Assignment 5: Data Visualization

Laura Exar

Spring 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. 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_PeterPaul_Processed.csv` version) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version).

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

```
#1 Loaded packages
library(tidyverse); library(lubridate); library(here); library(cowplot)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'lubridate'
```

```
##
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
##
##
## here() starts at /home/guest/R/EDA-Spring2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here(); library(ggthemes)
```

```
## [1] "/home/guest/R/EDA-Spring2023"
```

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

```
#Uploaded datafiles
PeterPaul.chem.nutrients <- read.csv(
  file=here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE
)
Litter <- read.csv(
  file=here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)

#2 Changed dates to date format
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Litter$collectDate <- ymd(Litter$collectDate)
```

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 Built my theme
my_theme <- theme_base() +
  theme(
    axis.line = element_line(
      linewidth = 3,
      colour = "black"),
    plot.background = element_rect(
      color='grey'
    ),
    plot.title = element_text(
      color='blue'
    )

  )
```

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()`).

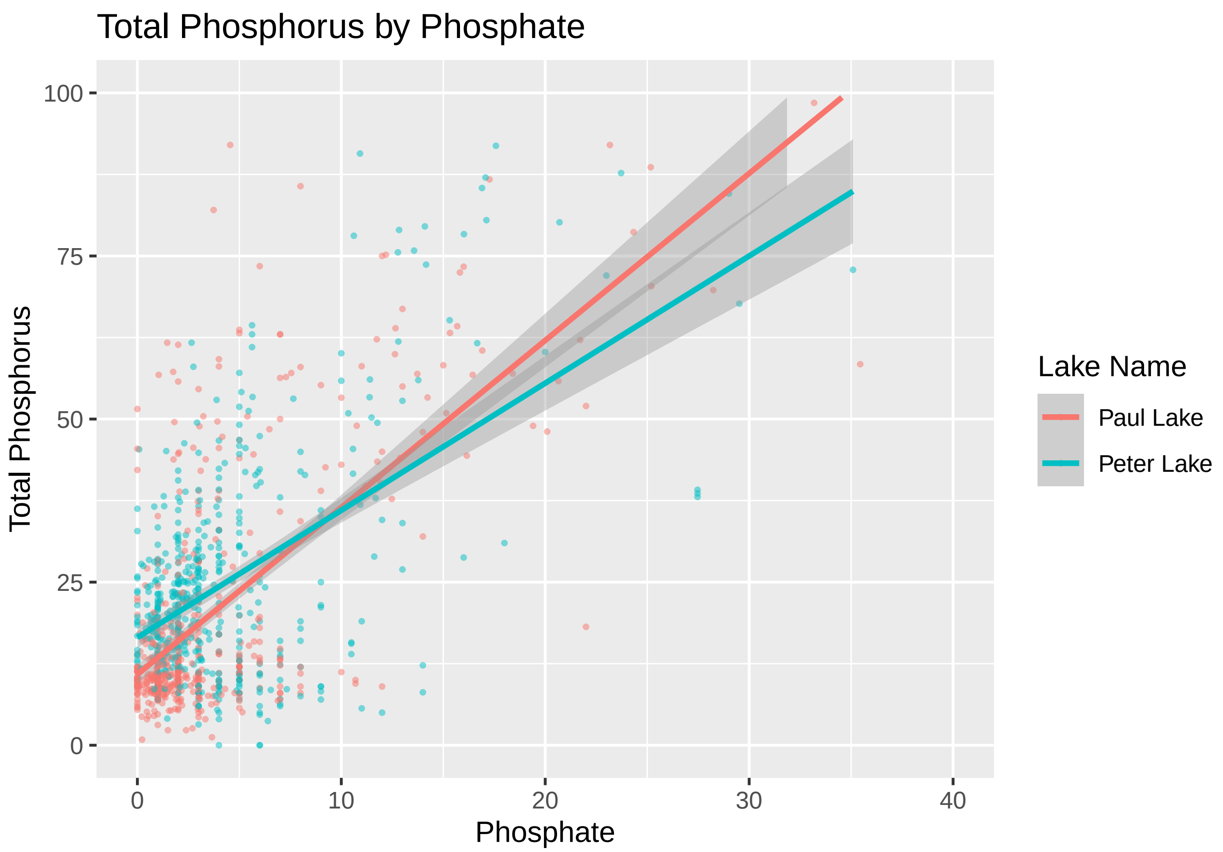
```
#4 Plotted total phosphorus by total phosphate
Phosphorus.by.Phosphate <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      x=po4,
      y=tp_ug,
      color= lakename)
      ) +
  geom_point(size=0.5, alpha=0.5) + #alpha sets transparency
  labs(title="Total Phosphorus by Phosphate",
       x="Phosphate",
       y="Total Phosphorus",
       color = "Lake Name"
       )+
  geom_smooth(method=lm) +
  xlim(0, 40) +
  ylim(0, 100)
Phosphorus.by.Phosphate
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 21965 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 21965 rows containing missing values (‘geom_point()‘).
```

```
## Warning: Removed 2 rows containing missing values (‘geom_smooth()‘).
```

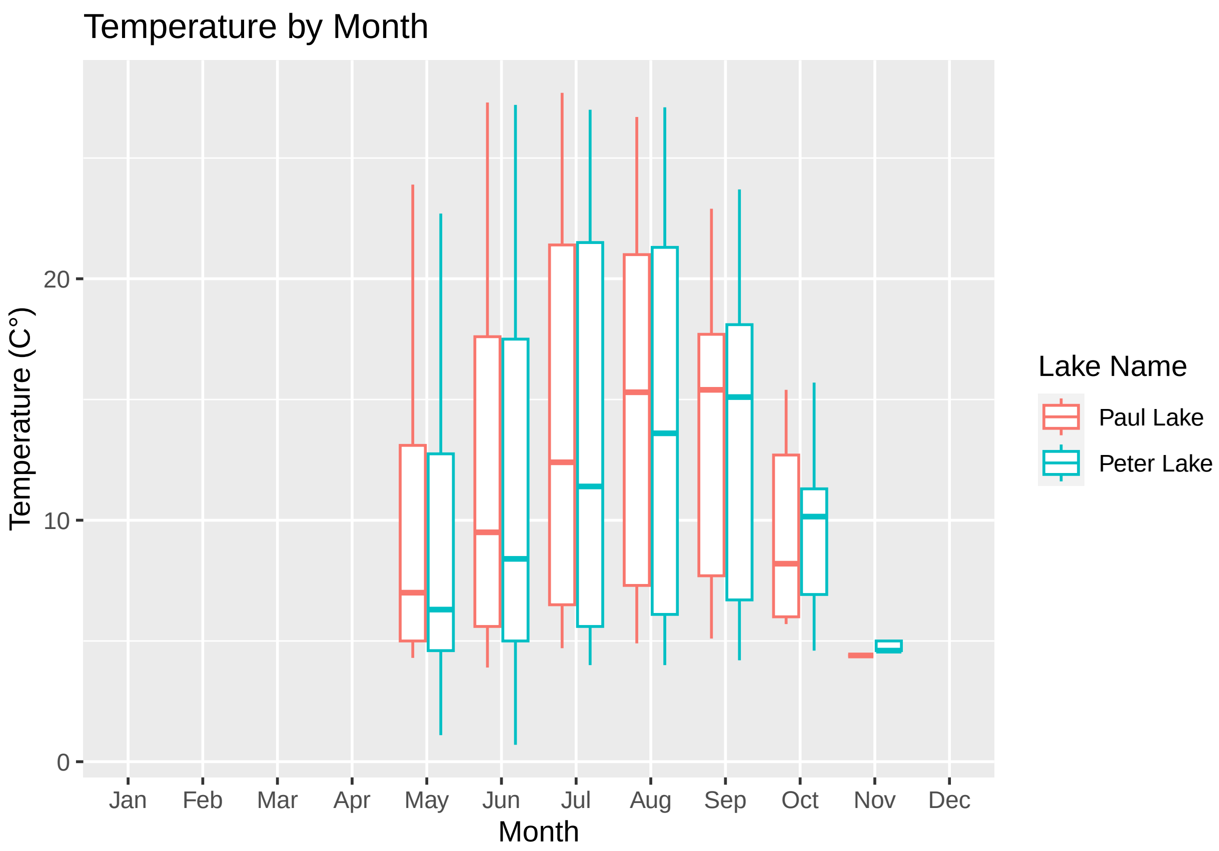


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

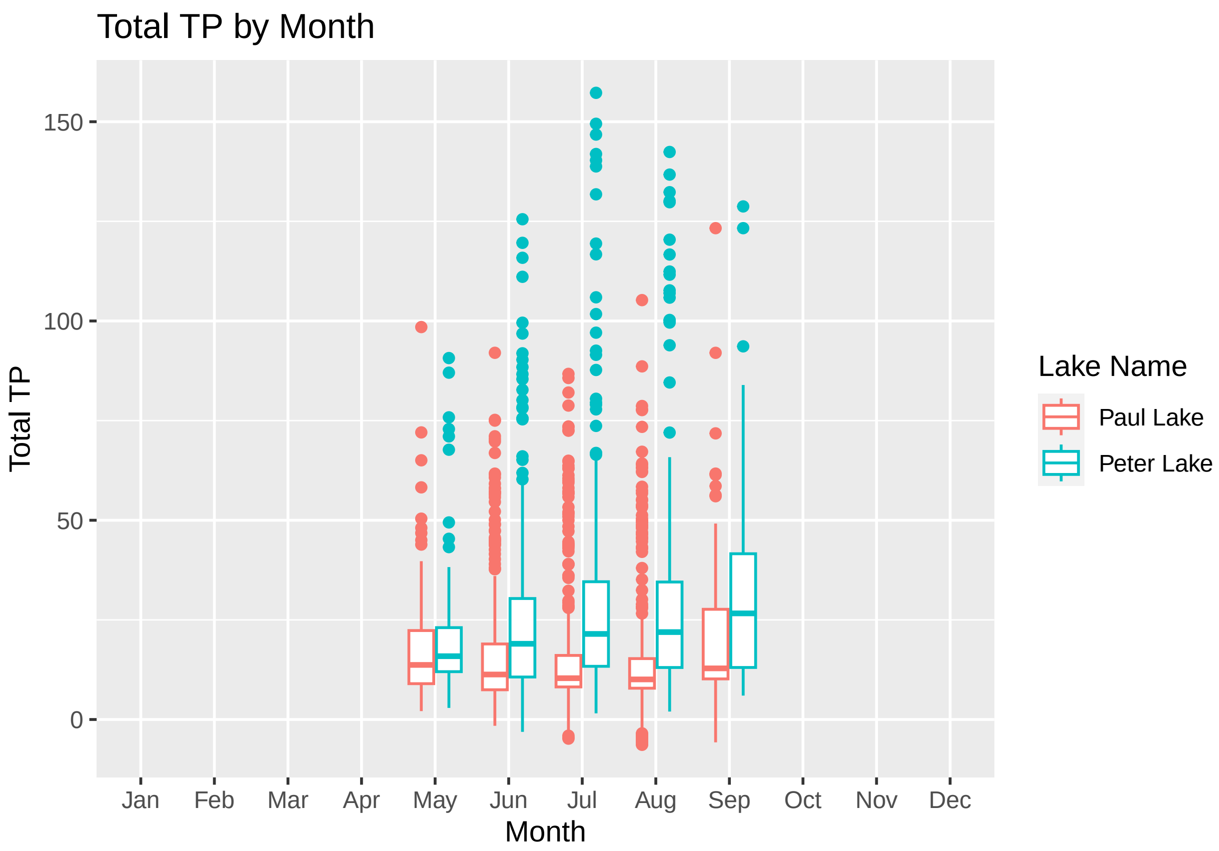
```
#5 Plotted Temperature by Month
Temp <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      x=factor(month, levels=1:12, labels=month.abb),
      y=temperature_C,
      color= lakename)
      ) +
  geom_boxplot() +
  scale_x_discrete(
    name="Month",
    drop=F) +
  labs(title="Temperature by Month",
       x="Month",
       y="Temperature (C°)",
       color = "Lake Name"
       )
Temp
```

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



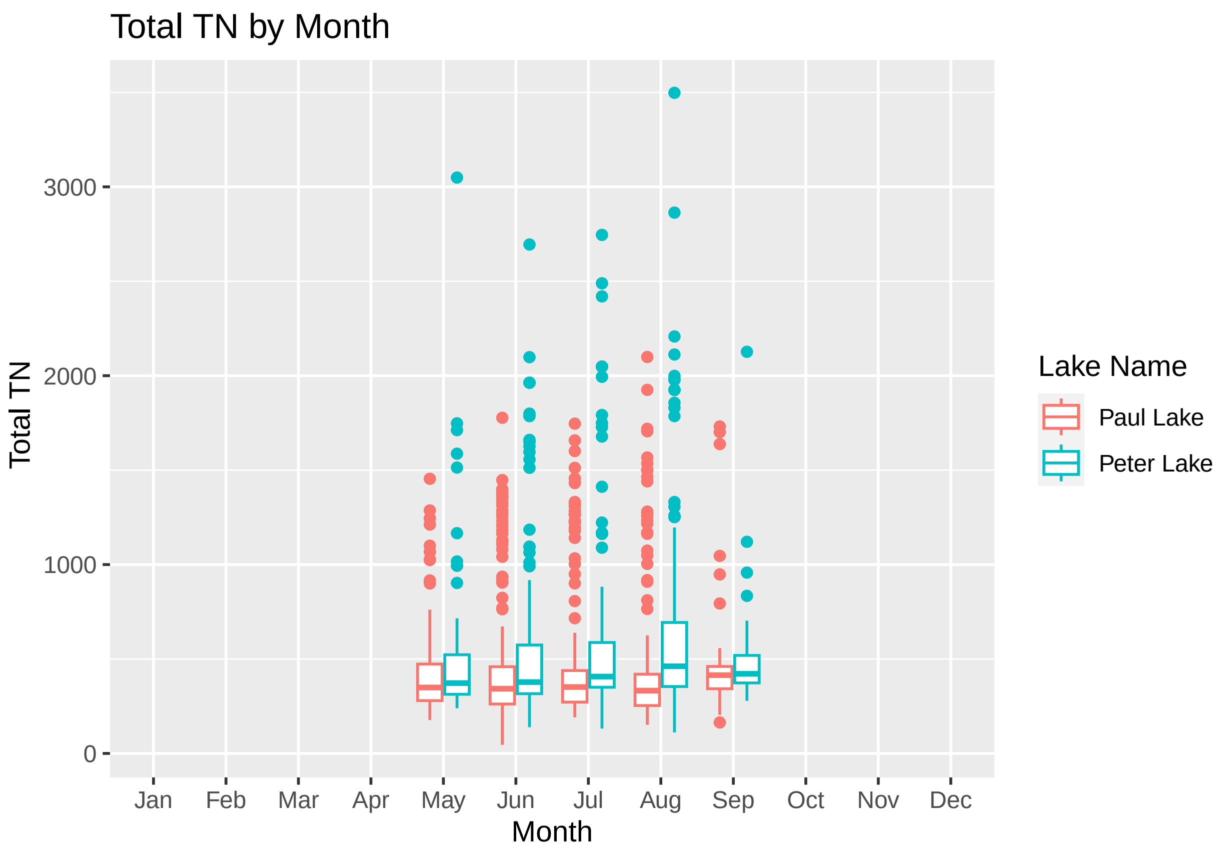
```
#Plotted TP by Month
TP <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
      x=factor(month, levels=1:12, labels=month.abb),
      y=tp_ug,
      color= lakename)
      ) +
  geom_boxplot() +
  scale_x_discrete(
    name="Month",
    drop=F) +
  labs(title="Total TP by Month",
       x="Month",
       y="Total TP",
       color = "Lake Name"
       )
TP
```

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



```
#Plotted TN by Month
TN <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
     x=factor(month, levels=1:12, labels=month.abb),
      y=tn_ug,
      color= lakename)
      ) +
  geom_boxplot() +
  scale_x_discrete(
    name="Month",
    drop=F) +
  labs(title="Total TN by Month",
       x="Month",
       y=" Total TN",
       color = "Lake Name"
       )
TN
```

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



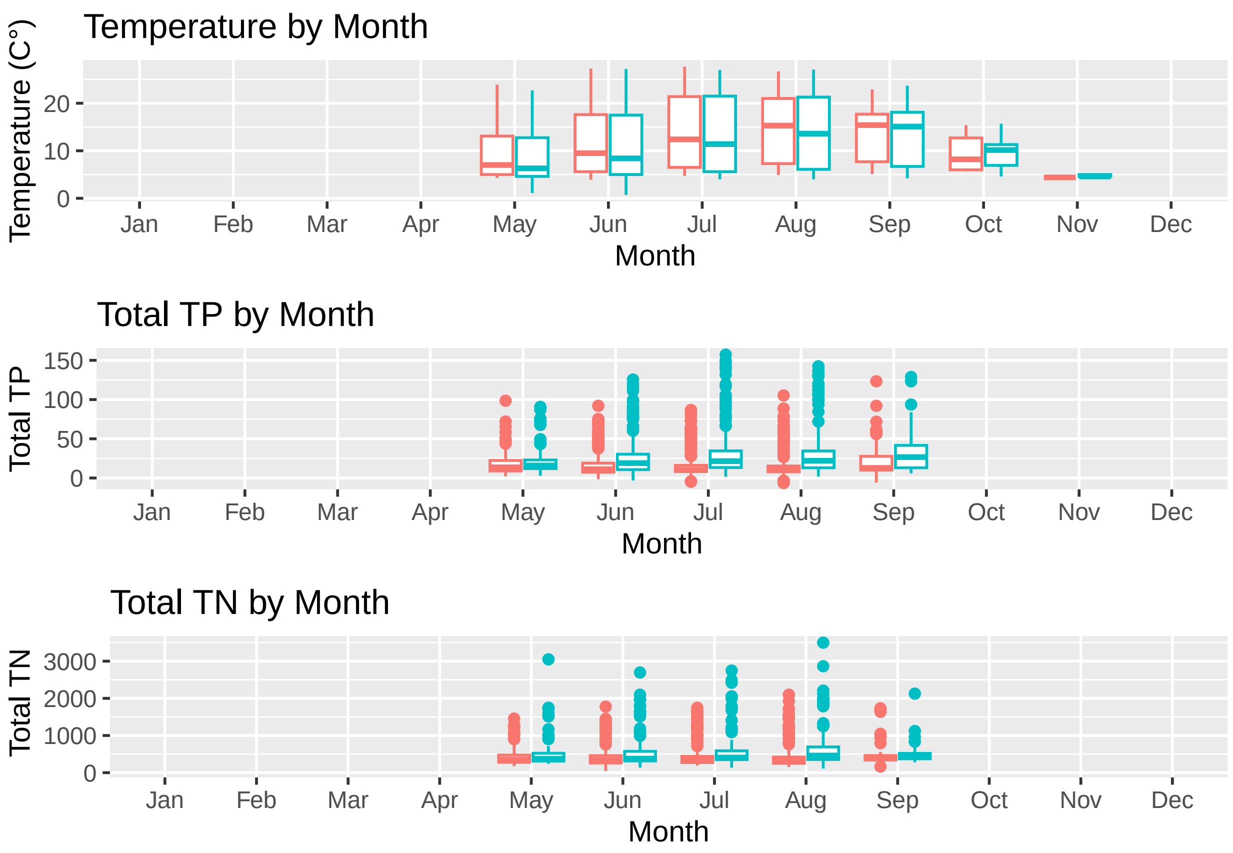
```
#Created cowplot with all three graphs and no legend
library(cowplot)
Cowplot <- plot_grid(Temp + theme(legend.position = "none"), TP + theme(legend.position = "none"), TN +
```

```
## 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()‘).
```

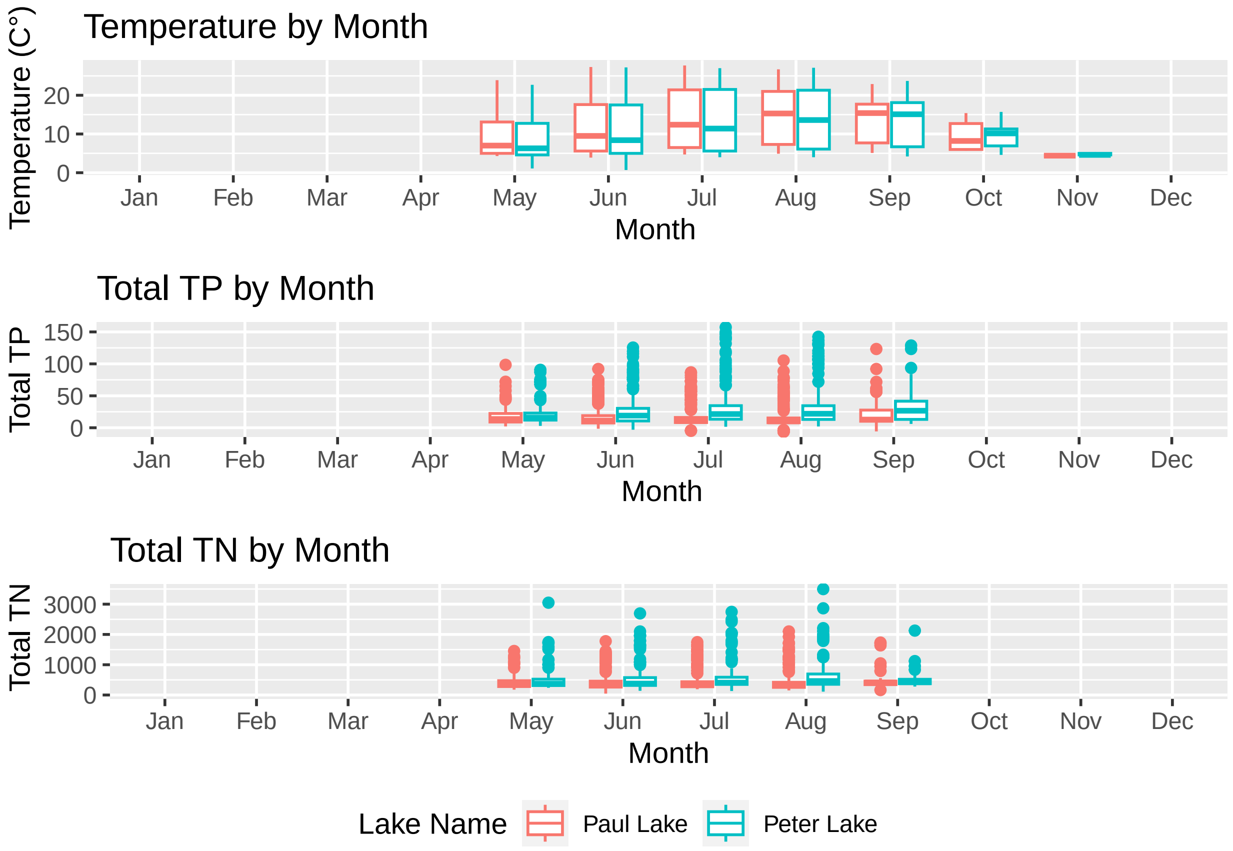
```
Cowplot
```



```
#Created a legend
legend <- get_legend(Temp + theme(legend.position="bottom"))
```

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

```
#Created a cowplot with a legend
Cowplot_legend <- plot_grid(Cowplot, legend, ncol=1, rel_heights = c(1, 0.1))
Cowplot_legend
```

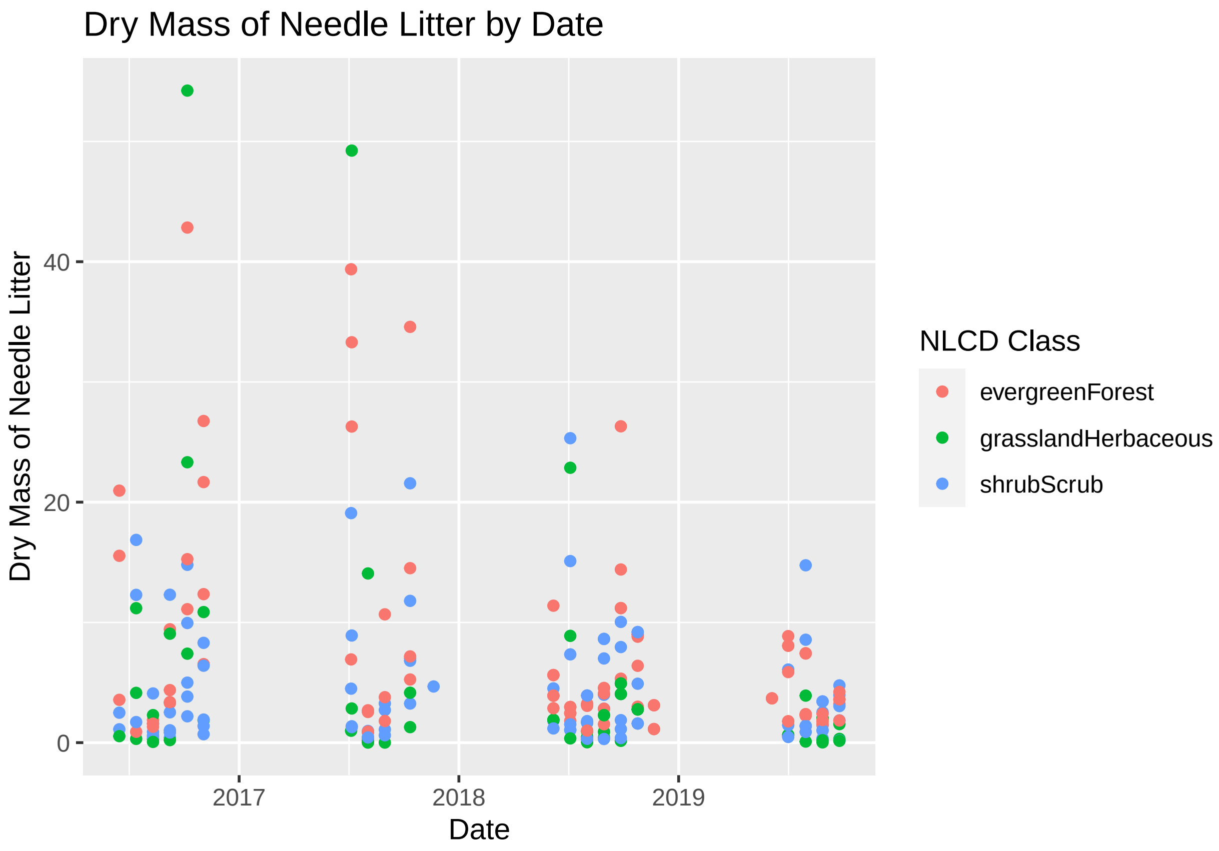


Question: What do you observe about the variables of interest over seasons and between lakes?

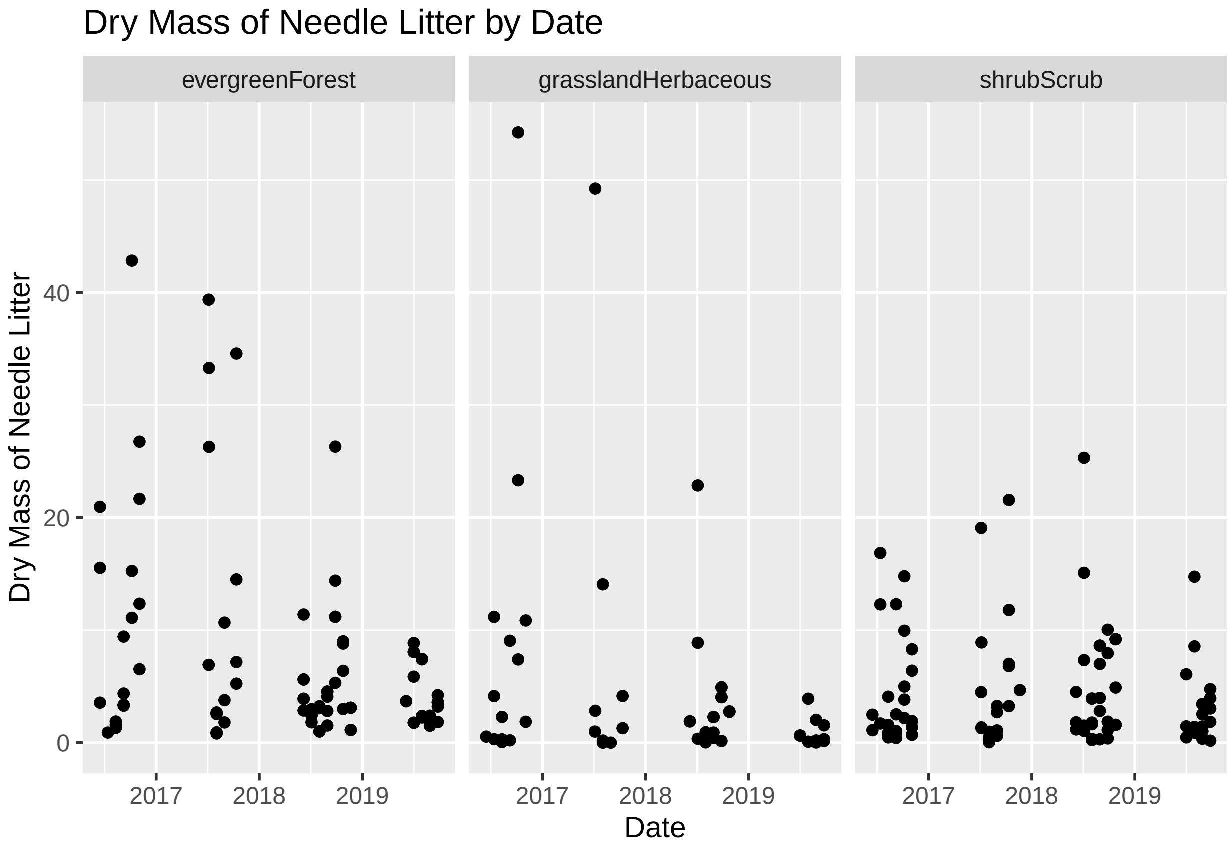
> Answer: Temperature in both lakes increases from May to July, and then decreases again, as expected. It also seems that the range in temperature is greater during the summer months. TP seems to increase overall throughout the year, and there is generally a greater amount of TP in Peter Lake than in Paul Lake. TN was overall the highest in August, although TN appears to be the most stable throughout the year. Peter Lake also appears to have higher amounts of TP than 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 Plotted Dry Mass of Needle Litter by Date
NeedleMass <- Litter %>%
  filter(functionalGroup == 'Needles') %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
    ) +
  geom_point() +
  labs(title="Dry Mass of Needle Litter by Date",
       x="Date",
       y="Dry Mass of Needle Litter",
       color="NLCD Class"
  )
NeedleMass
```



```
#7 Plotted Dry Mass of Needle Litter by Date, separated by NLCD Class
NeedleMass.by.Class <- Litter %>%
  filter(functionalGroup == 'Needles') %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass)
    ) +
  geom_point() +
  facet_wrap(~nlcdClass, ncol=3) +
  labs(title="Dry Mass of Needle Litter by Date",
       x="Date",
       y="Dry Mass of Needle Litter"
  )
NeedleMass.by.Class
```



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

Answer: I think that Plot 6 is more effective, because it was easier for me to directly compare the dry mass of needle litter betweeen the three classes when they are on the same plot, rather than when they are separated into different plots. I can see how Plot 7 would be helpful if you were more interested in seeing how the needle mass in each class changes over time, because Plot 6 has so many points close together that some may be obscured.