Assignment 5: Data Visualization

Keanu Valibia

Spring 2024

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

# Load necessary packages
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
```

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

```
## here() starts at /home/guest/R/R Projects/EDA_Spring2024
```

```
library(cowplot)
```

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

```
# Load appropriate .csv packages
ntl.lter.nutrients <- read.csv(here("~/R/R Projects/EDA_Spring2024/Data/Processed/NTL-LTER_Lake_Chemist
peter.paul.processed <- read.csv(here("~/R/R Projects/EDA_Spring2024/Data/Processed/NEON_NIWO_Litter_ma

#2

# Format date fields
ntl.lter.nutrients$sampledate <- as.Date(ntl.lter.nutrients$sampledate,
                                         format = "%Y-%m-%d")
peter.paul.processed$collectDate <- as.Date(peter.paul.processed$collectDate, format = "%Y-%m-%d")
```

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

# Create custom theme
customtheme <- theme_classic(base_size = 11) +
  theme(axis.text = element_text(color = "black"),
        panel.background = element_rect(fill ="#EDE6E3"),
        panel.grid.major = element_line(color = "#36382E",
                                        linetype = "dotted"),
        plot.title = element_text(size = 15),
        axis.title.x = element_text(size = 13),
        axis.title.y = element_text(size = 13),
        legend.position = "right")
```

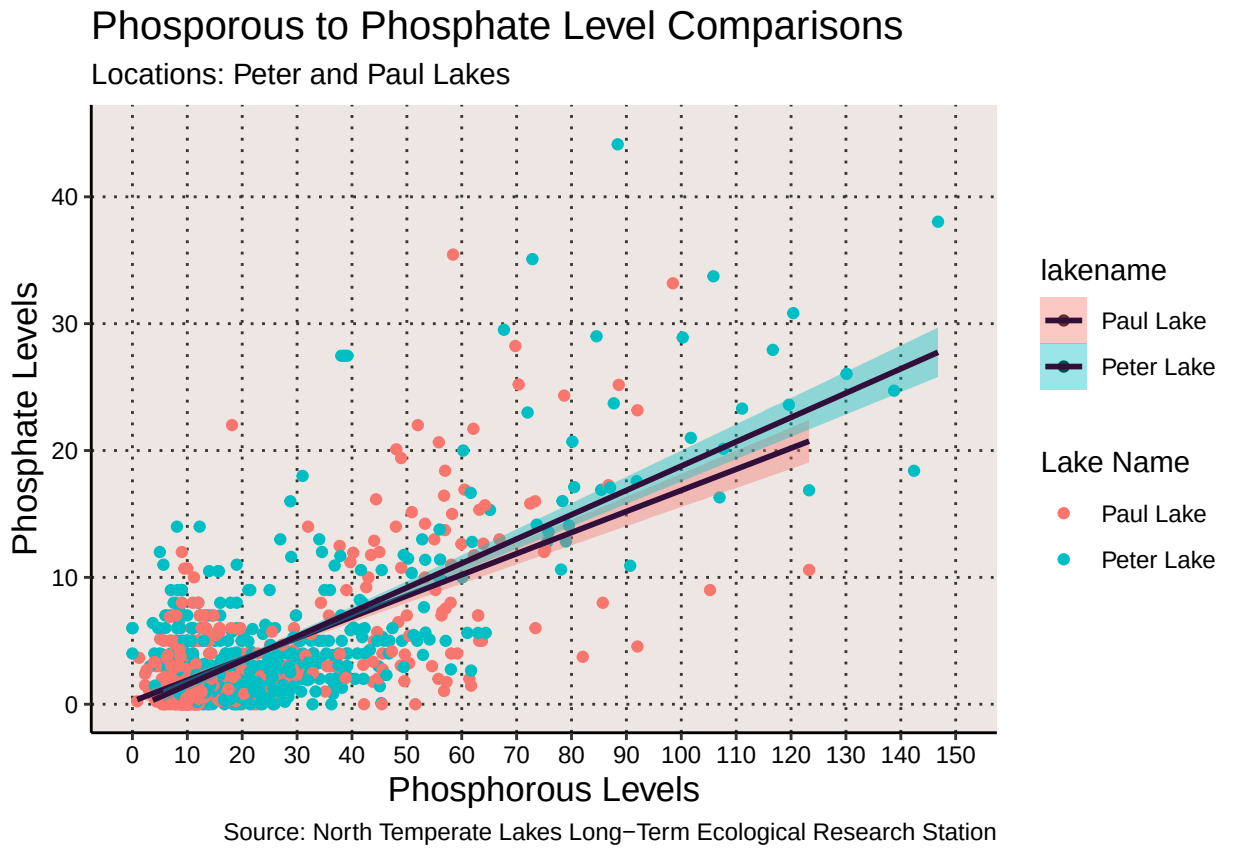
Create graphs

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 line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

# Build P vs. PO4 plot
ntl.lter.phosphorousPlot <- ggplot(ntl.lter.nutrients, aes(x = tp_ug,
                                      y=po4,
                                      color = lakename,
                                      fill = lakename)) +
  geom_point() +
  labs(title = "Phosporous to Phosphate Level Comparisons",
       subtitle = "Locations: Peter and Paul Lakes",
       caption = "Source: North Temperate Lakes Long-Term Ecological Research Station",
       color = "Lake Name") +
  scale_x_continuous(name="Phosphorous Levels", breaks=seq(0, 150, 10), limits=c(0,150)) +
  scale_y_continuous(name="Phosphate Levels", breaks=seq(0, 50, 10), limits=c(0,45)) +
  stat_smooth(method = "lm", col = "#320E3B") +
  customtheme

# Call plot
ntl.lter.phosphorousPlot
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21948 rows containing missing values (`geom_point()`).

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



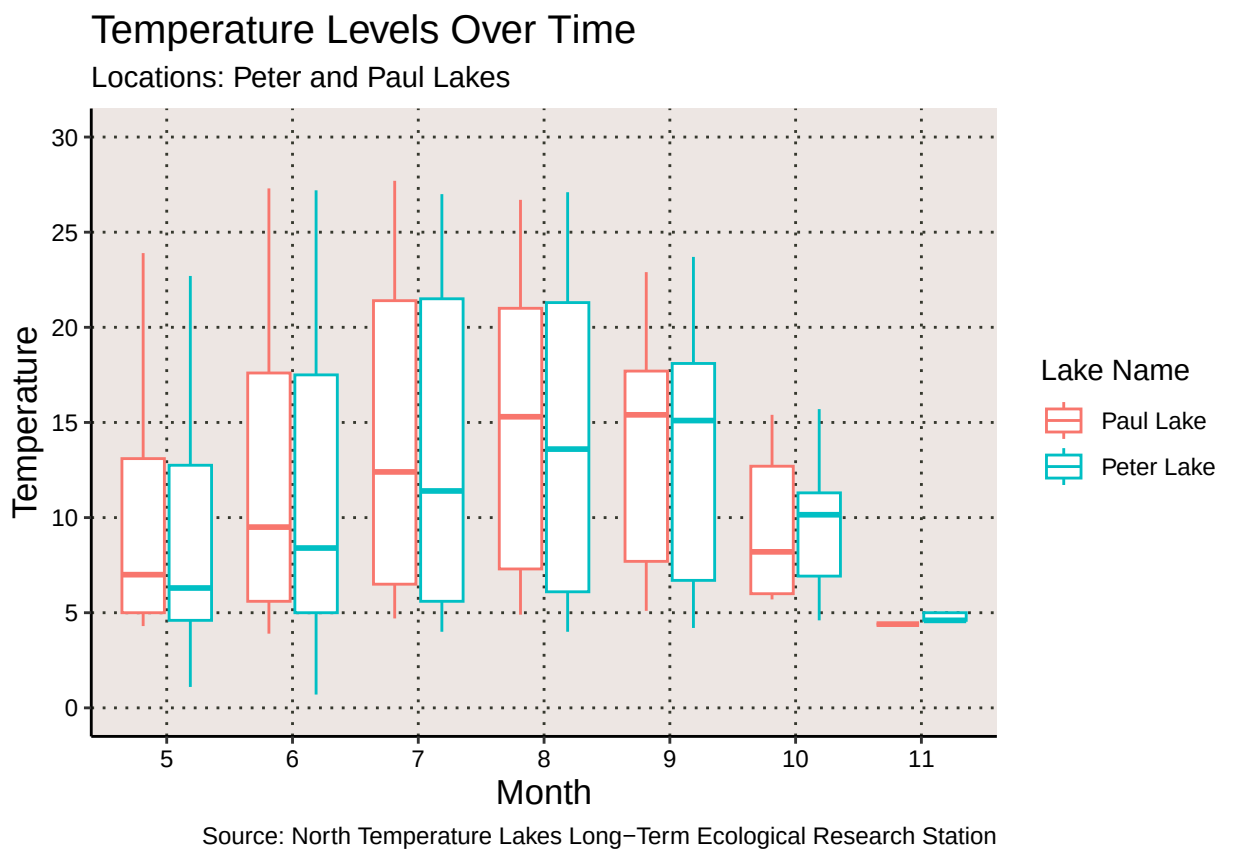
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5

# Building Temperature box plot
ntl.lter.boxPlot.temp <- ggplot(ntl.lter.nutrients, aes(x = as.factor(month), y = temperature_C)) +
  geom_boxplot(aes(color = lakename )) +
  labs(title ="Temperature Levels Over Time",
       subtitle = "Locations: Peter and Paul Lakes",
       caption = "Source: North Temperature Lakes Long-Term Ecological Research Station",
       color = "Lake Name") +
  xlab("Month") +
  scale_x_discrete(limits=c("5" , "6" , "7" , "8" , "9" , "10" , "11")) +
  scale_y_continuous(name="Temperature", breaks=seq(0, 30, 5), limits=c(0,30)) +
  customtheme

ntl.lter.boxPlot.temp
```

```
## Warning: Removed 16 rows containing missing values (‘stat_boxplot()‘).
```

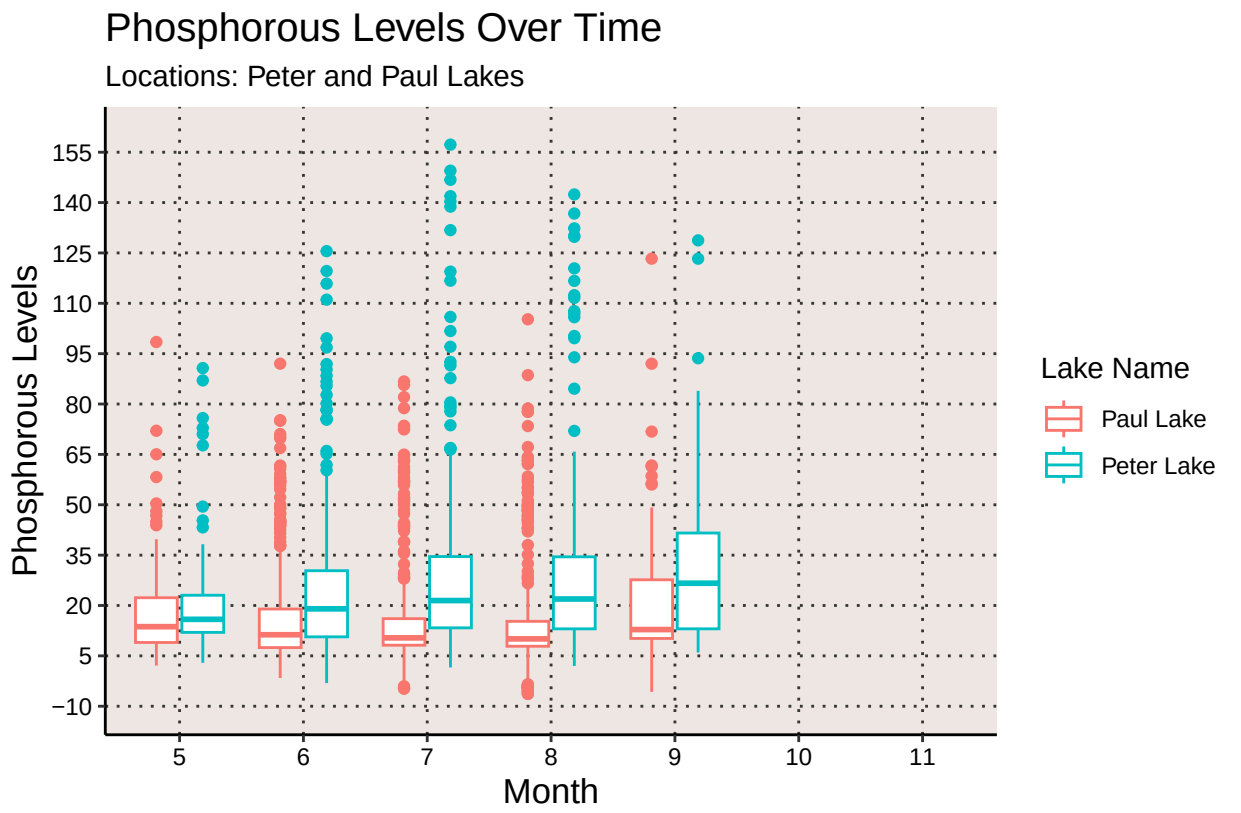
```
## Warning: Removed 3550 rows containing non-finite values (`stat_boxplot()`).
```




```
# Building Phosphorous box plot
ntl.lter.boxPlot.tp <- ggplot(ntl.lter.nutrients, aes(x = as.factor(month),
                                                      y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(title ="Phosphorous Levels Over Time",
       subtitle = "Locations: Peter and Paul Lakes",
       caption = "Source: North Temperature Lakes Long-Term Ecological Research Station",
       color = "Lake Name") +
  xlab("Month") +
    scale_x_discrete(limits=c("5" , "6" , "7" , "8" , "9" , "10" , "11")) +
  scale_y_continuous(name="Phosphorous Levels", breaks=seq(-10, 160, 15), limits=c(-10,160)) +
  customtheme

ntl.lter.boxPlot.tp
```

```
## Warning: Removed 16 rows containing missing values (`stat_boxplot()`).
```

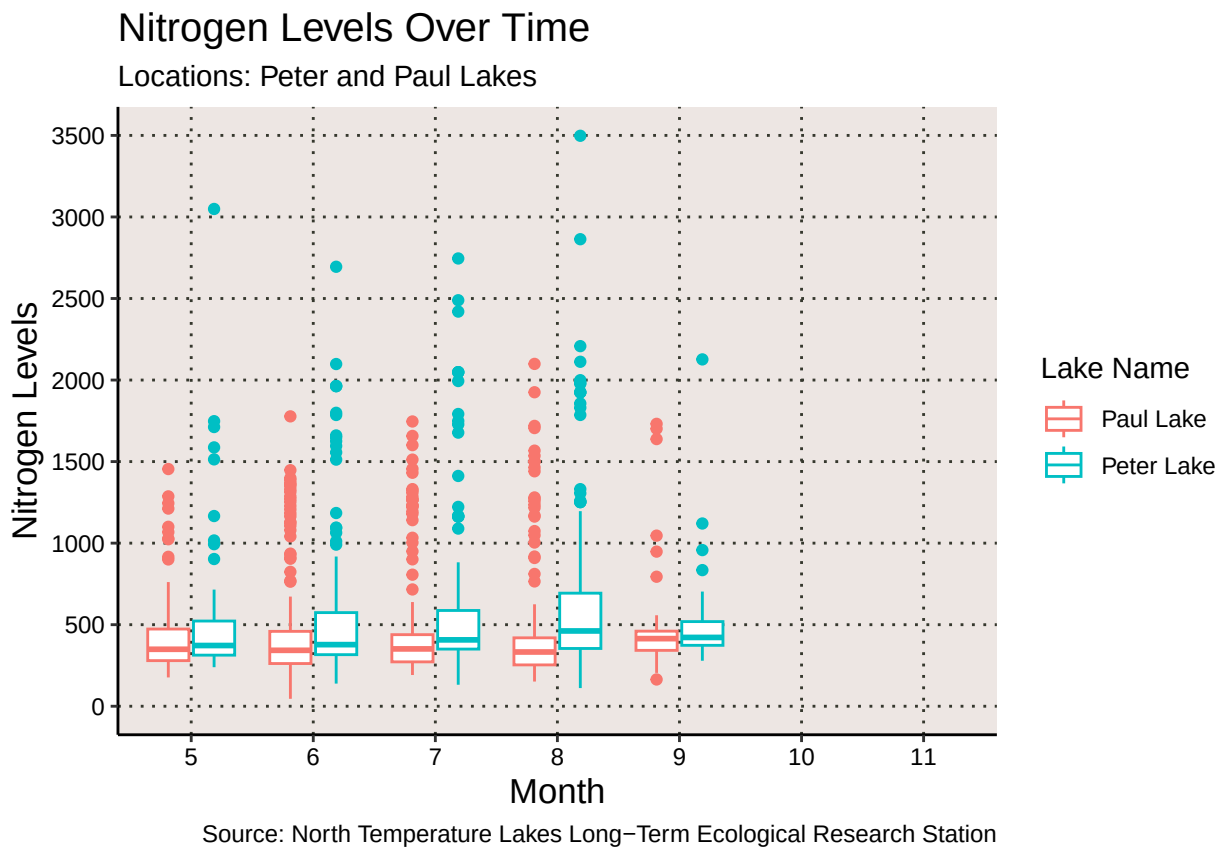
```
## Warning: Removed 20713 rows containing non-finite values (`stat_boxplot()`).
```




```
# Building Nitrogen box plot

ntl.lter.boxPlot.tn <- ggplot(ntl.lter.nutrients, aes(x = as.factor(month),
                                                      y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  labs(title ="Nitrogen Levels Over Time",
       subtitle = "Locations: Peter and Paul Lakes",
       caption = "Source: North Temperature Lakes Long-Term Ecological Research Station",
       color = "Lake Name") +
  xlab("Month") +
  scale_x_discrete(limits=c("5" , "6" , "7" , "8" , "9" , "10" , "11")) +
  scale_y_continuous(name="Nitrogen Levels", breaks=seq(0, 3500, 500), limits=c(0,3500)) +
  customtheme

ntl.lter.boxPlot.tn
```

```
## Warning: Removed 16 rows containing missing values (‘stat_boxplot()‘).
```

```
## Warning: Removed 21567 rows containing non-finite values (‘stat_boxplot()‘).
```




```
# Building cowplot

cowPlot.legend <- get_legend(ntl.lter.boxPlot.temp)
```

```
## Warning: Removed 16 rows containing missing values (‘stat_boxplot()‘).
```

```
## Warning: Removed 3550 rows containing non-finite values (‘stat_boxplot()‘).
```

```
cowPlot.grid <- plot_grid(ntl.lter.boxPlot.temp +
                            theme(axis.title.x = element_blank(),
                                  legend.position = "none",
                                  plot.caption = element_blank(),
                                  plot.margin = unit(c(0,0,0,0),"cm")) +
                          labs(title = "Temperature, Phosphorous, and Nitrogen Levels Over Time"),
                                ntl.lter.boxPlot.tp +
                                  theme(axis.title.x = element_blank(),
                                        plot.title = element_blank(),
                                        title = element_blank(),
                                        legend.position = "none",
                                        plot.margin = unit(c(0,0,0,0),"cm")),
                                ntl.lter.boxPlot.tn +
                                  theme(title = element_blank(),
                                        plot.title = element_blank(),
                                        legend.position = "none",
                                        plot.margin = unit(c(0,0,0,0),"cm")),
```

```
                          nrow = 3,
                          rel_heights = c(1,1,1),
                          align = 'hv') +
  customtheme
```

```
## Warning: Removed 16 rows containing missing values (`stat_boxplot()`).
## Removed 3550 rows containing non-finite values (`stat_boxplot()`).
```

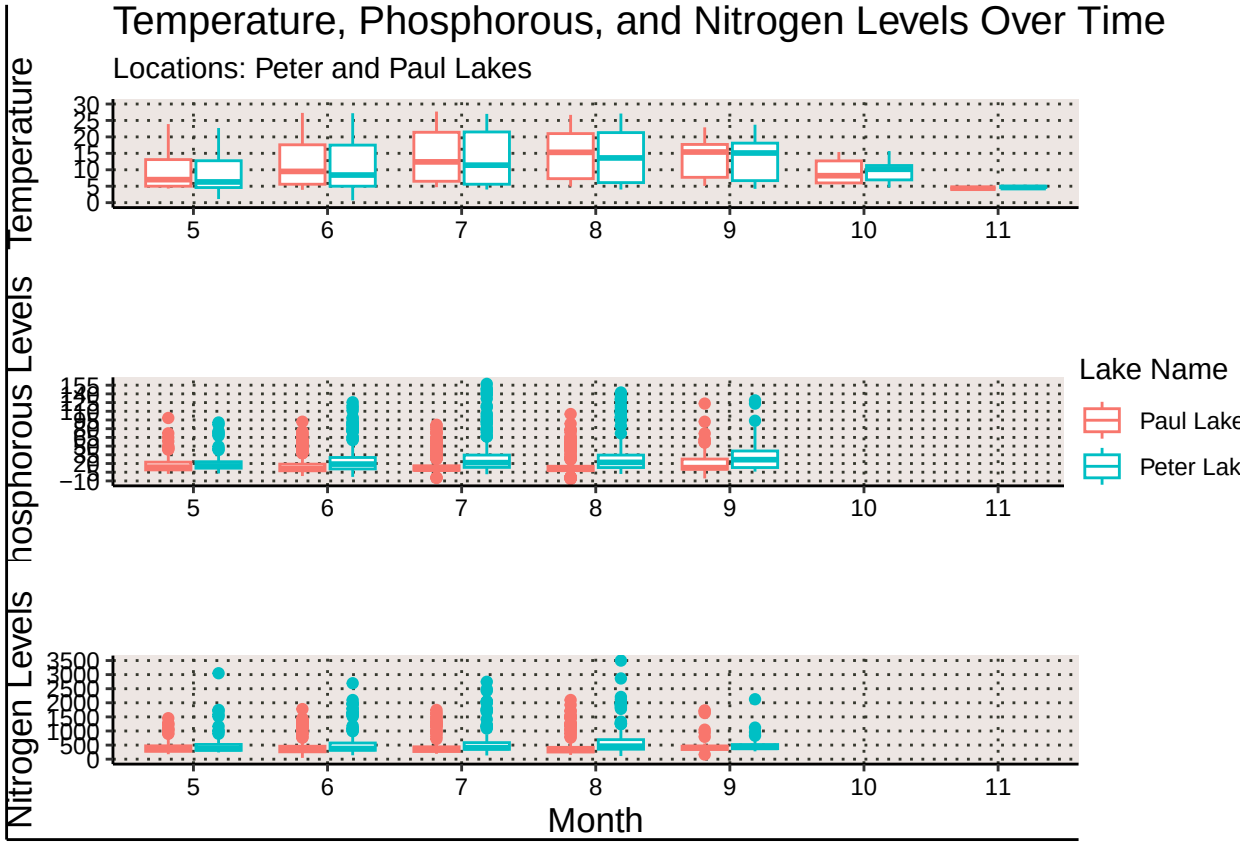
```
## Warning: Removed 16 rows containing missing values (`stat_boxplot()`).
```

```
## Warning: Removed 20713 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 16 rows containing missing values (`stat_boxplot()`).
```

```
## Warning: Removed 21567 rows containing non-finite values (`stat_boxplot()`).
```

```
ntl.combined.cowPlot <- plot_grid(cowPlot.grid, cowPlot.legend,
                                  rel_widths = c(3, .4),
                                  scale = 1)

ntl.combined.cowPlot
```



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

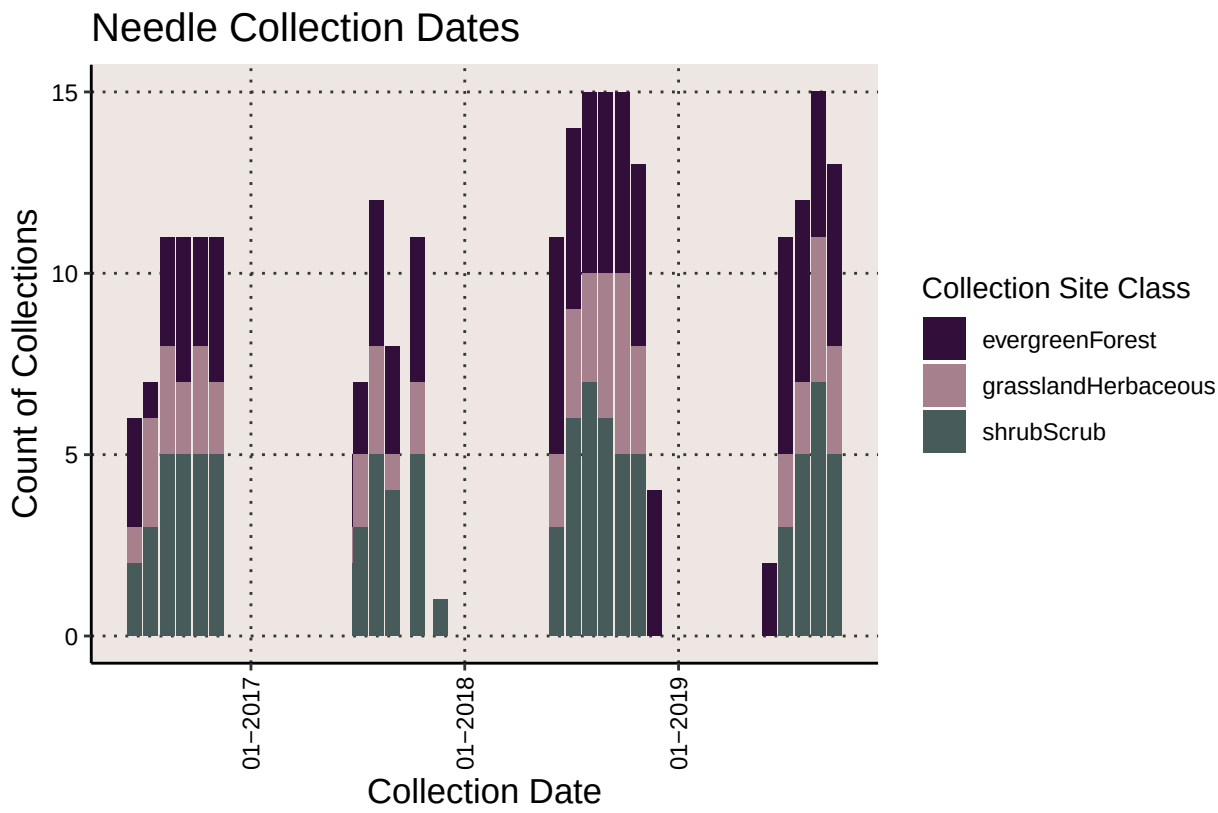
Answer: Temperature medians, IQRs, and ranges are generally similar across the year. However Peter Lake consistently has higher levels of both Phosphorous and Nitrogen, with higher outliers, wider IQRs, and larger ranges.

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.

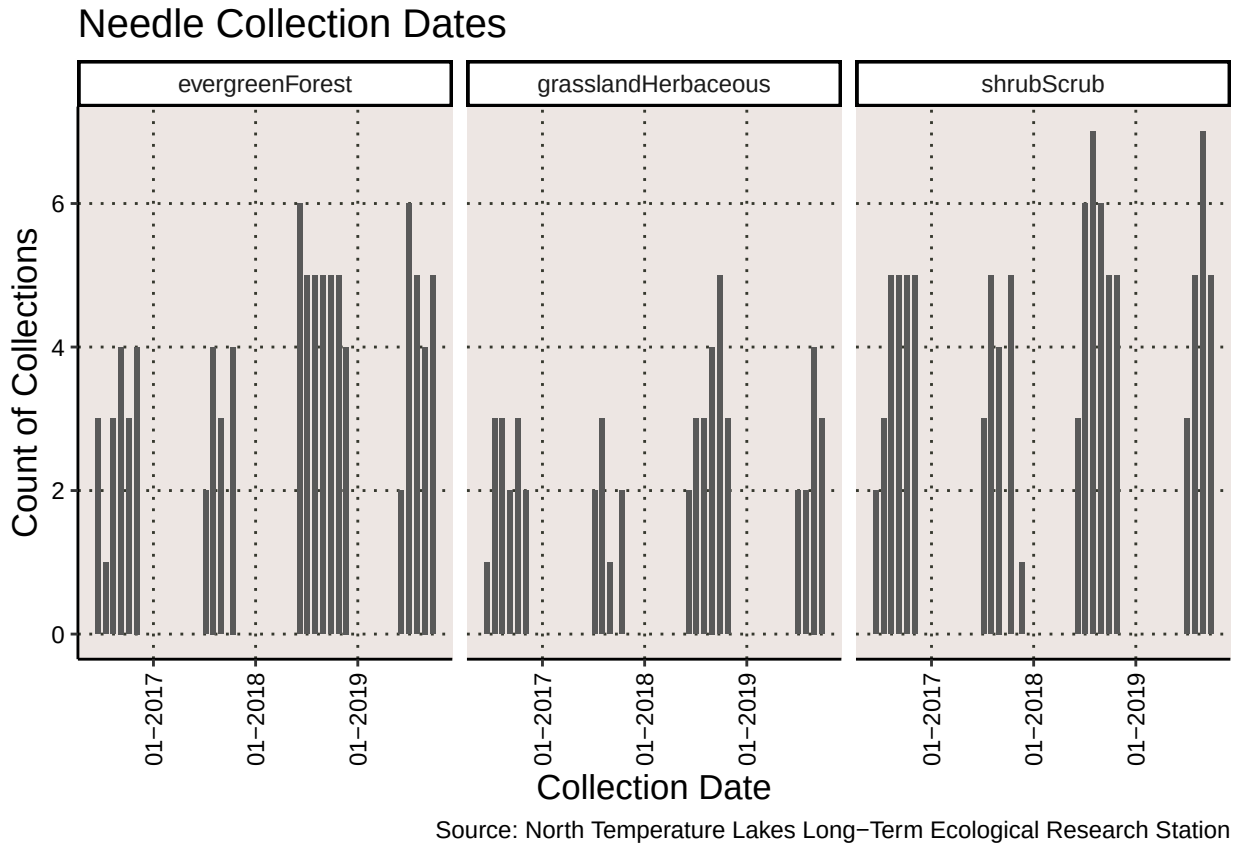
```
#6

niwot.plot <- ggplot(subset(peter.paul.processed, functionalGroup %in% "Needles"),
                     aes(x = collectDate)) +
  geom_bar(width = 25, aes(fill = nlcdClass)) +
  labs(title ="Needle Collection Dates",
     caption = "Source: North Temperature Lakes Long-Term Ecological Research Station",
     fill = "Collection Site Class") +
  xlab("Collection Date") +
  ylab("Count of Collections") +
  scale_x_date(date_breaks = "years", date_labels = "%m-%Y") +
  customtheme +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_color_manual(values=c("#320E3B", "#A6808C", "#475B5A")) +
  scale_fill_manual(values=c("#320E3B", "#A6808C", "#475B5A"))

niwot.plot
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
```




```
#7

niwot.facetPlot <- ggplot(subset(peter.paul.processed, functionalGroup %in% "Needles"),
                          aes(x = collectDate)) +
  geom_bar(width = 20) +
  labs(title ="Needle Collection Dates",
     caption = "Source: North Temperature Lakes Long-Term Ecological Research Station") +
  xlab("Collection Date") +
  ylab("Count of Collections") +
  scale_x_date(date_breaks = "years", date_labels = "%m-%Y") +
  customtheme +
  facet_wrap(~ nlcdClass, scales="free_x", drop = TRUE) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_color_manual(values=c("#320E3B", "#A6808C", "#475B5A")) +
  scale_fill_manual(values=c("#320E3B", "#A6808C", "#475B5A"))

niwot.facetPlot
```

```
## Warning: `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
## `position_stack()` requires non-overlapping x intervals
```



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

Answer: The facted grid (#7) is more effective. Having three facets allows the user to more easily identify the different functional groups. Additionally, the months that had no values are removed from the grid.