Assignment 5: Data Visualization

Iddrisu Sharu Deen

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.

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

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     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
```

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

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

```
library(cowplot)
```

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

```
NTL_LTER_data <- read.csv(
  "./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
  stringsAsFactors = TRUE)

NIWOT_LTER_data <- read.csv(
  "./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

#2
NTL_LTER_data$sampledate <- as.Date(NTL_LTER_data$sampledate, format = "%Y-%m-%d")
NIWOT_LTER_data$collectDate <- as.Date(NIWOT_LTER_data$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

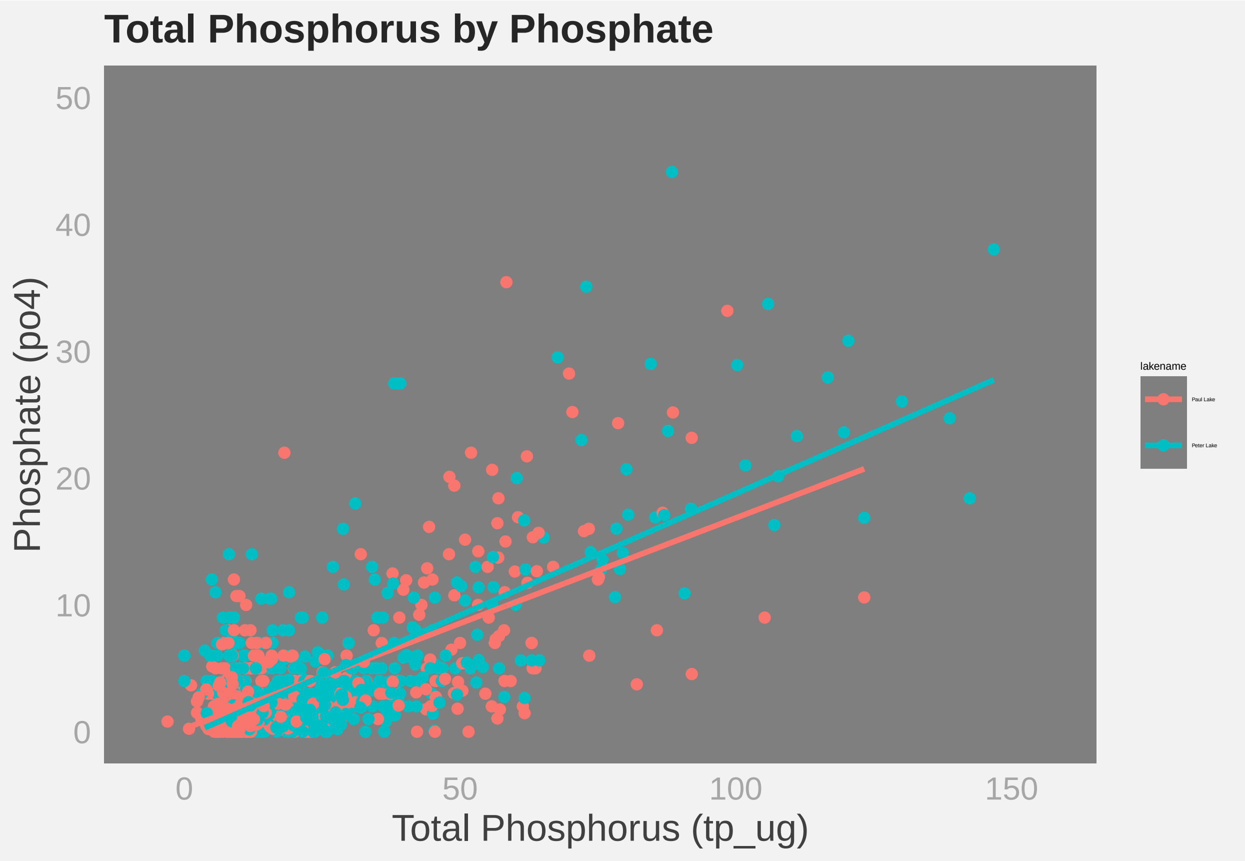
```
#3
mytheme <- theme_dark() +
  theme(
    line = element_blank(),
    rect = element_rect(fill = "grey95"),
    text = element_text(size = rel(2)),
    plot.background = element_rect(),
    plot.title = element_text(size = 15, color = "grey15", face = "bold"),
    axis.text = element_text(size = 12, color = "gray65"),
    axis.title = element_text(size = 14, color = "gray25")

  )

mytheme2 <- theme_dark() +
  theme(
    line = element_blank(),
    rect = element_rect(fill = "grey95"),
    text = element_text(size = rel(2)),
    plot.background = element_rect(),
    plot.title = element_text(size = 15, color = "grey15", face = "bold"),
    axis.text = element_text(size = 12, color = "gray65"),
    axis.title.x = element_blank(),
    axis.title.y = element_text(size = 12, color = "gray65")
  )
```

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()`).

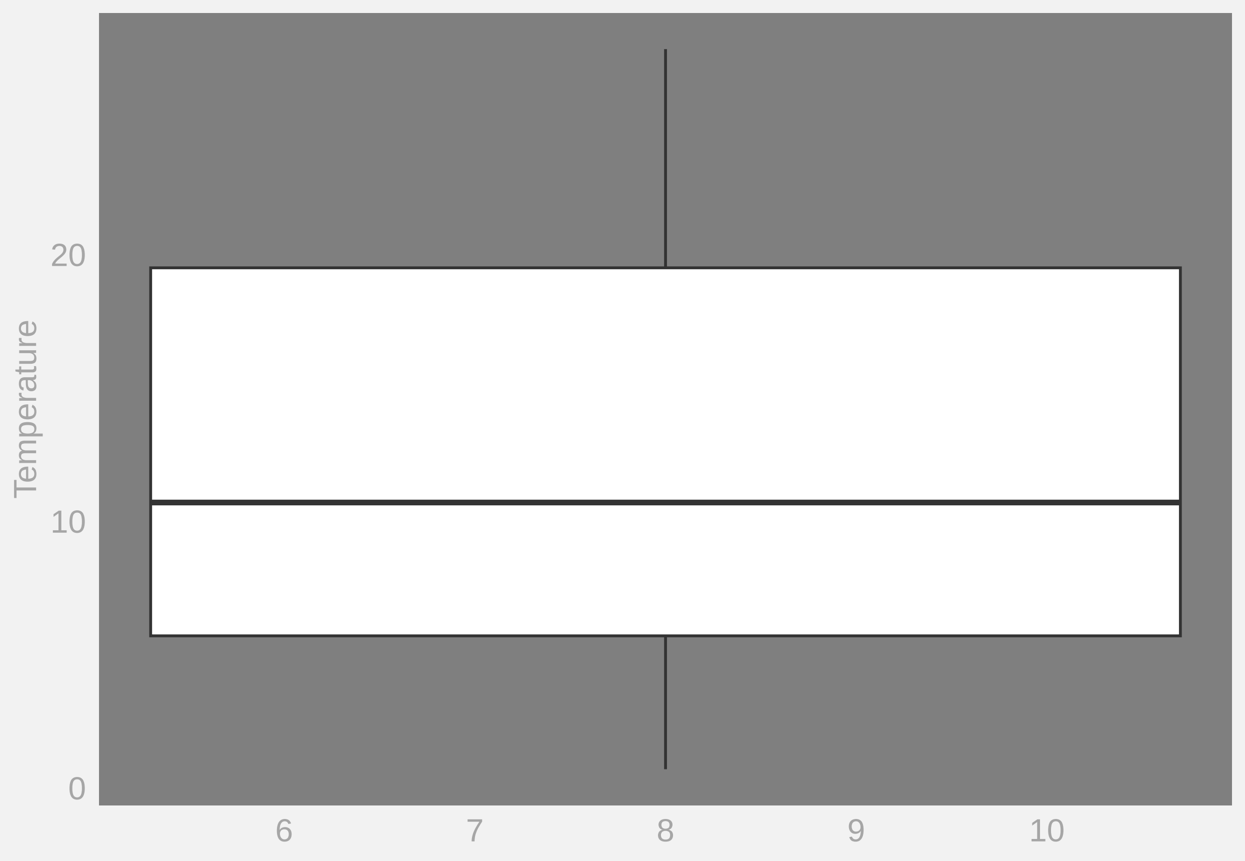
```r
#4
p1 <- ggplot(NTL_LTER_data, aes(x = tp_ug, y = po4, color = lakename)) +
     geom_point() +
  geom_smooth(method = "lm", se = FALSE)+
   labs(title = "Total Phosphorus by Phosphate",
       y = "Phosphate (po4)",
       x = "Total Phosphorus (tp_ug)") +
   ylim(0,50) +
  mytheme

print(p1)
```



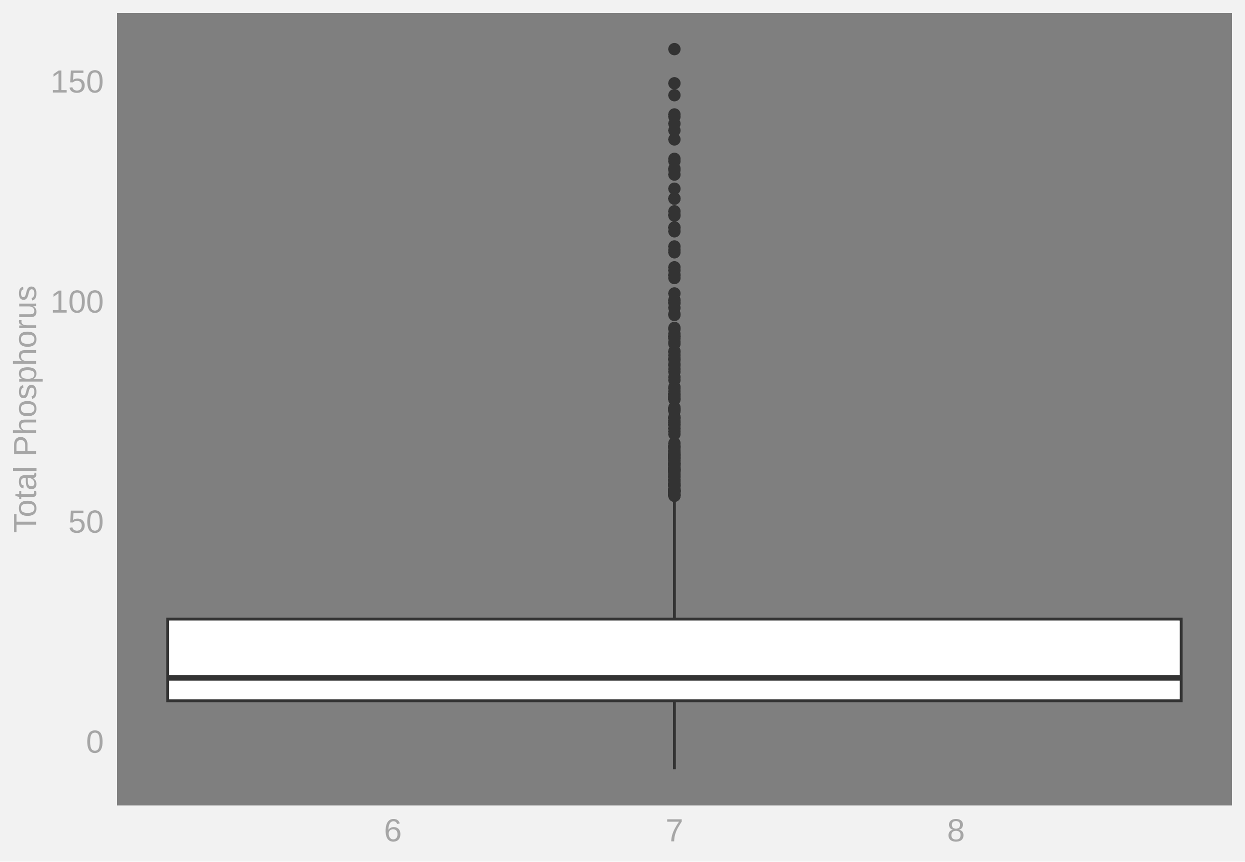
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

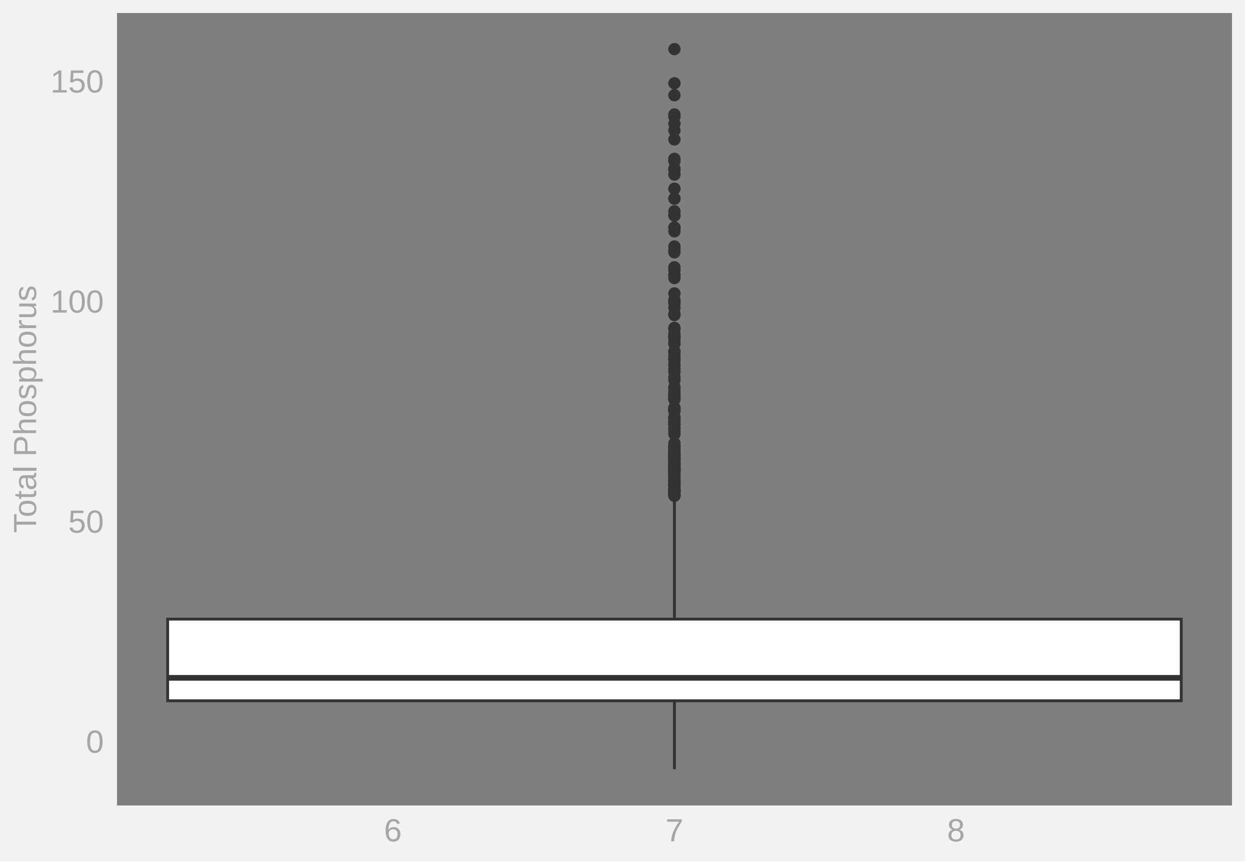
```
#5
boxplot_temp <- ggplot(NTL_LTER_data, aes(x = month(sampledate), y = temperature_C)) +
     geom_boxplot() +
   labs(y = "Temperature") +
  mytheme2
print(boxplot_temp)
```



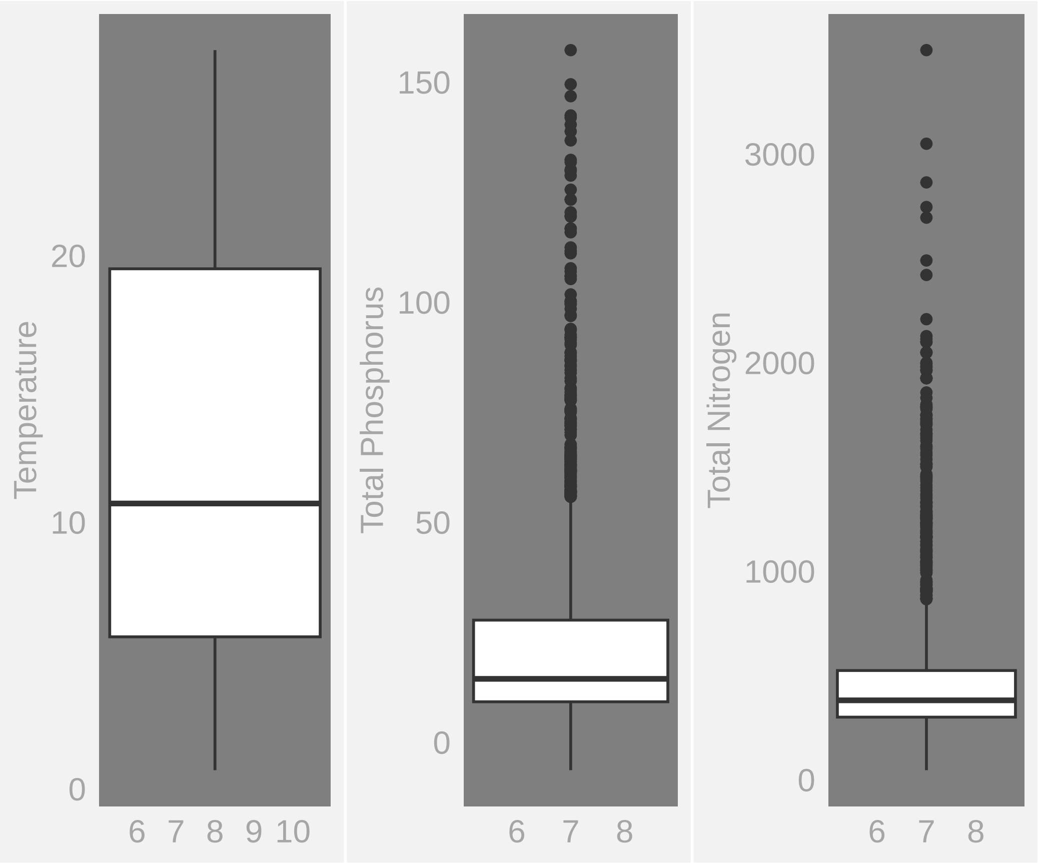
```
boxplot_tp <- ggplot(NTL_LTER_data, aes(x = month(sampledate), y = tp_ug)) +
     geom_boxplot() +
   labs(y = "Total Phosphorus") +
  mytheme2
print(boxplot_tp)
```



```
boxplot_tn <- ggplot(NTL_LTER_data, aes(x = month(sampledate), y = tn_ug)) +
     geom_boxplot() +
   labs(y = "Total Nitrogen") +
  mytheme2
print(boxplot_tp)
```



```
combined_plot1 <- plot_grid(
  plot_grid(boxplot_temp, boxplot_tp, boxplot_tn, nrow = 1),
  ggdraw() +
    draw_label("", fontface = 'bold', size = 14, x = 0.5),
  rel_widths=c(1, 0.2)
  )

print(combined_plot1)
```

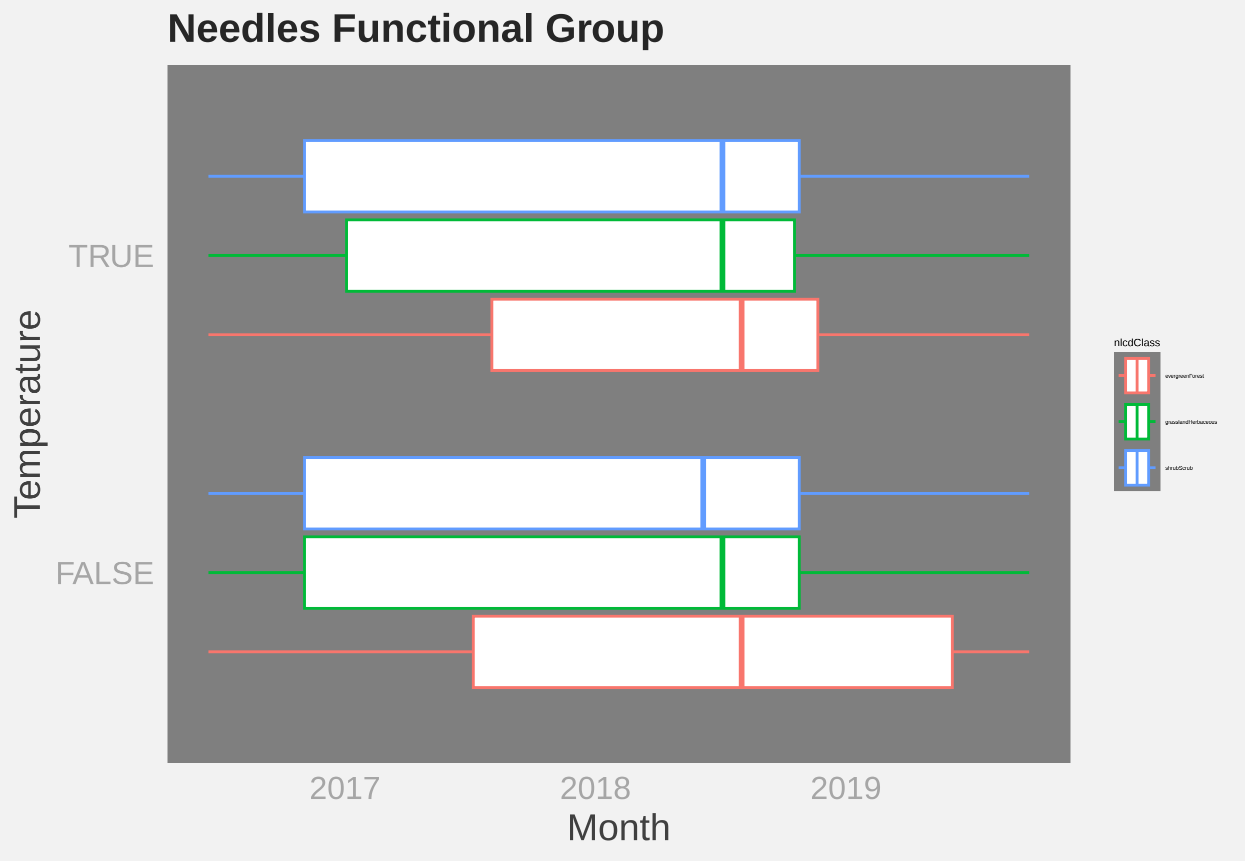


Question: What do you observe about the variables of interest over seasons and between lakes?

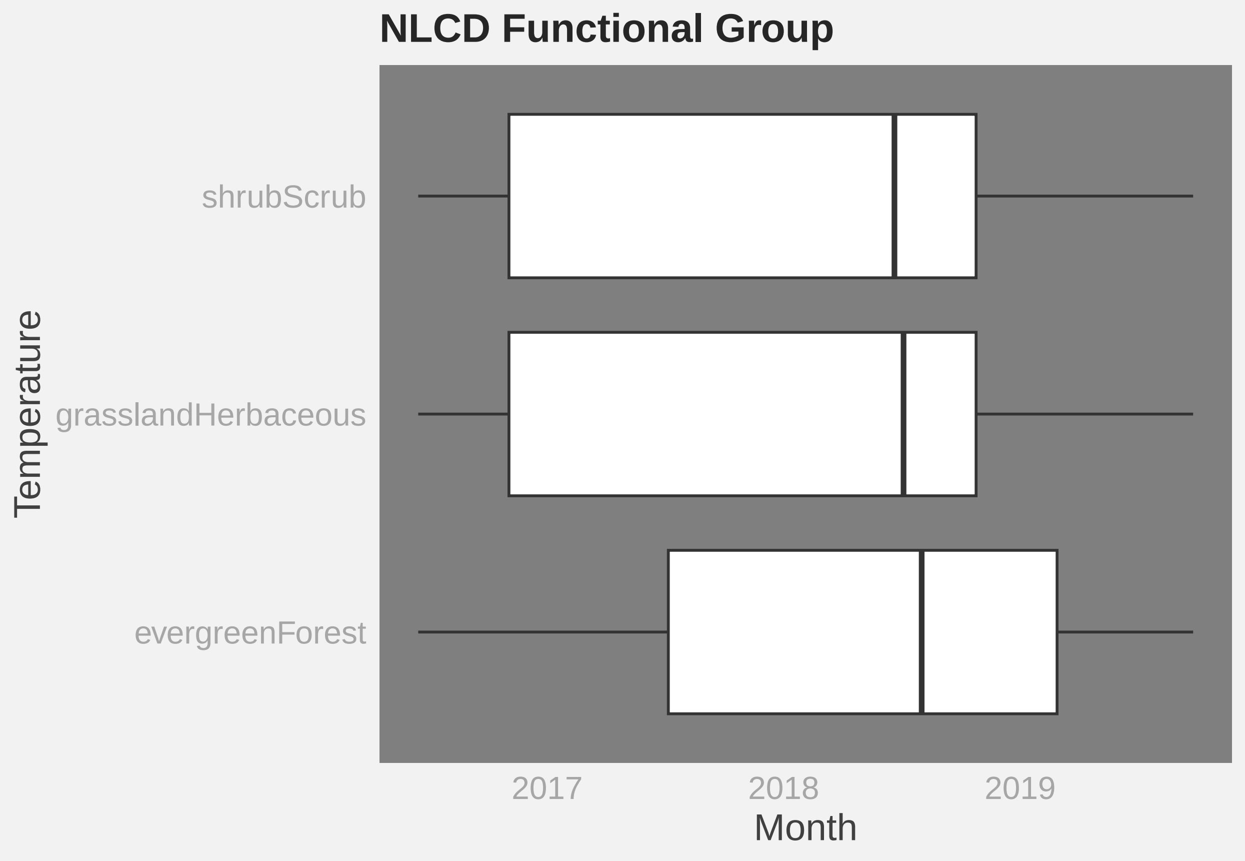
> Answer: There seems to be some seasonal variations in temperature. Both nitrogen and phosphorus seems to have some variability (a lot more variability in nitrogen) over the months. This might be due to levels and water quality.

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
plot_needles <- ggplot(NIWOT_LTER_data, aes(x = collectDate, y = functionalGroup == "Needles", color = r
     geom_boxplot() +

   labs(title = "Needles Functional Group",
       x = "Month",
       y = "Temperature") +
  mytheme
print(plot_needles)
```



```
#7
plot_nlcdClass <- ggplot(NIWOT_LTER_data, aes(x = collectDate, y = nlcdClass)) +
     geom_boxplot() +

   labs(title = "NLCD Functional Group",
       x = "Month",
       y = "Temperature") +
  mytheme
print(plot_nlcdClass)
```



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

> Answer: Plot 6 uses colors to further disaggregate the NLCD classes and is easier to understand. It may not be the optimal choice when there are more classes per functional group.