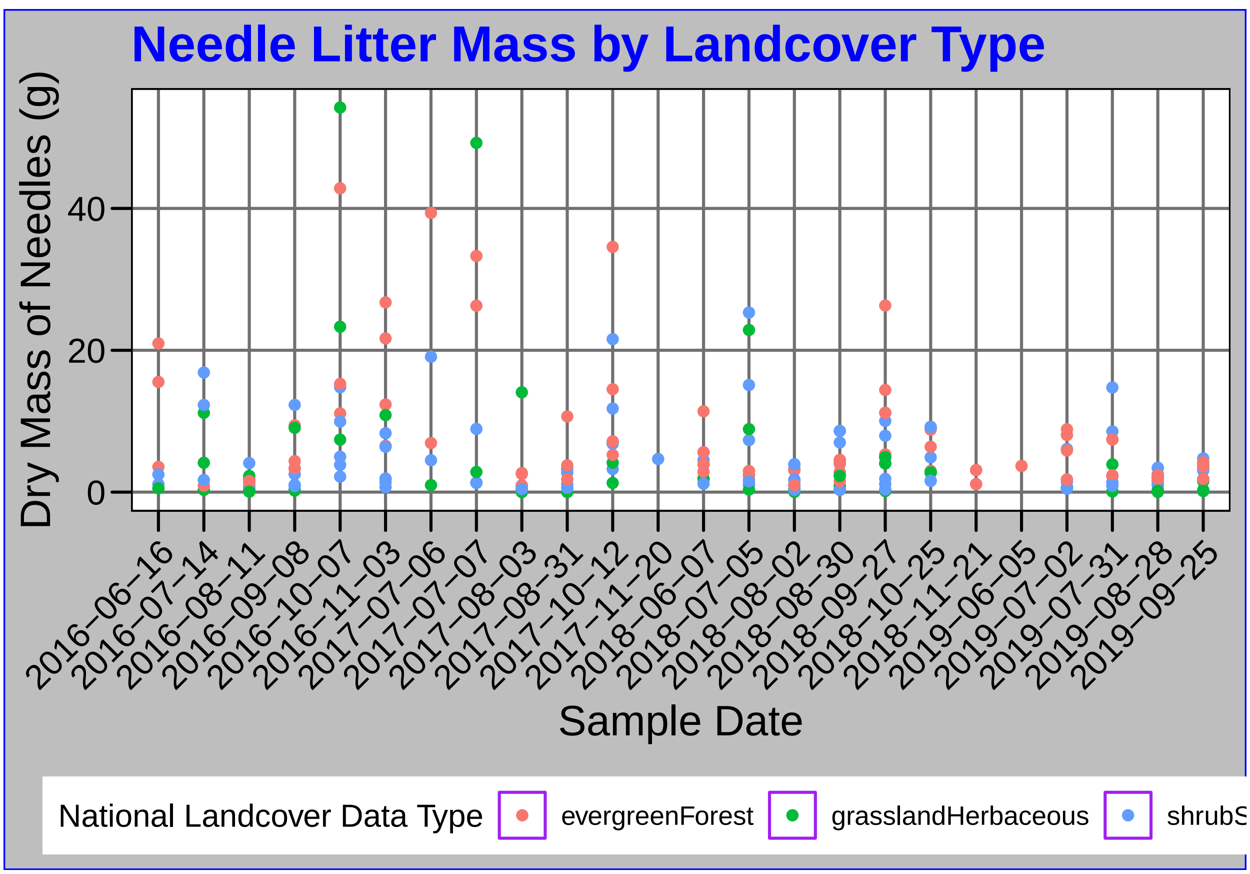
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Peter lake has a slower rate of temperature change, which suggests that Peter lake is deeper than Paul lake. Nitrogen and phosphorous levels also seem to rise in Peter lake over the course of the summer. This may be because intensive agriculture tends to occur in warmer

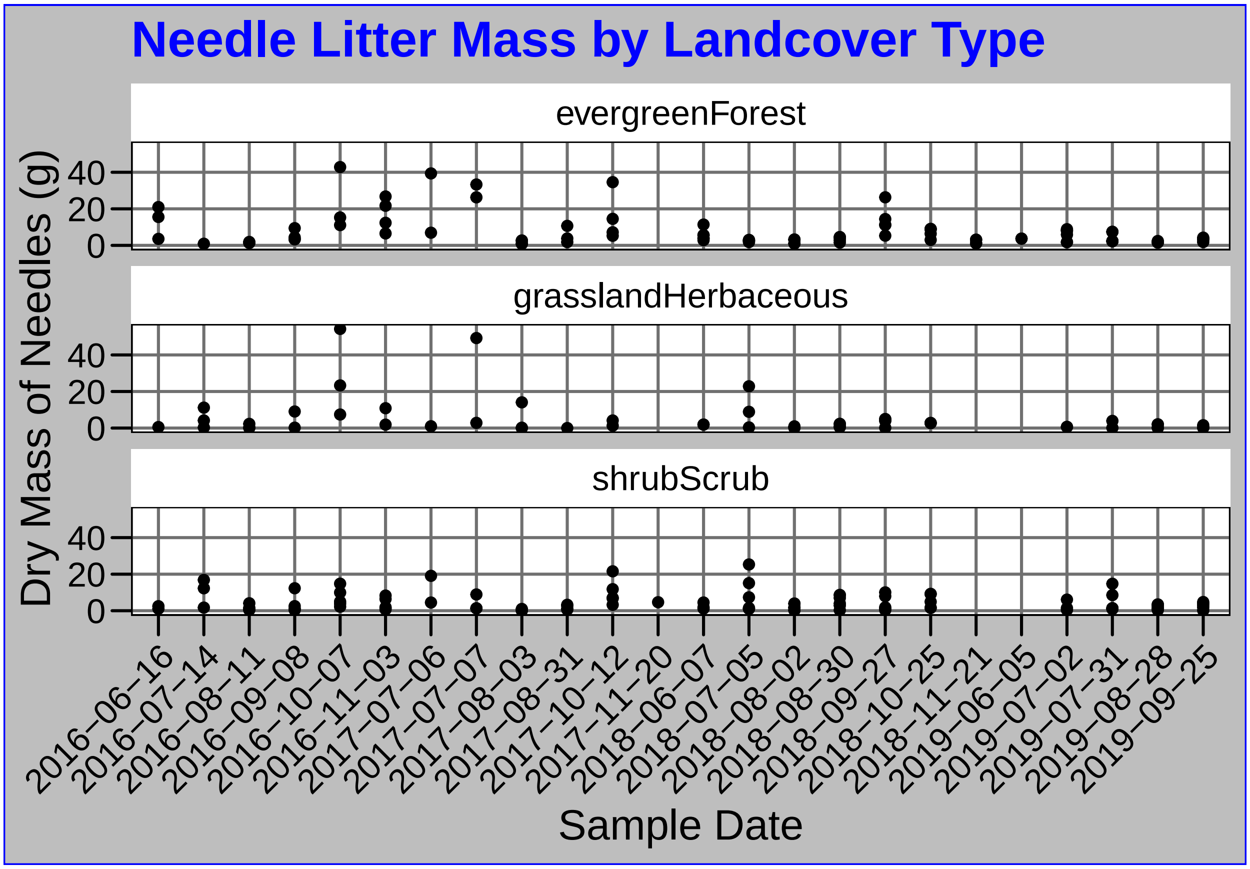
months, and nitrogen and phosphorous are pollutants of agricultural runoff. The nitrogren and phosphorous concentrations are indicators that over the summer months, nearby agriculture is causing more pollution into the watershed.

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

Needles_month <- ggplot(filter(NeonLitter, functionalGroup=="Needles"),
                              aes(x=factor(collectDate),
                              y = dryMass, color=nlcdClass)) +
  geom_point()+
  xlab("Sample Date")+
  ylab("Dry Mass of Needles (g)")+
  Simon_theme+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 12), legend.position="bottom")+
  ggtitle("Needle Litter Mass by Landcover Type")+
  labs(color="National Landcover Data Type")

Needles_month
```



```r
#7
Needles_month_faceted <- ggplot(filter(NeonLitter, functionalGroup=="Needles"),
                                aes(x=factor(collectDate), y = dryMass)) +
  geom_point()+
  xlab("Sample Date")+
  ylab("Dry Mass of Needles (g)")+
  Simon_theme+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))+
facet_wrap(vars(nlcdClass), nrow = 3)+
  ggtitle("Needle Litter Mass by Landcover Type")

Needles_month_faceted
```



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

> Answer: The second plot is more effective since it provides a more clear visualization of how needle mass varies by landcover type and over time, whereas the first graph only effectively shows how needles mass varies over time and not by landcover type. Plotting all of the data on one plot makes it difficult to see how dry mass varies by land cover type, since it is difficult to see how the color of the points (landcover type) varies along the y axis (by mass). With the data seperated into three plots by landcover type, it is easy to see how mass varies by landcover type.