Assignment 5: Data Visualization

Sakina Shahid

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.

```r
#1
library(tidyverse)
```

```
## -- 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
```

```
library(lubridate)
library(cowplot)
```

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

```
library(ggplot2)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2023
```

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
lakes.df <- read.csv(here('Data','Processed_KEY',
                    'NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
                     stringsAsFactors = TRUE)
litter.df <- read.csv(here('Data','Processed_KEY',
                    'NEON_NIWO_Litter_mass_trap_Processed.csv'),
                     stringsAsFactors = TRUE)
#2

#changing columns to dates for peter paul lakes
lakes.df$sampledate <- ymd(lakes.df$sampledate)
class(lakes.df$sampledate)
```

```
## [1] "Date"
```

```
litter.df$collectDate <- ymd(litter.df$collectDate)
class(litter.df$collectDate)
```

```
## [1] "Date"
```

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

```
#3

library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
library(colormap)


theme <- theme_linedraw(base_size = 13) +
  theme(axis.text = element_text
    (color = "black"),plot.title = element_text(color="black",size=14),
legend.position = "right",
legend.box.background = element_rect(color= "black"),
plot.background=element_rect(fill="lightyellow"),
panel.border=element_rect(linewidth = 1.5,linetype = "solid"),
axis.ticks= element_line(color="darkgrey",size=2),
panel.grid.major = element_line(size=0.1))
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

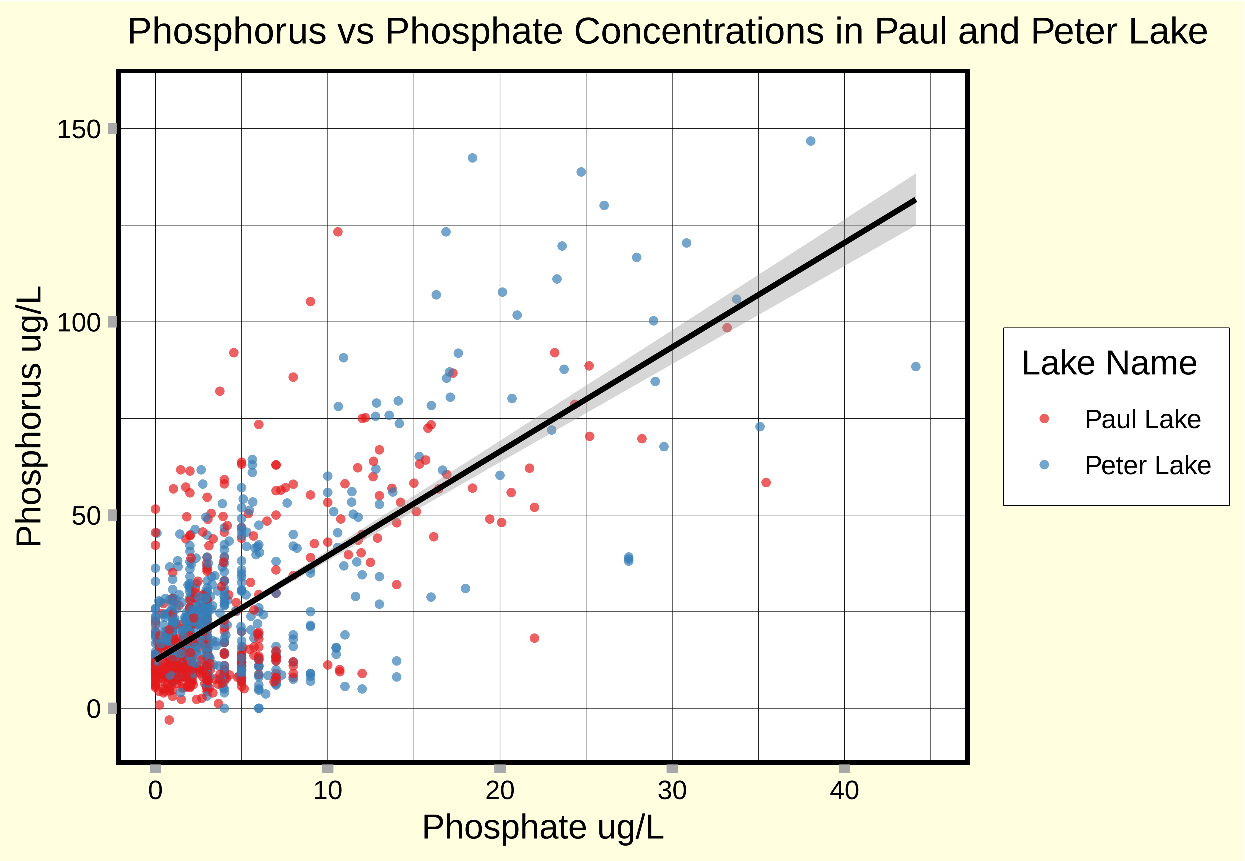
Create graphs

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
lake <- ggplot(lakes.df,aes(x = po4, y = tp_ug, color=lakename))+
  geom_point(alpha = 0.7, size = 1)+
  geom_smooth(method="lm",color="black")+
  theme+
  xlim(0,45)+
  xlab("Phosphate ug/L")+
  ylab("Phosphorus ug/L")+
  ggtitle(" Phosphorus vs Phosphate Concentrations in Paul and Peter Lake")+
  labs(color="Lake Name")+
 scale_colour_brewer(palette = "Set1")

  print(lake)
```

```
## ‘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()‘).
```

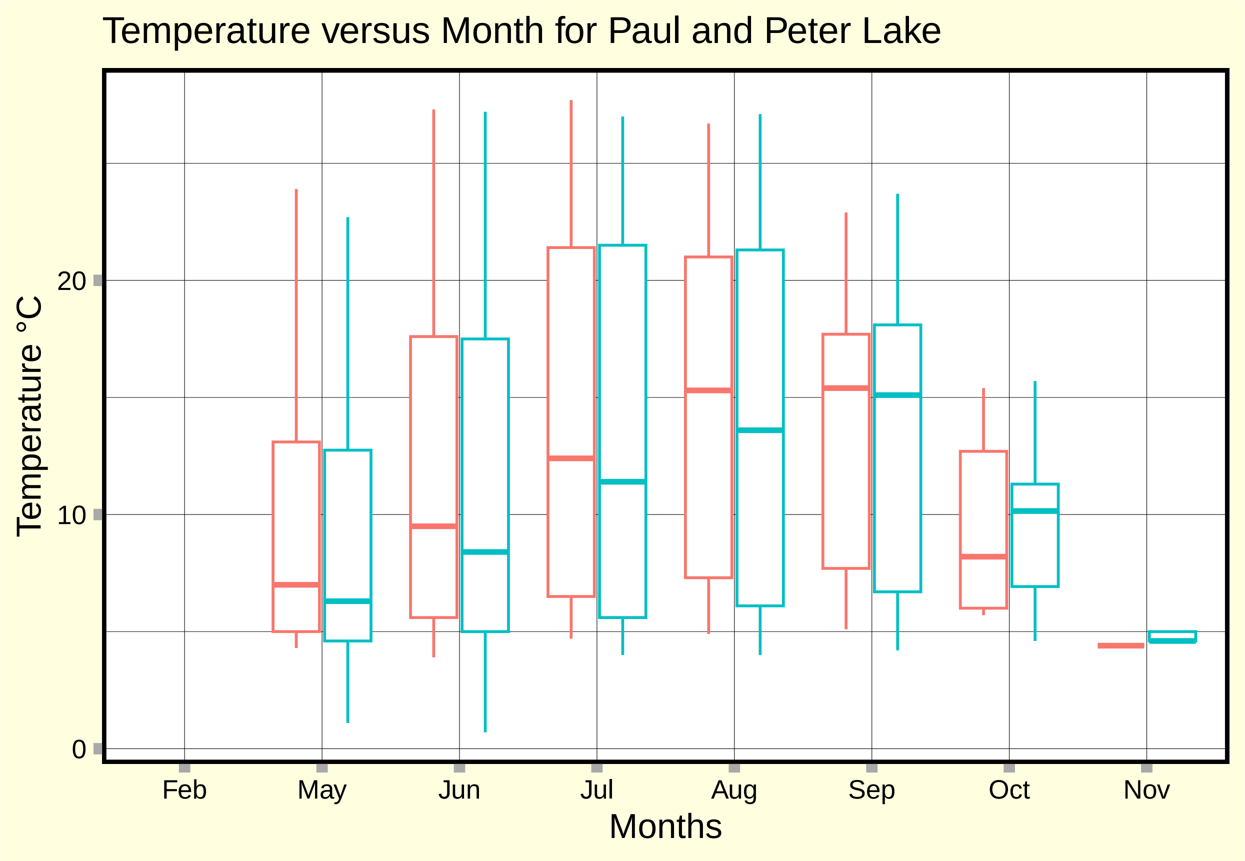


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

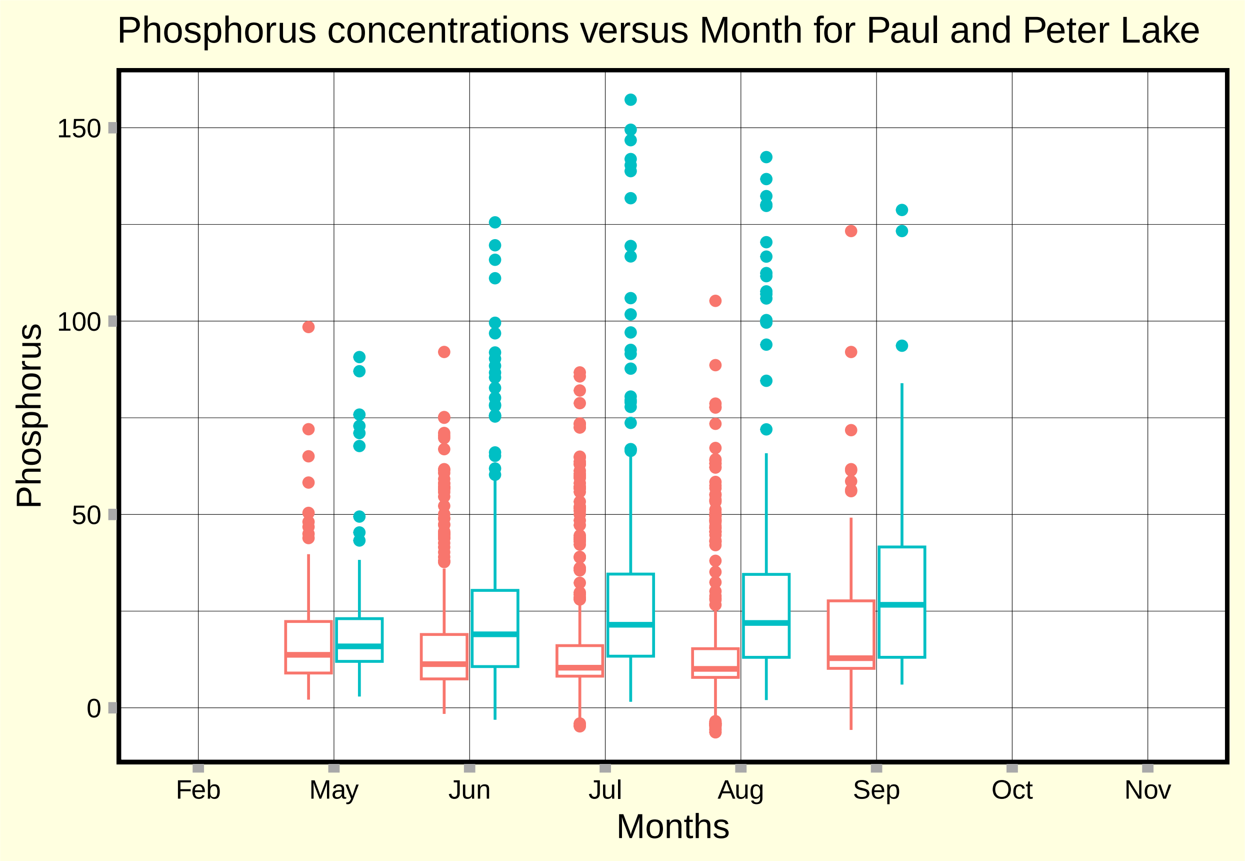
```r
#5

#Boxplot 1
temp_boxplot <- ggplot(lakes.df,
aes(x=factor(lakes.df$month, levels=1:12, labels=month.abb),
  y = temperature_C, color=lakename))+
  geom_boxplot()+
  xlab("Months")+
  ylab("Temperature °C")+
  labs(color="Lake Name")+
  ggtitle("Temperature versus Month for Paul and Peter Lake")+
  theme + theme(legend.position="none")

print(temp_boxplot)
```

```
## Warning: Use of `lakes.df$month` is discouraged.
## i Use `month` instead.

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



```
#Boxplot 2 with Phosphorus
TP_boxplot <- ggplot(lakes.df,
aes(x=factor(lakes.df$month,
levels=1:12, labels=month.abb),
y = tp_ug, color=lakename))+
  geom_boxplot()+
  xlab("Months")+
  ylab("Phosphorus")+
  labs(color="Lake Name")+
  ggtitle("Phosphorus concentrations versus Month for Paul and Peter Lake")+
  theme + theme(legend.position="none")

print(TP_boxplot)
```

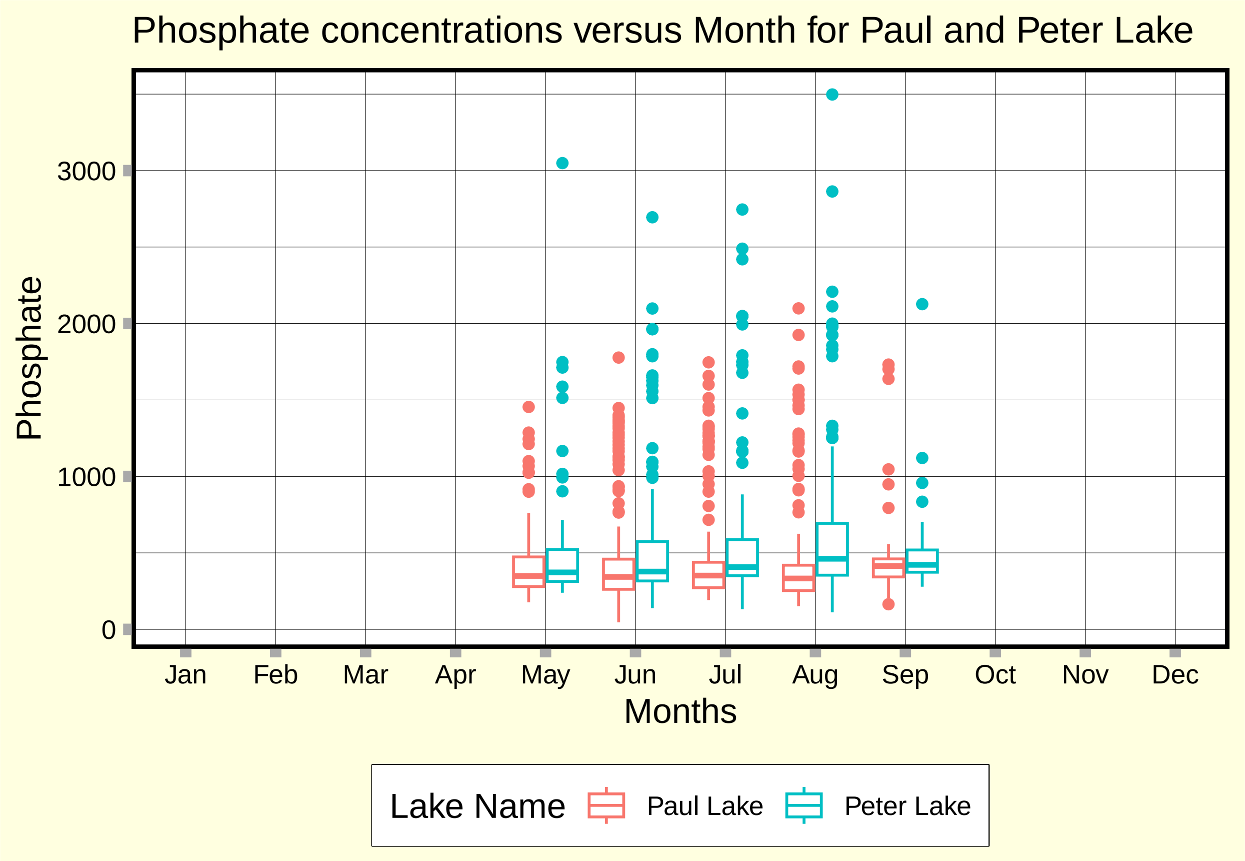
```
## Warning: Use of `lakes.df$month` is discouraged.
## i Use `month` instead.

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



```
#Boxplot 3

TN_boxplot <- ggplot(lakes.df,
aes(x=factor(lakes.df$month,
levels=1:12, labels=month.abb),
y = tn_ug, color=lakename))+
  geom_boxplot()+
  xlab("Months")+
  ylab("Phosphate")+
  labs(color="Lake Name")+
  ggtitle("Phosphate concentrations versus Month for Paul and Peter Lake")+
  theme +
  theme(legend.position="bottom")+
  scale_x_discrete(name="Months",drop=FALSE)

print(TN_boxplot)
```

```
## Warning: Use of ‘lakes.df$month‘ is discouraged.
## i Use ‘month‘ instead.
```

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



```
#cowplot
library(ggpubr)
```

```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:cowplot':
##
##     get_legend
```

```
library(cowplot)

cowplot <- plot_grid(temp_boxplot,TP_boxplot,TN_boxplot,
             nrow = 3, align = "vh",
  rel_heights = c(2,2,3.0))
```

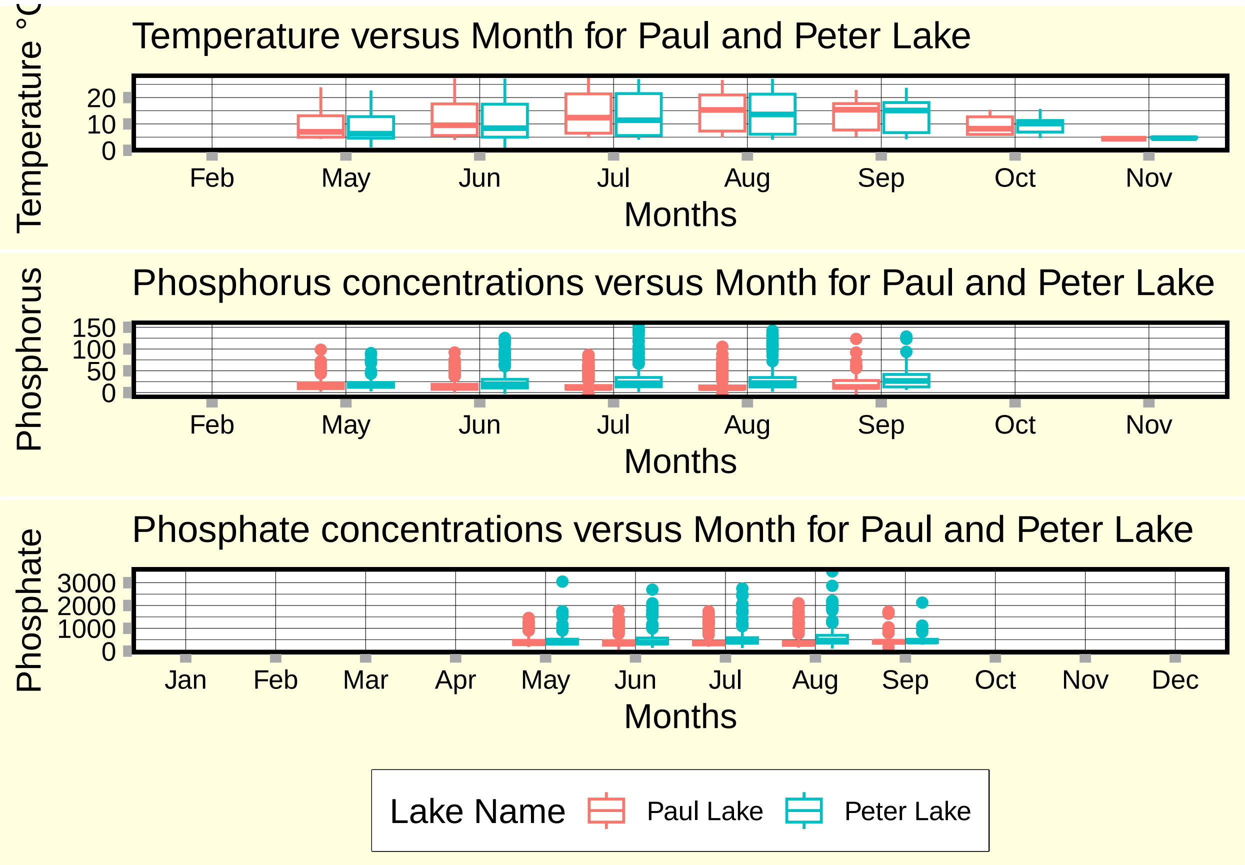
```
## Warning: Use of `lakes.df$month` is discouraged.
## i Use `month` instead.

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

## Warning: Use of `lakes.df$month` is discouraged.
## i Use `month` instead.
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Use of `lakes.df$month` is discouraged.
## i Use `month` instead.
```

```
## 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(cowplot)
```



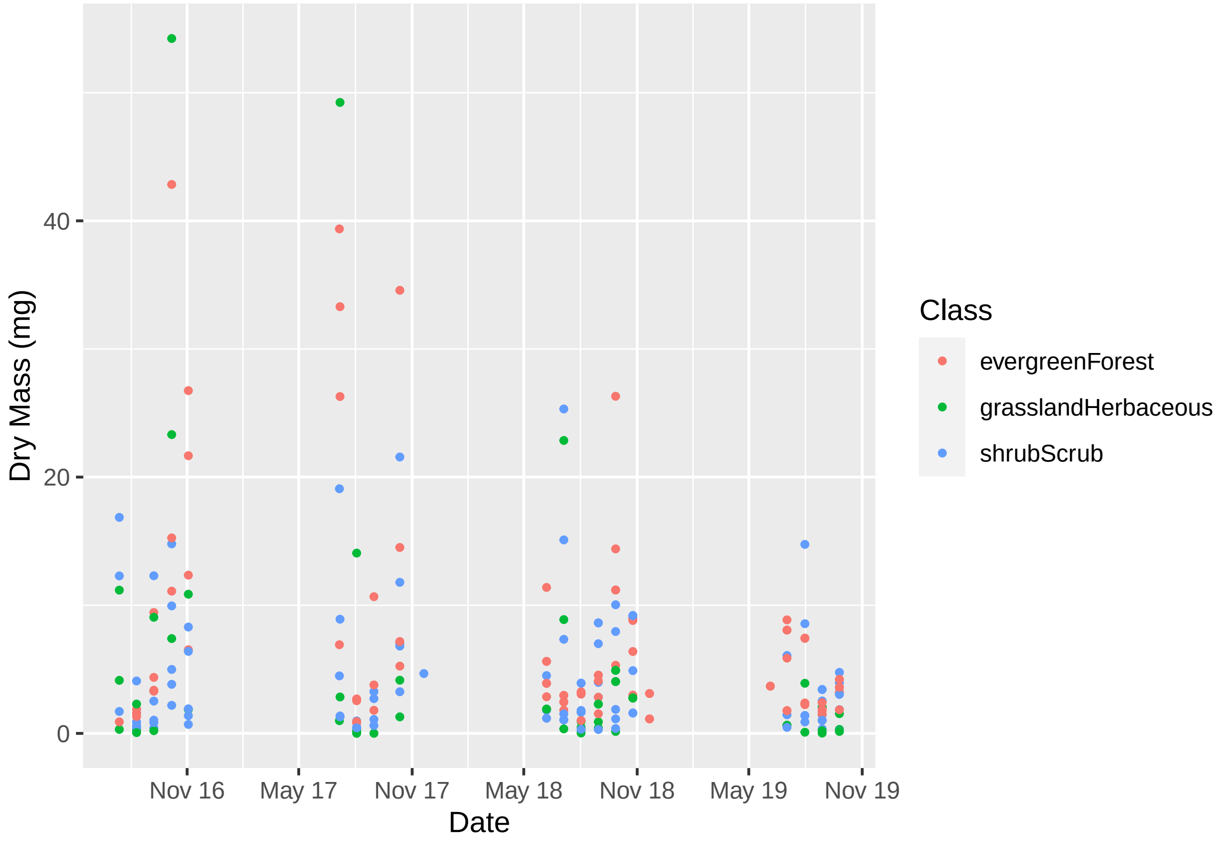
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The medians for Paul and Peter Lake for variable Temperature are similar across all months except October. There is no data for both lakes between the months of December to April.Phosphorus concentrations are higher in Peter Lake,observable starting in June going until September. I notice that while concentrations remain relatively consistent in Paul Lake, there is a gradual increase in Phosphorus concentrations in Peter Lake. The median Phosphate concentrations fall within the range of the IQR for each lake, making the difference relatively small. There is observably more Phosphate in Peter Lake in August and the outliers across all months are larger compared to Paul Lake, indicating more culmulative concentrations of Phosphate in Peter 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.

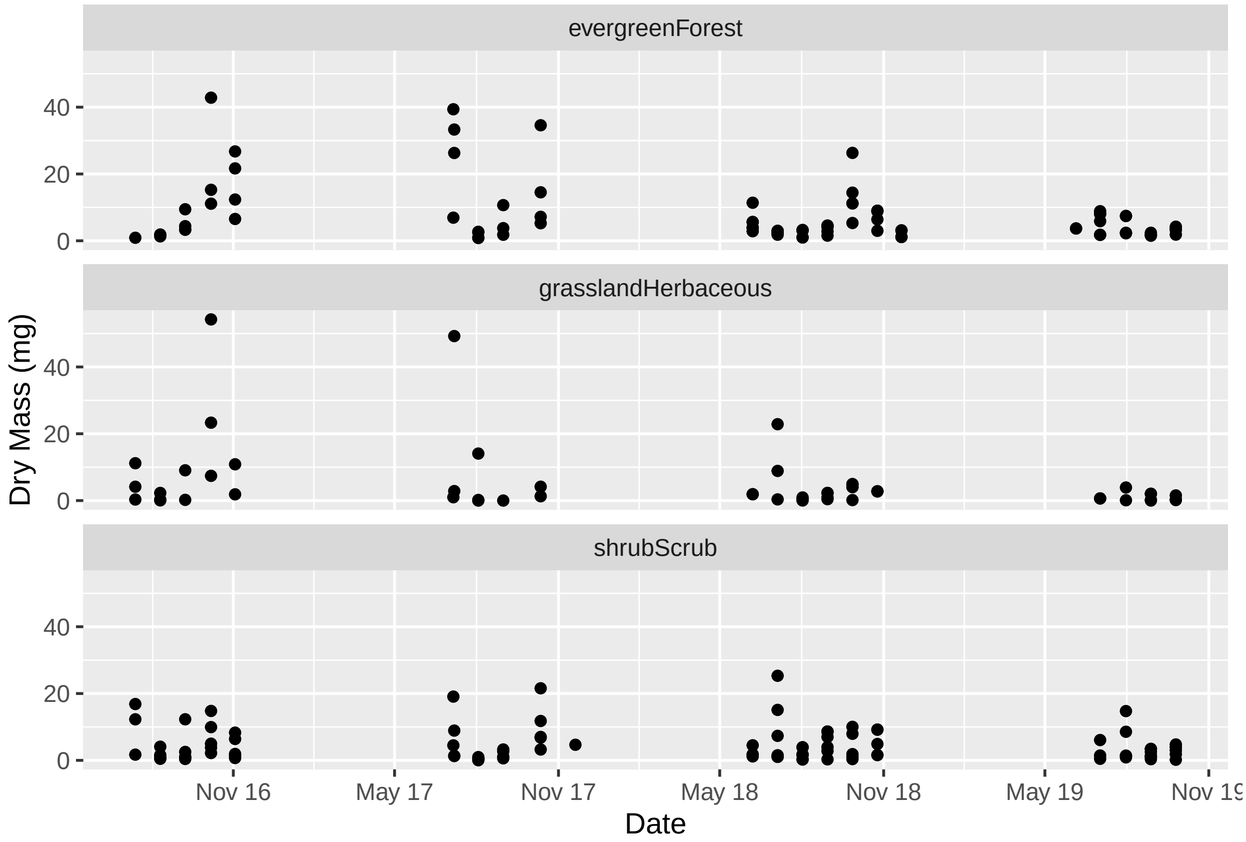
```
#6
needles.df <- litter.df %>% filter(functionalGroup=="Needles")

needle_plot <- ggplot(needles.df,aes(x=collectDate,
                                     y=dryMass,
                                     color=nlcdClass))+ geom_point(size=0.9)+xlab("Date")+ylab("Dry Mas
  labs(color="Class")+
  scale_x_date(limits = as.Date(c("2016-07-14", "2019-09-25")),
    date_breaks = "6 months", date_labels = "%b %y")

print(needle_plot)
```

```
## Warning: Removed 6 rows containing missing values (`geom_point()`).
```



```
#7
needle_plot2 <-
  ggplot(needles.df, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)+
  geom_point(size=0.9)+xlab("Date")+ylab("Dry Mass (mg)")+
  labs(color="Class")+
  scale_x_date(limits = as.Date(c("2016-07-14", "2019-09-25")),
    date_breaks = "6 months", date_labels = "%b %y")

print(needle_plot2)
```

```
## Warning: Removed 6 rows containing missing values (`geom_point()`).
## Removed 6 rows containing missing values (`geom_point()`).
```



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

> Answer: I think Plot 6 is more effective because I can compare all of the classes within one plot. It is difficult to understand the variation in reference to other classes in Plot 7 because I cannot see them relative to each other. Even within Plot 6, it is difficult to understand whether the distribution is statistically signficantly different from each other. Both graphs are showcasing frequency rather than giving insight into characteristics and differences between data (like a violin plot).