Assignment 5: Data Visualization

Desa Bolger

Fall 2023

OVERVIEW

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

Directions

1. Rename this file `Desa_Bolger_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 Getting things added!
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 errors
```

```r
library(lubridate)
library(here)
```

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

```r
library(cowplot)
```

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

```r
library(ggplot2)
library(ggthemes)
```

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

```r
library(viridis)
```

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

```r
library(RColorBrewer)

here() #"/home/guest/EDE_Fall2023"
```

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

```r
processed_data = "Data/Processed/Processed_KEY"
PeterPaul.chem.nutrients <- read.csv(
  here(processed_data,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)

#view(PeterPaul.chem.nutrients) #Looks Good

Niwot_Ridge <- read.csv(
  here(processed_data,"NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)
view(Niwot_Ridge)

#2 Change to Date!

class(PeterPaul.chem.nutrients$sampledate)
```

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

```
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
class(PeterPaul.chem.nutrients$sampledate)
```

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

```
class(Niwot_Ridge$collectDate)
```

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

```
Niwot_Ridge$collectDate <- ymd(Niwot_Ridge$collectDate)
class(Niwot_Ridge$collectDate)
```

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

```
#Date changed
```

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 Building my plot theme

desa_theme <- theme_base() +
  theme(line = element_line(color='black',linewidth =.5),
        text = element_text(color='black'),
        panel.grid.major =  element_line(color='black', linewidth = .5),
        rect = element_rect(color = 'lightgrey', fill = 'lightgrey'),
        plot.background = element_rect(color = 'lightgrey', fill = 'lightgrey'),
        panel.background =  element_rect(color = 'lightgrey', fill = 'lightgrey'),
        legend.background = element_rect(color='lightblue', fill = 'lightblue'),
        legend.title = element_text(color='darkblue'))

theme_set(desa_theme) #set as default
```

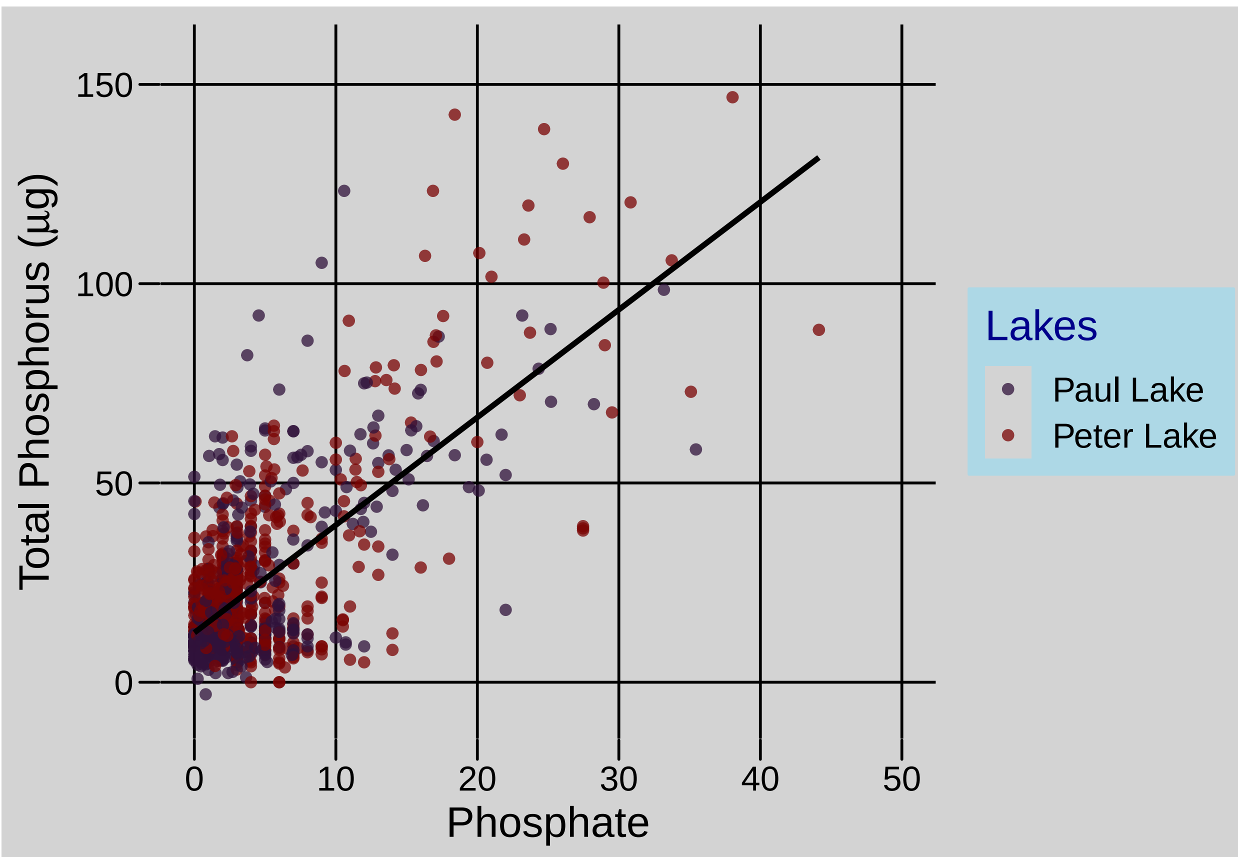
Create graphs

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()`).

```r
#4 PHOS x PHOSPHATE.

PhosByPhos <- ggplot(PeterPaul.chem.nutrients, aes(x = po4, y = tp_ug,
                                                   color = lakename)) +
  geom_point(alpha = 0.75, size = 1.5) +
  xlim(0,50)+
  scale_color_viridis(discrete = TRUE, option = "turbo", name = "Lakes") +
  stat_smooth(method = lm, se = F, color = 'black')+
  xlab(expression(paste("Phosphate" )))+
  ylab(expression(paste("Total Phosphorus (", mu, "g)")))

print(PhosByPhos)
```

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

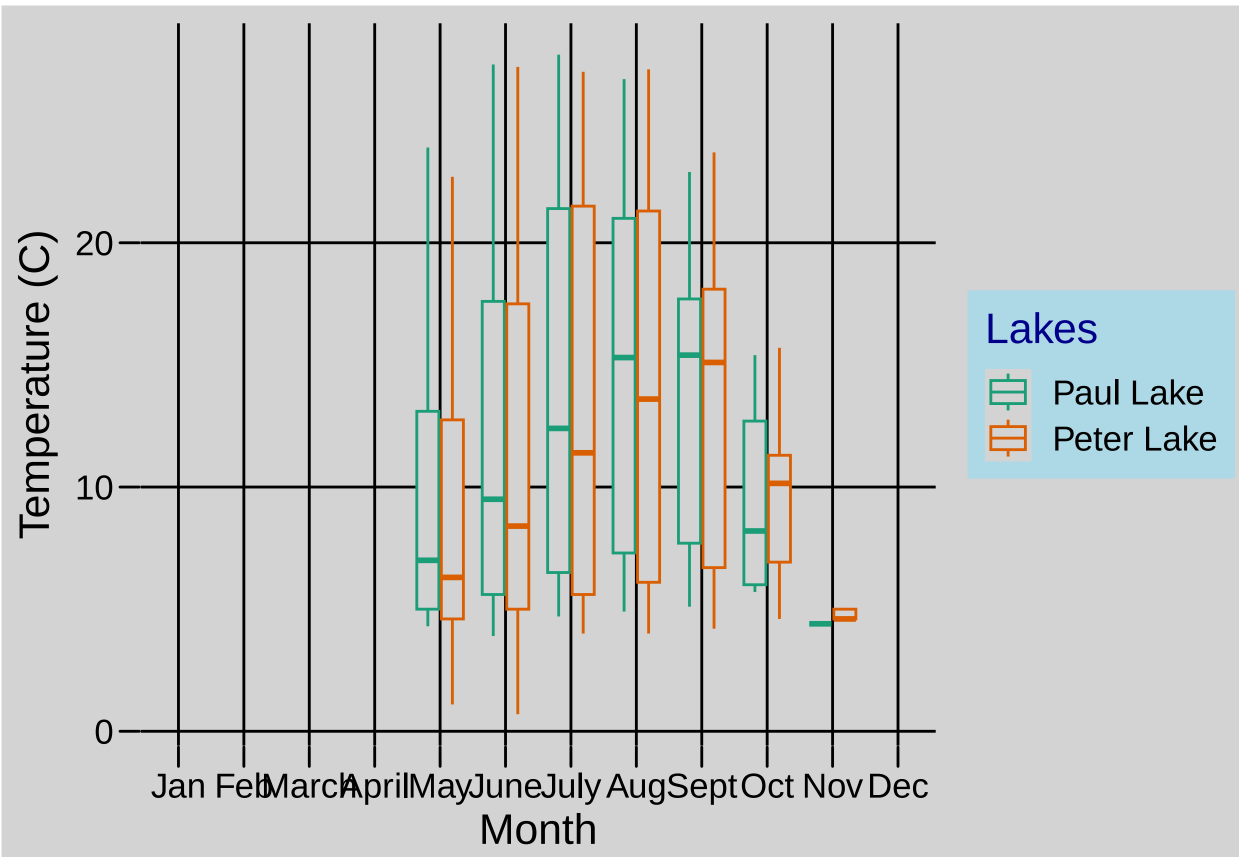


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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!!!

PeterPaul.chem.nutrients$month <- as.factor(PeterPaul.chem.nutrients$month)

Temp <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = temperature_C,
                                              color = lakename)) +
  geom_boxplot(fill = 'lightgrey')+
  scale_x_discrete(limits = factor("1":"12"), labels = c("Jan", "Feb", "March",
  "April", "May", "June", "July", "Aug", "Sept", "Oct", "Nov", "Dec"))+
  xlab(expression(paste("Month")))+
  ylab(expression(paste("Temperature (C)")))+
  scale_color_brewer(palette = "Dark2", name = "Lakes")

print(Temp)
```

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

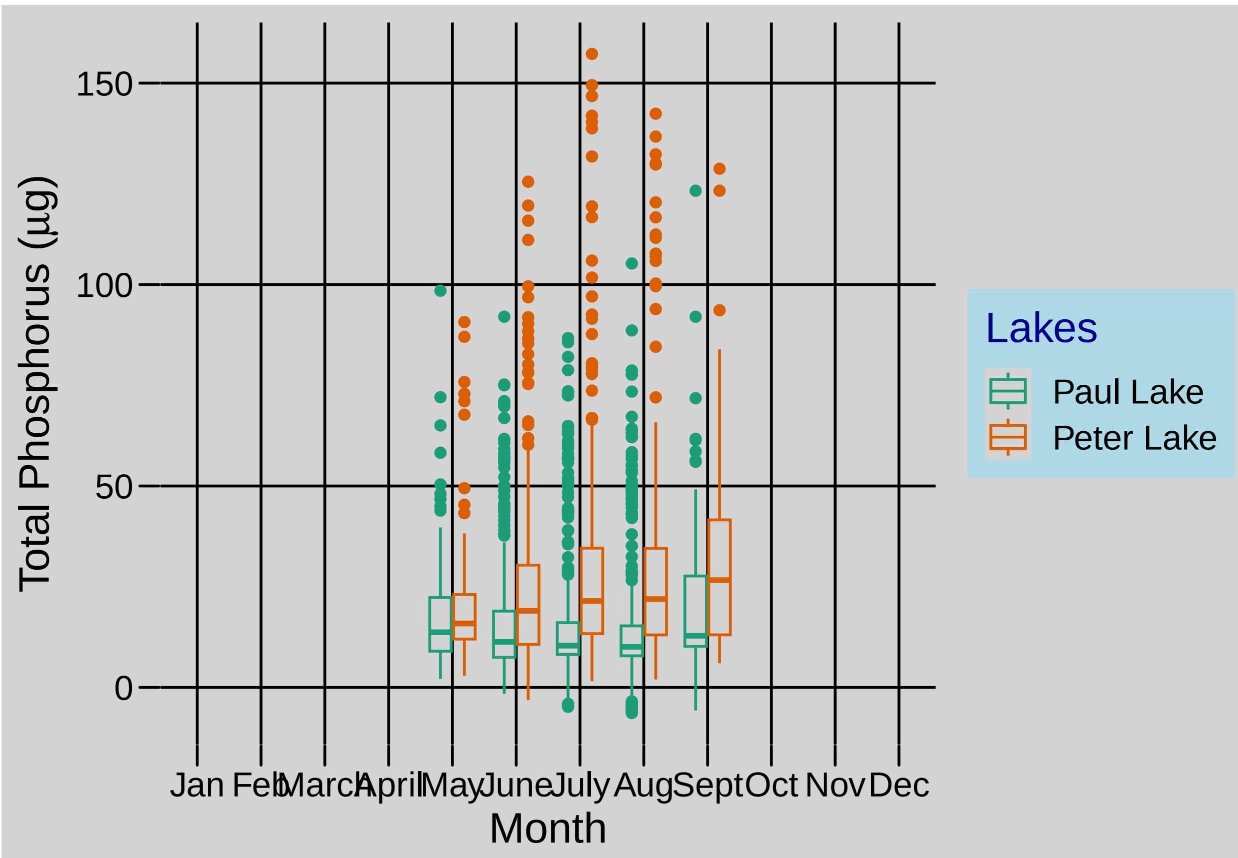



```r
# getting the legend
FinalLegend <- get_legend(Temp)
```

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

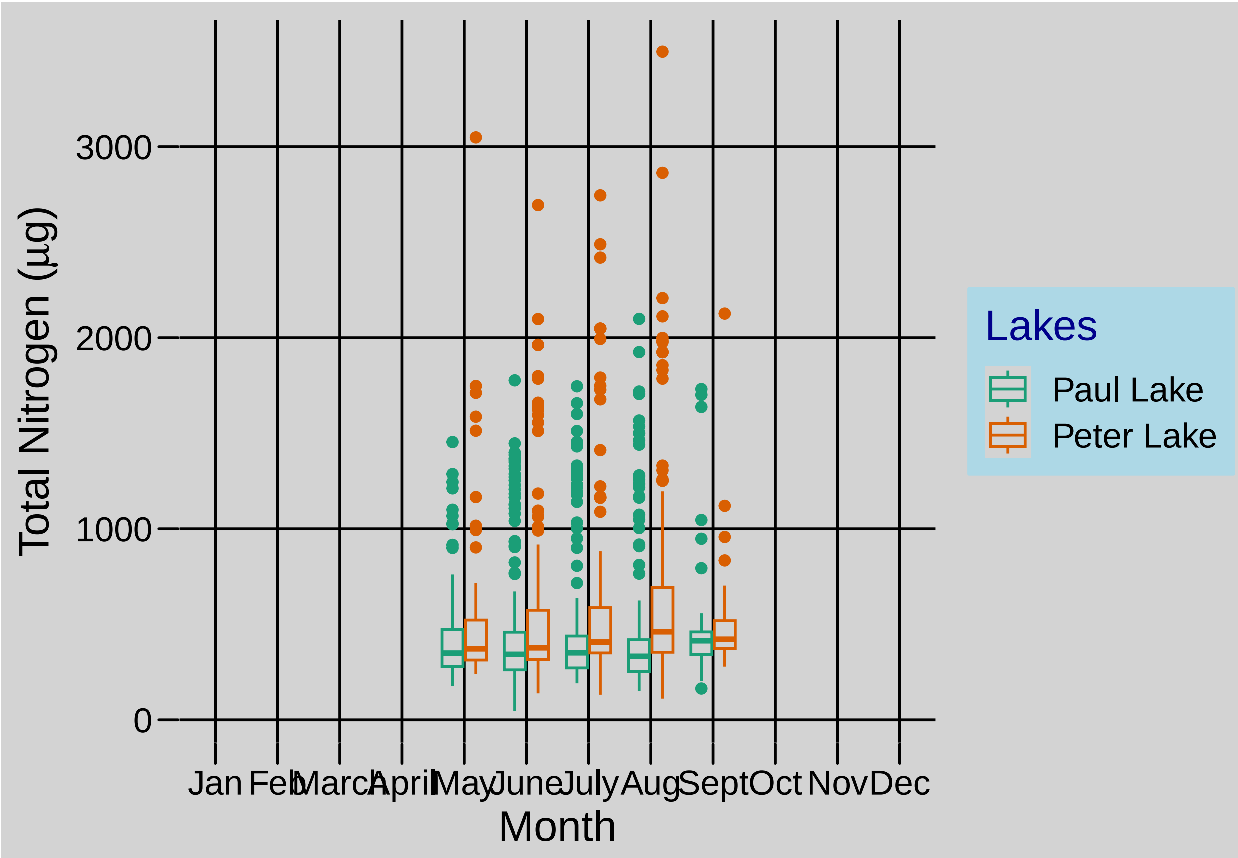
```r
TP <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tp_ug,
                                            color = lakename)) +
  geom_boxplot(fill = 'lightgrey')+
 scale_x_discrete(limits = factor("1":"12"), labels = c("Jan", "Feb", "March",
  "April", "May", "June", "July", "Aug", "Sept", "Oct", "Nov", "Dec"))+
  xlab(expression(paste("Month")))+
  ylab(expression(paste("Total Phosphorus (", mu, "g)")))+
  scale_color_brewer(palette = "Dark2", name = "Lakes")

print(TP)
```

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




```r
TN <- ggplot(PeterPaul.chem.nutrients, aes(x = month, y = tn_ug,
                                            color = lakename)) +
  geom_boxplot(fill = 'lightgrey')+
 scale_x_discrete(limits = factor("1":"12"), labels = c("Jan", "Feb", "March",
  "April", "May", "June", "July", "Aug", "Sept", "Oct", "Nov", "Dec"))+
  xlab(expression(paste("Month")))+
  ylab(expression(paste("Total Nitrogen (", mu, "g)")))+
  scale_color_brewer(palette = "Dark2", name = "Lakes")

print(TN)
```

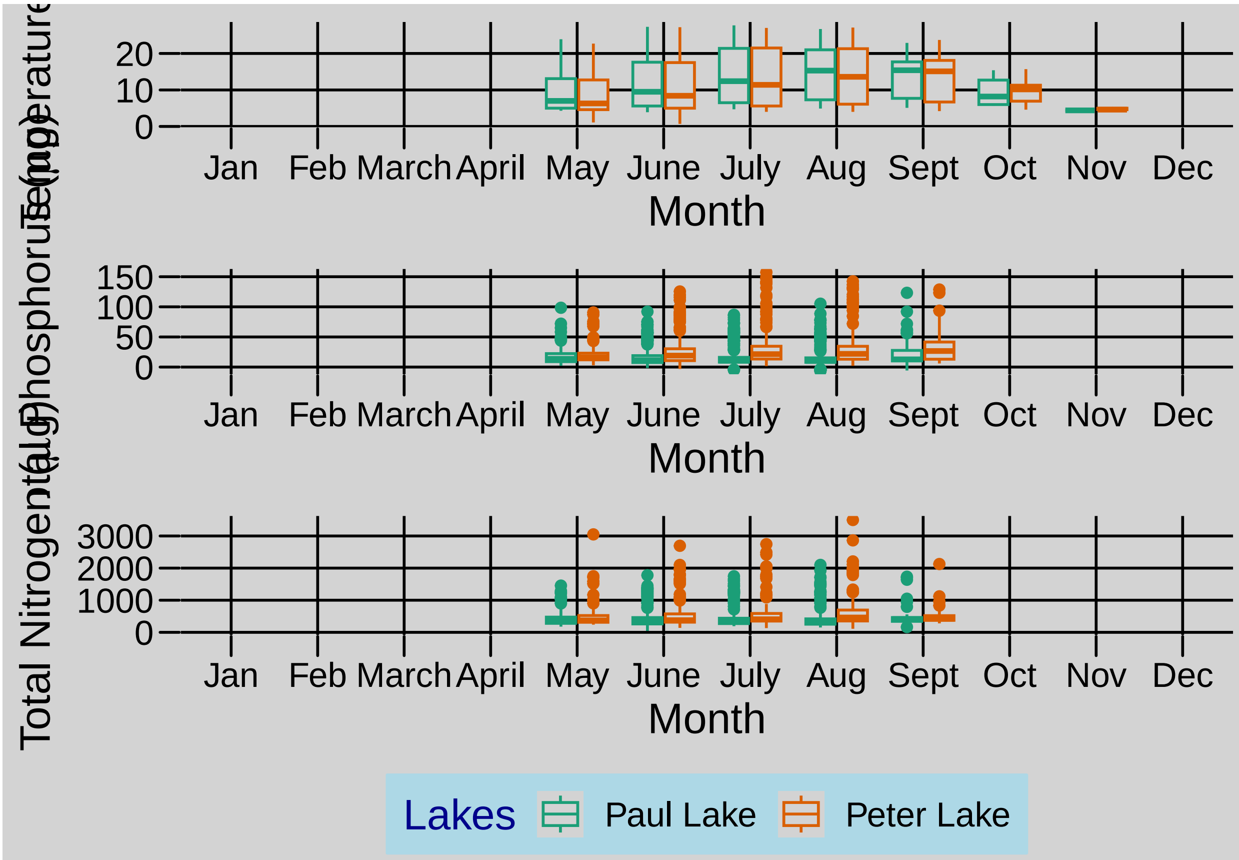
```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```




```
finished <- plot_grid(
  Temp + theme(legend.position = "none"),
  TP + theme(legend.position = "none"),
  TN + theme(legend.position = "bottom"),
  rel_heights = c(1,1,1.5),
  ncol = 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()‘).

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

```
print(finished)
```

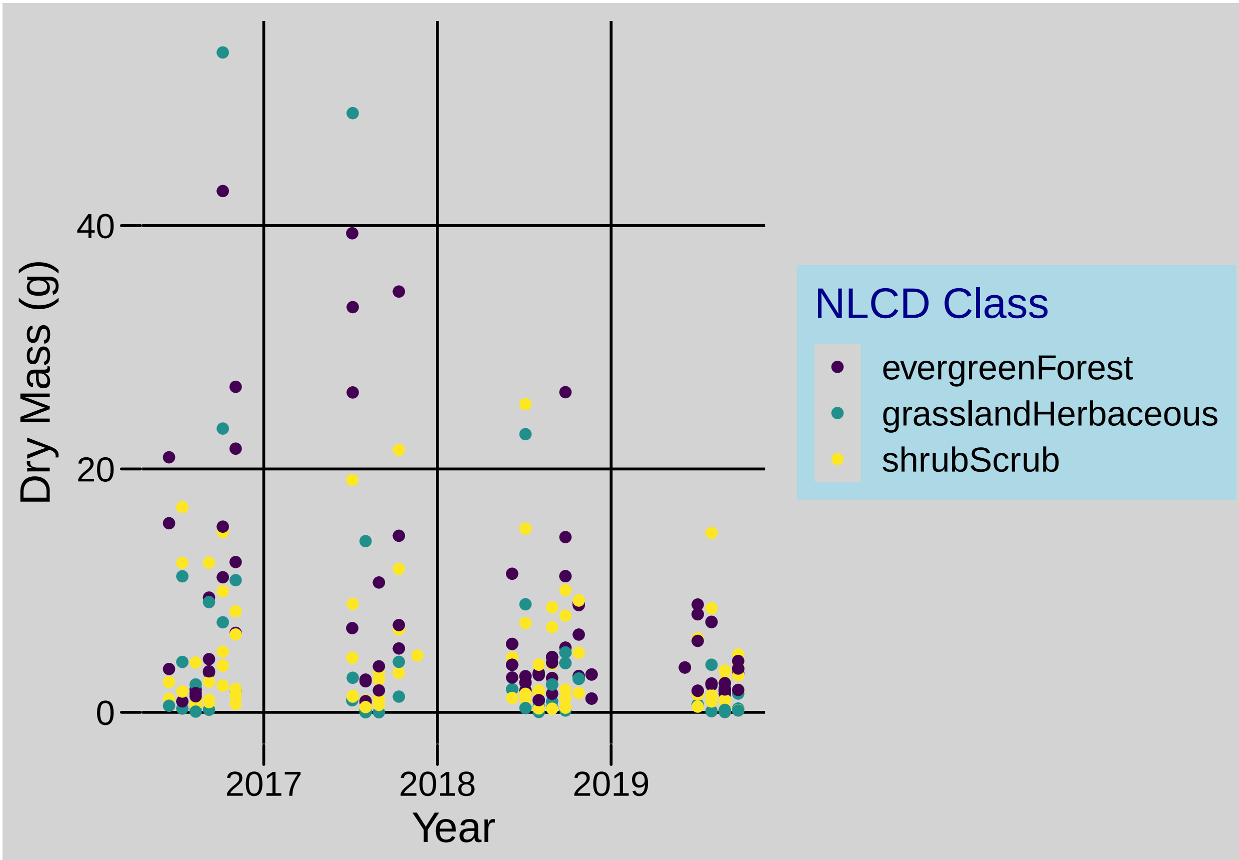


Question: What do you observe about the variables of interest over seasons and between lakes?

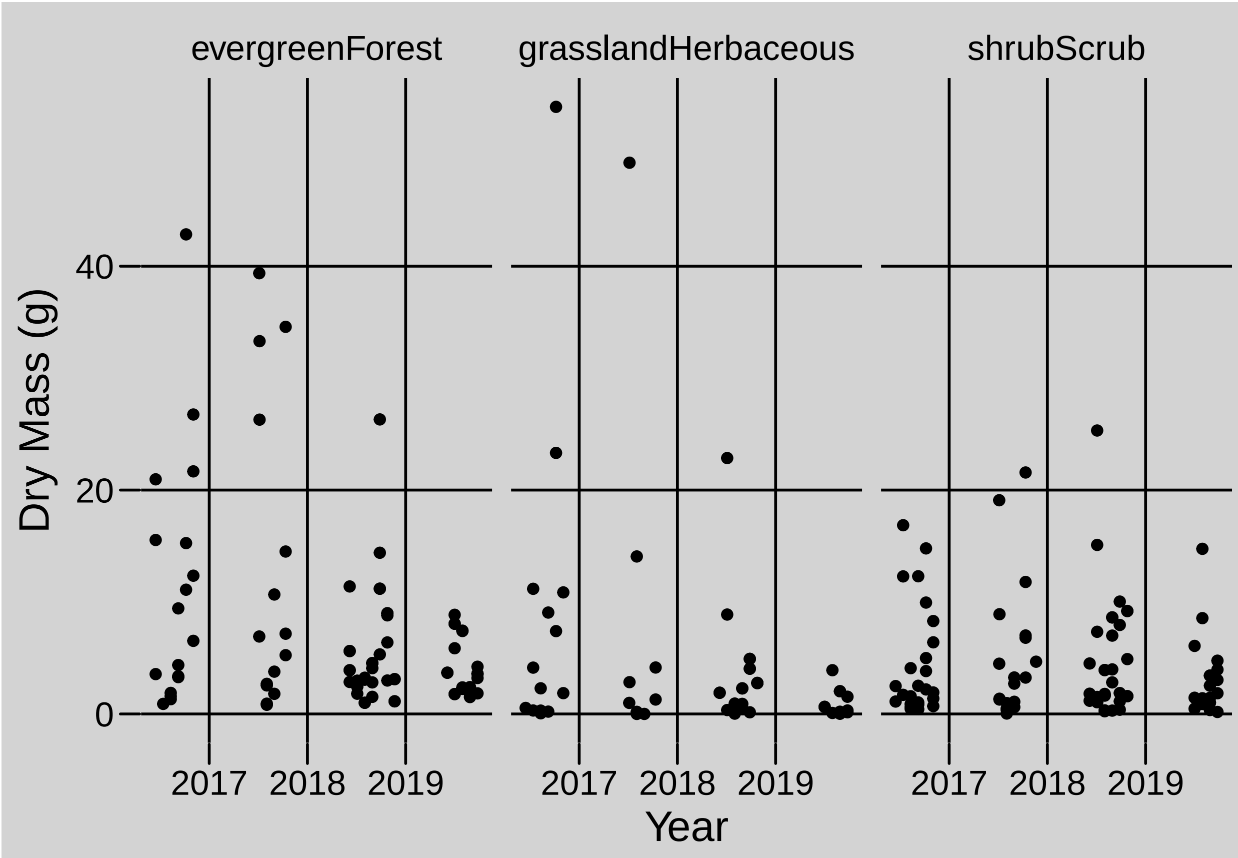
> Answer: Temperature is highest in the summer months (July and August), but fairly consistent for Paul and Peter Lakes. Total Phosphate is highest in July, and it seems Peter lake has a higher TP level than Paul. Lastly, TN seems fairly consistent, though Peter lake values seem higher.

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 Isolating Needles.

Needles <- ggplot(subset(Niwot_Ridge, functionalGroup == 'Needles' ), aes(
  x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  scale_color_viridis(discrete = TRUE, name = "NLCD Class") +
  xlab(expression(paste("Year" )))+
  ylab(expression(paste("Dry Mass (g)")))
  print(Needles)
```




```r
#7

Needles2 <- ggplot(subset(Niwot_Ridge, functionalGroup == 'Needles' ), aes(
  x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  xlab(expression(paste("Year")))+
  ylab(expression(paste("Dry Mass (g)")))
print(Needles2)
```



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

Answer: 7 because its easier to see the spread for the three different locations since they are not overlapped (like in 6).