Assignment 5: Data Visualization

Sammy DiLoreto

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 setting up
#Load necessary packages
library(tidyverse);library(lubridate);library(here);library(cowplot)
library(viridis);library(RColorBrewer);library(colormap);library(ggthemes)
#Check working directory
getwd()
```

```
## [1] "/Users/sammydiloreto/Library/CloudStorage/Box-Box/ENV872-EDA/EDA-Spring2023"
```

```
#Upload data sets
processed_data = "Data/Processed_KEY"
NTL.LTER <- read.csv(
  here(processed_data,
```

```
      "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = T)
Litter <- read.csv(
  here(processed_data,
      "NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = T)

#2 Change date columns to date format
class(NTL.LTER$sampledate)
```

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

```
NTL.LTER$sampledate <- ymd(NTL.LTER$sampledate)
class(NTL.LTER$sampledate)
```

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

```
class(Litter$collectDate)
```

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

```
Litter$collectDate <- ymd(Litter$collectDate)
class(Litter$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
my_theme <- theme_base() +
  theme(
    plot.background = element_rect(
      linewidth = 1
      ),
    plot.title = element_text(
      size = rel(1),
      face = "bold"
      ),
    panel.background = element_rect(
      fill = NA
      ),
```

```
    panel.grid = element_line(
      color = "gray80",
      linewidth = 0.25
      ),
    axis.title = element_text(
      size = rel(0.7),
      face = "bold"
      ),
    axis.text = element_text(
        size = rel(0.5),
      face = "bold"
      ),
    axis.line = element_line(
      linewidth = 0.7
      ),
    legend.title = element_text(
      size = rel(0.7),
      face = "bold"
      ),
    legend.text = element_text(
      size = rel(0.6)
      ),
    legend.background = element_rect(
      fill = "gray95"
        ),
    legend.position = 'bottom',
    complete = TRUE
  )
```

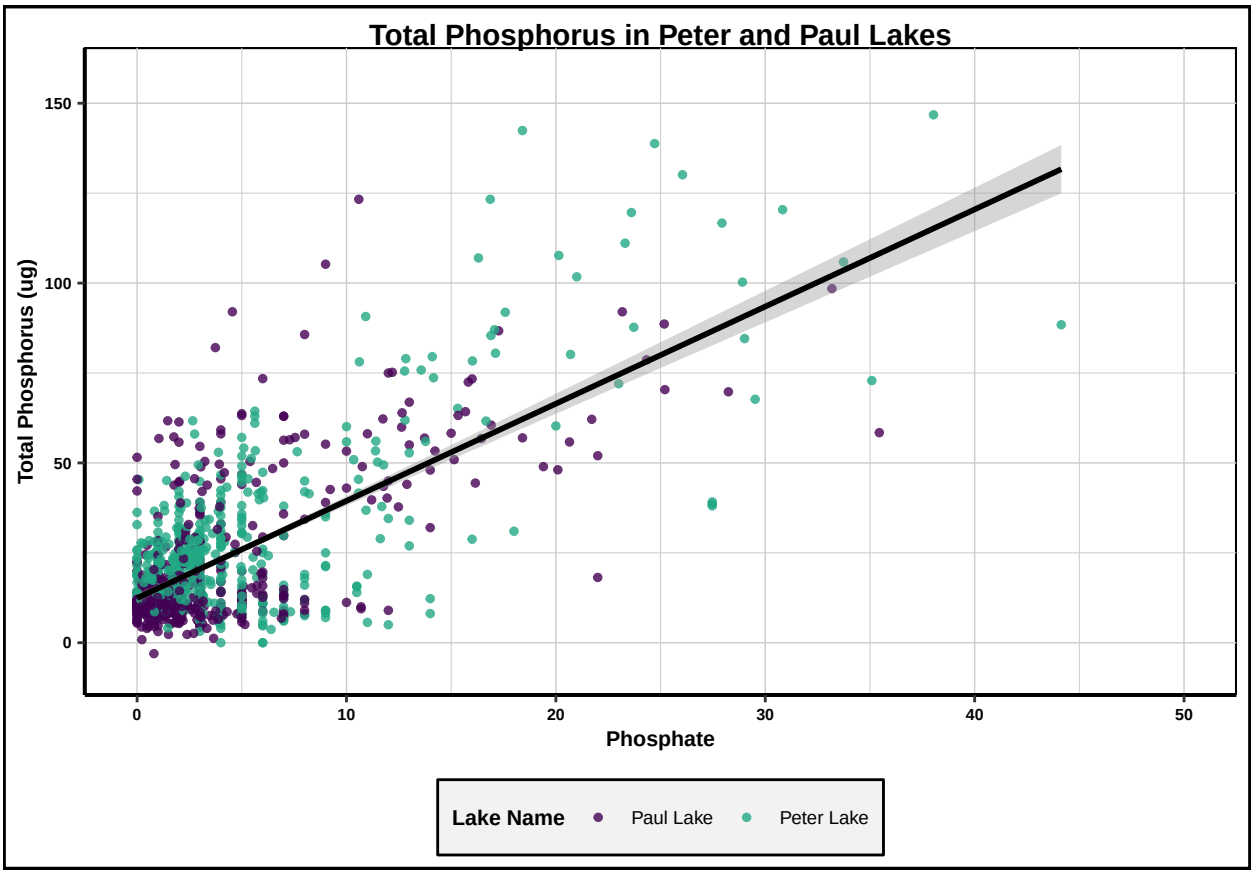
Create graphs

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
totalP_plot <- NTL.LTER %>%
  ggplot(
    mapping = aes(
      x=po4,
      y=tp_ug,
      color=lakename)
    ) +
  geom_point(size=1, alpha=0.8)+
  xlim(0,50)+
  geom_smooth(
    method = "lm",
    color = "black")+
  scale_color_viridis(
```

```
    discrete = TRUE,
    end = 0.6)+
  labs(
    x = "Phosphate" ,
    y = "Total Phosphorus (ug)",
    title = "Total Phosphorus in Peter and Paul Lakes",
    color = "Lake Name")+
  my_theme
totalP_plot
```

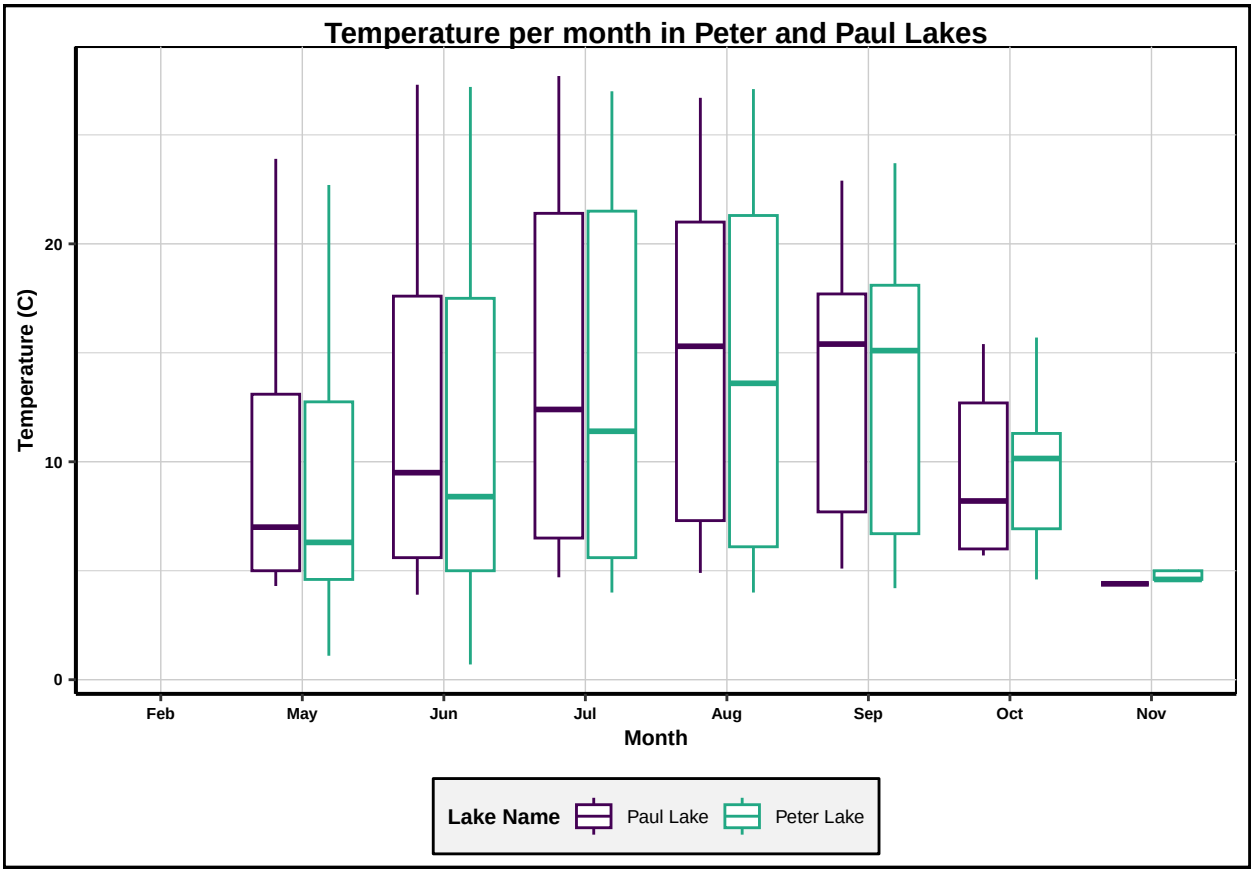


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

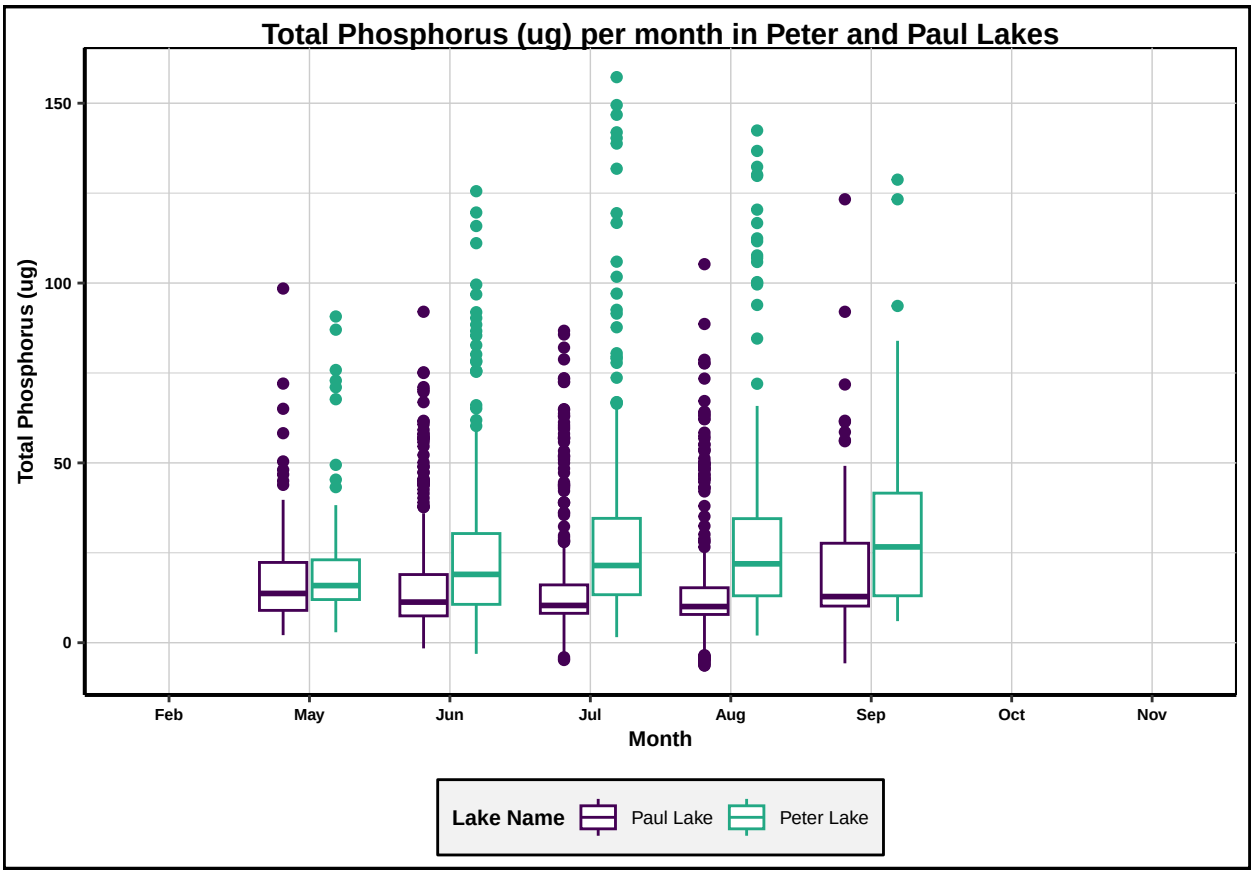
```
#5
#temperature plot
temp_plot <- NTL.LTER %>%
  ggplot(
    mapping = aes(
      x=factor(
         month,
         levels = 1:12,
```

```
        labels = month.abb),
      y=temperature_C,
      color = lakename)
    ) +
  geom_boxplot()+
  scale_color_viridis(
    discrete = TRUE,
    end = 0.6) +
  labs(
    x = "Month" ,
    y = "Temperature (C)",
    title = "Temperature per month in Peter and Paul Lakes",
    color = "Lake Name"
  )+
  my_theme
temp_plot
```



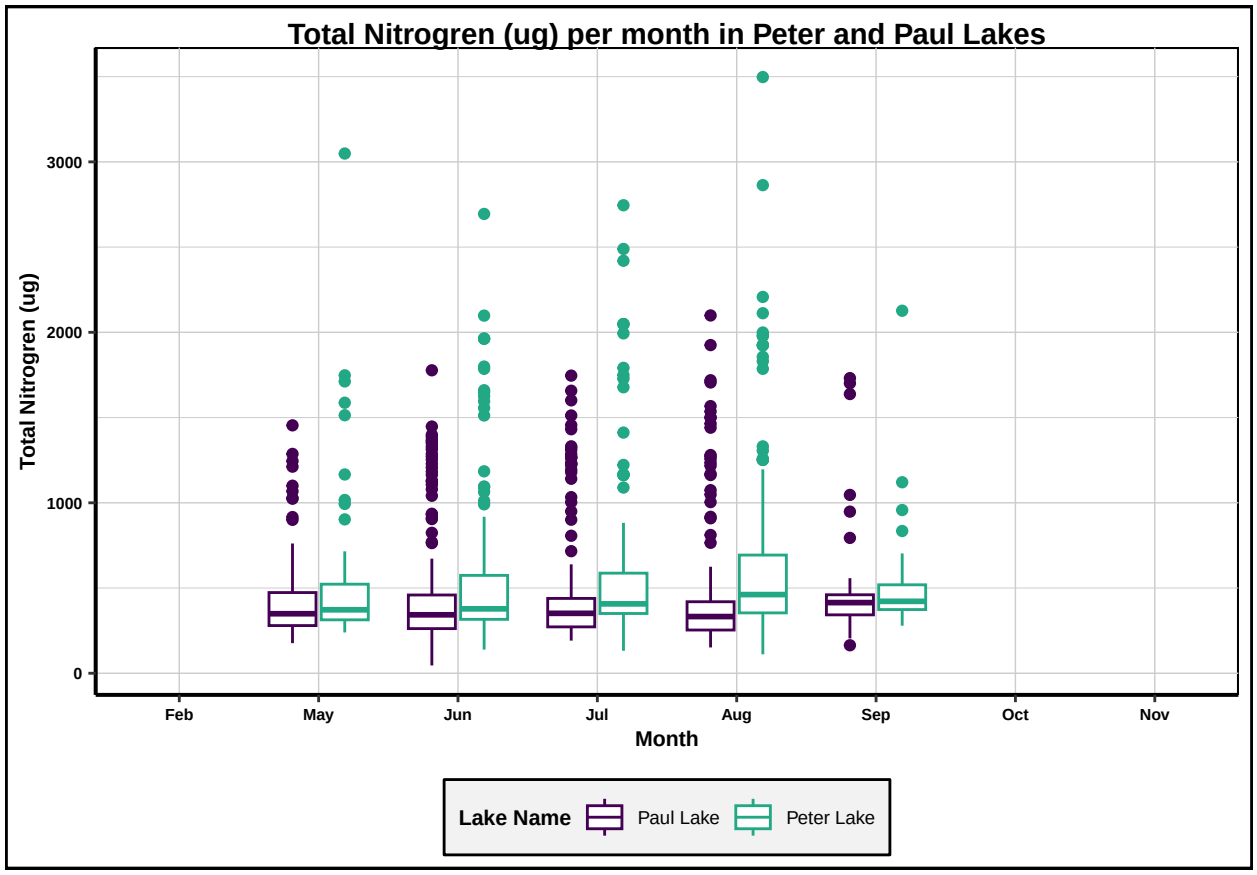

```
#TP plot
TP_plot <- NTL.LTER %>%
  ggplot(
    mapping = aes(
      x=factor(
        month,
        levels = 1:12,
        labels = month.abb),
```

```
      y=tp_ug,
      color = lakename)
    ) +
  geom_boxplot()+
  scale_color_viridis(
    discrete = TRUE,
    end = 0.6) +
  labs(
    x = "Month" ,
    y = "Total Phosphorus (ug)",
    title = "Total Phosphorus (ug) per month in Peter and Paul Lakes",
    color = "Lake Name"
  )+
  my_theme
TP_plot
```



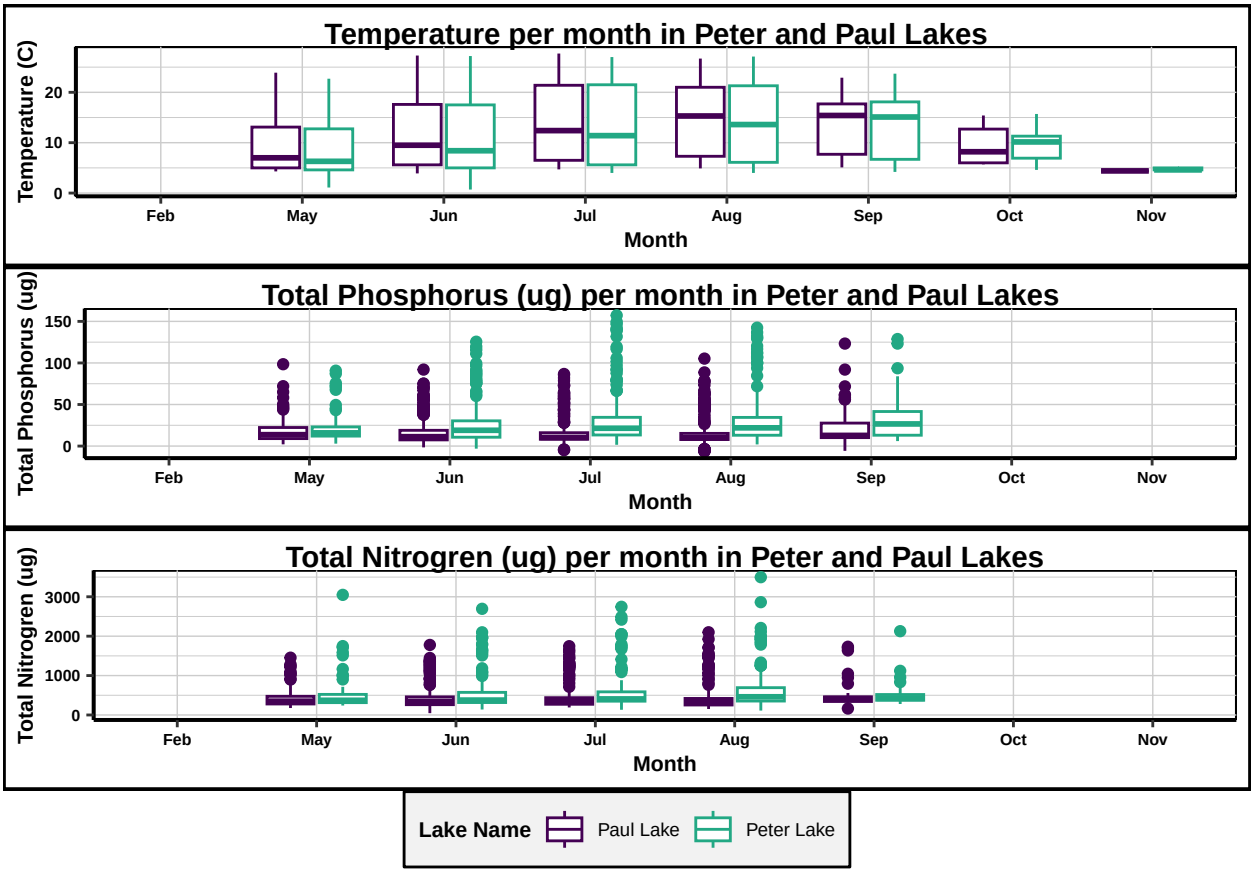

```
#TN plot
TN_plot <- NTL.LTER %>%
  ggplot(
    mapping = aes(
      x=factor(
         month,
         levels = 1:12,
         labels = month.abb),
      y=tn_ug,
```

```
      color = lakename)
    ) +
  geom_boxplot()+
  scale_color_viridis(
    discrete = TRUE,
    end = 0.6) +
  labs(
    x = "Month" ,
    y = "Total Nitrogren (ug)",
    title = "Total Nitrogren (ug) per month in Peter and Paul Lakes",
    color = "Lake Name"
  )+
  my_theme
TN_plot
```




```
#combine the three plots, removing the legends
combined_plots <- plot_grid(
  temp_plot + theme(legend.position="none"),
  TP_plot + theme(legend.position="none"),
  TN_plot + theme(legend.position="none"),
  nrow = 3)
#get the legend
legend<- get_legend(
  temp_plot +
    guides(color = guide_legend(nrow = 1)) +
```

```
    theme(legend.position = "bottom")
)
# add the legend underneath the row we made earlier. Give it 10%
# of the height of one plot (via rel_heights).
temp_TP_TN_plot <- plot_grid(
  combined_plots,
  legend,
  ncol = 1,
  rel_heights = c(1, .1))
temp_TP_TN_plot
```



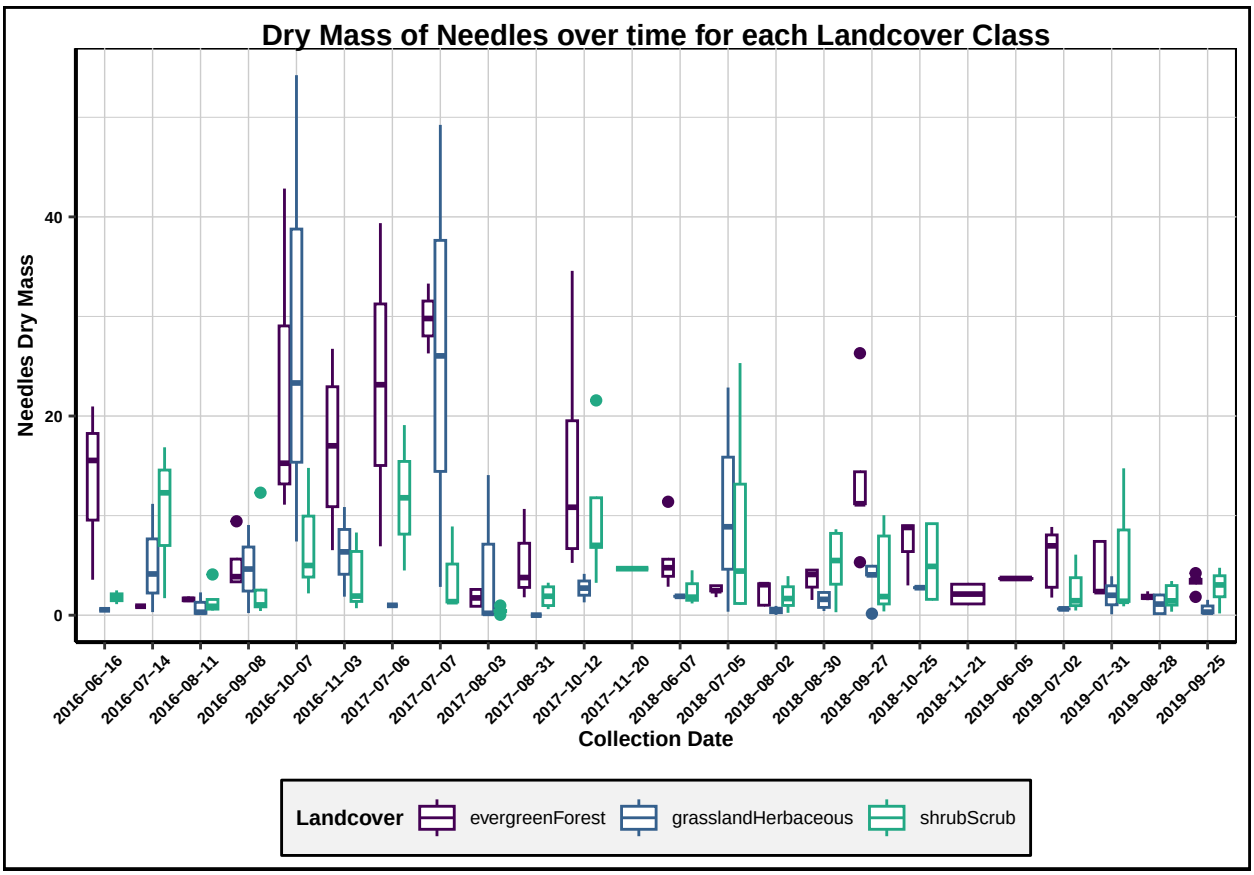
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature medians for both lakes are higher in July, August, and September and lower in the surrounding months, with the lowest in November. This makes sense in terms of seasonal air temperatures. Between lakes there is not much differences in temperatures medians except in October, Peter Lake appears to have a much a higher median. Most other months, Paul Lake has a slightly higher median than Peter Lake, with a somewhat larger difference in August. For both total phosphorus and total nitrogen, there are many outliers in each month for both lakes. Peter Lake always has higher medians than Paul Lake for both phosphorus and nitrogen. The warmer months have higher total phosphorus. The same trend appears for nitrogen but the variation between months is not as large.

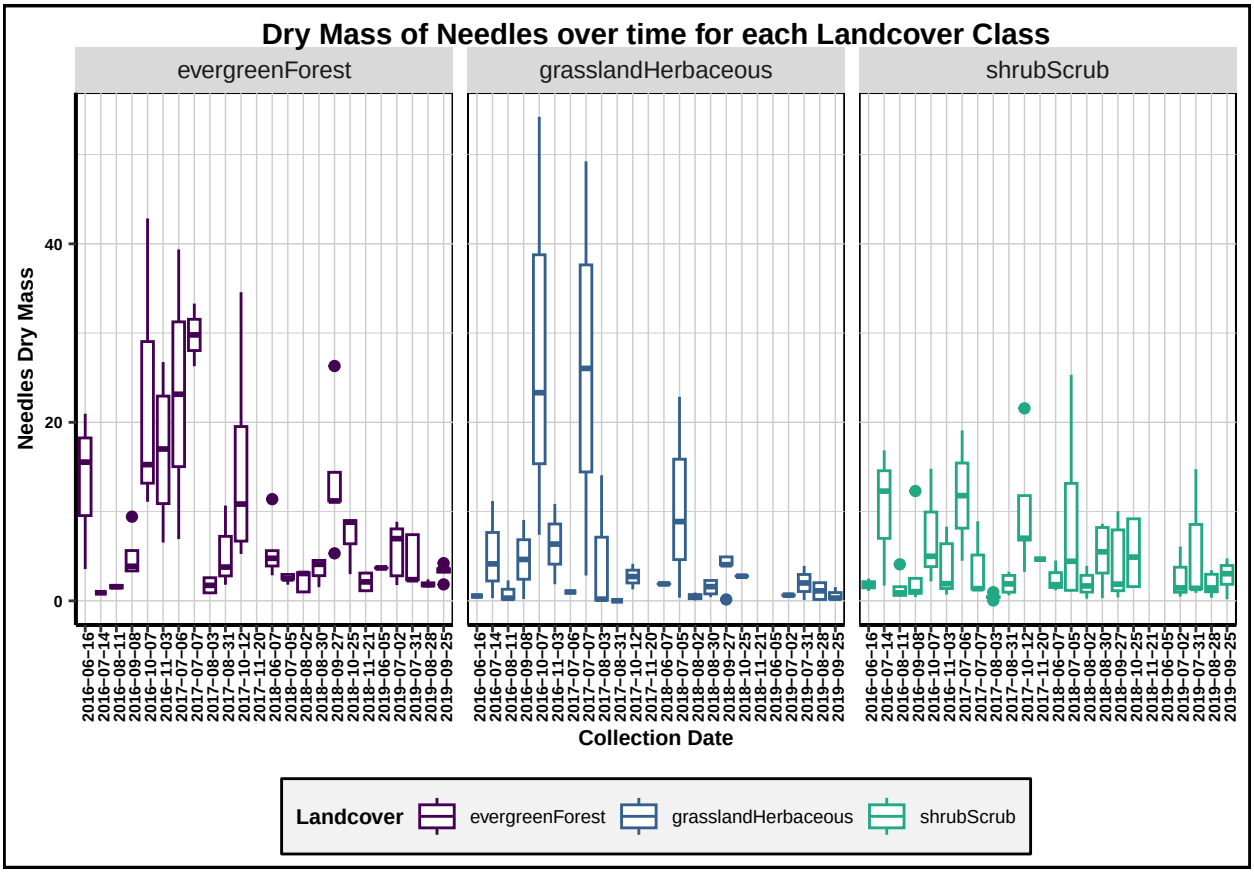
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
Needles_plot <- Litter %>%
  filter(functionalGroup %in% c("Needles")) %>%
  ggplot(
    mapping = aes(
      x=factor(collectDate),
      y=dryMass,
      color= nlcdClass)
    ) +
  geom_boxplot()+
  scale_color_viridis(
    discrete = TRUE,
    end = 0.6) +
  labs(
    x = "Collection Date" ,
    y = "Needles Dry Mass",
    title = "Dry Mass of Needles over time for each Landcover Class",
    color = "Landcover"
  )+
  my_theme +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.75, hjust = 0.8))
Needles_plot
```



```r
#7
Needles_plot2 <- Litter %>%
  filter(functionalGroup %in% c("Needles")) %>%
  ggplot(
    mapping = aes(
      x=factor(collectDate),
      y=dryMass,
      color = nlcdClass)
    ) +
  geom_boxplot()+
  facet_wrap(vars(nlcdClass))+
  scale_color_viridis(
    discrete = TRUE,
    end = 0.6) +
  labs(
    x = "Collection Date" ,
    y = "Needles Dry Mass",
    title = "Dry Mass of Needles over time for each Landcover Class",
    color = "Landcover"
  )+
  my_theme +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.75, hjust = 0.75))
Needles_plot2
```



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

Answer: Both plots show dry mass of needles over time for each land cover type, evergreen forest, grassland herbaceous, and shrub scrub. Plot 6 shows each boxplots for the different landcover types at each collection date whereas plot 7 splits up the graphs into three facets for each land cover type. Plot 7 is more effective because when separated out by landcover type, you can make better comparisons for needle dry mass over time. You can also still compare needle dry mass over time between landcover types. For example, in plot 7 you can more easily see that for both evergreen and grasslands, needle dry mass has decreased over time and for shrub scrub, dry mass has gone up and down. You can also see that evergreen and grasslands have higher medians for a few collection dates than most of shrub scrub. That information is very difficult to see on plot 6. One thing plot 6 does show well is which colection dates had the highest medians and largest range of dry mass, this can be seen on plot 7 but because the dates are more squished on the x-axis, it is not as easy to detect.