Assignment 5: Data Visualization

Alex Lopez

Fall 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

#load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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/EDE_Fall2024
```

```r
library(cowplot)
```

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

```r
#check working directory
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
#read in data files
NutrientsPeterPaul <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)

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

#2

#convert dates from factors to date objects
NutrientsPeterPaul$sampledate <-
  ymd(as.character(NutrientsPeterPaul$sampledate))

LitterNiwotRidge$collectDate <- ymd(as.character(LitterNiwotRidge$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

```r
#3

#set my theme
```

```
mytheme <- theme_classic(base_size = 14) +
  theme(plot.background = element_rect(fill = "black", color = "black",
                                       size = 1),
        plot.title = element_text(face = "bold", size = 20, color = "white",
                                  hjust = 0.5),
        axis.text = element_text(color = "white"),
        axis.title = element_text(color = "white"),
        axis.ticks = element_line(color = "white", size = 1),
        panel.grid.major = element_line(color = "gray"),
        panel.grid.minor = element_line(color = "gray"),
        panel.background = element_rect(fill = "black", color = "black"),
        legend.background = element_rect(fill = "black", color = "white"),
        legend.title = element_text(face = "bold", size = 12, color = "white"),
        legend.text = element_text(size = 10, color = "white"),
        legend.position = "right")
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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.

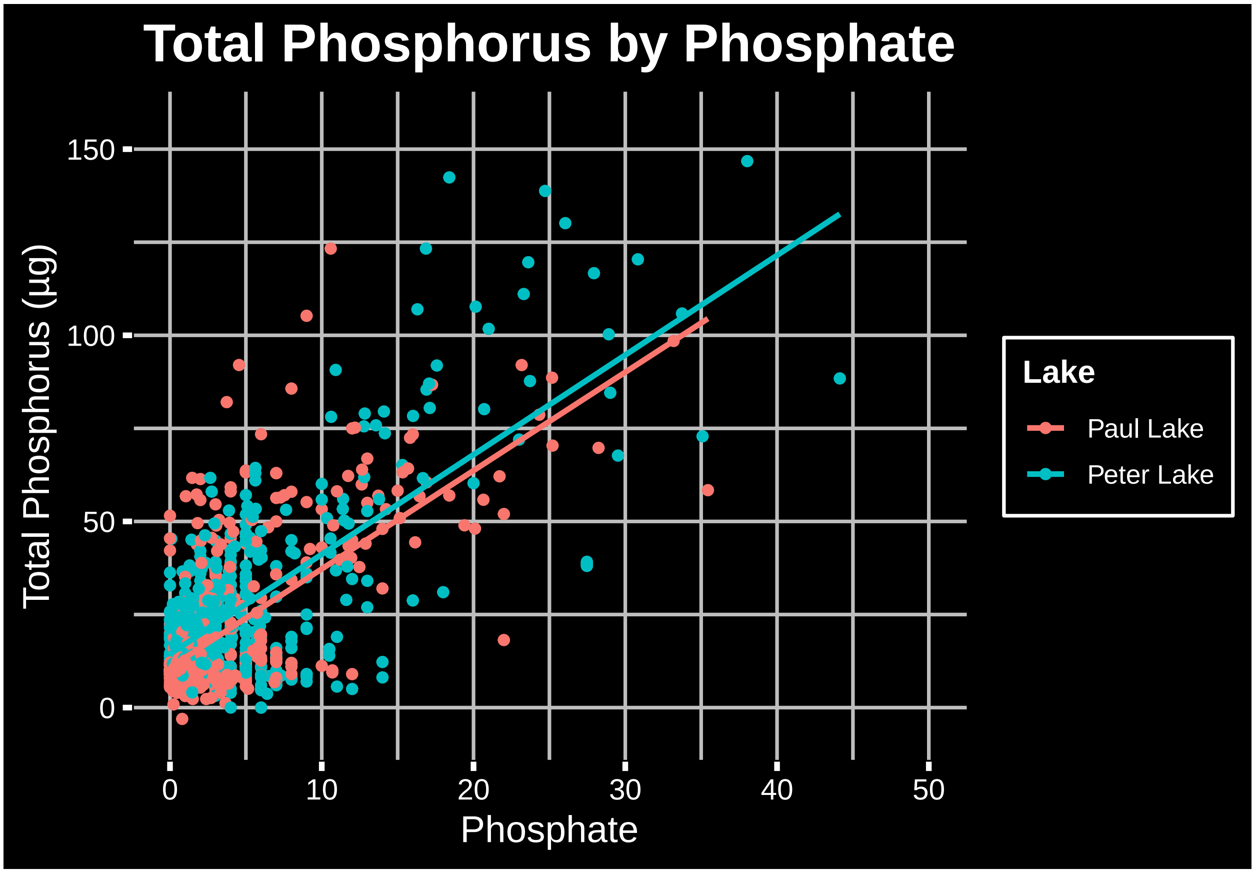
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

NutrientsPeterPaul %>%
  ggplot(aes(x = po4,
             y = tp_ug,
             color = lakename)) +
  geom_point() +
  geom_smooth(method = lm, se = F) +
  xlim(0,50) +
  labs(title = 'Total Phosphorus by Phosphate', x = 'Phosphate',
       y = 'Total Phosphorus (µg)', color = 'Lake') +
  mytheme
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 21947 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (`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.

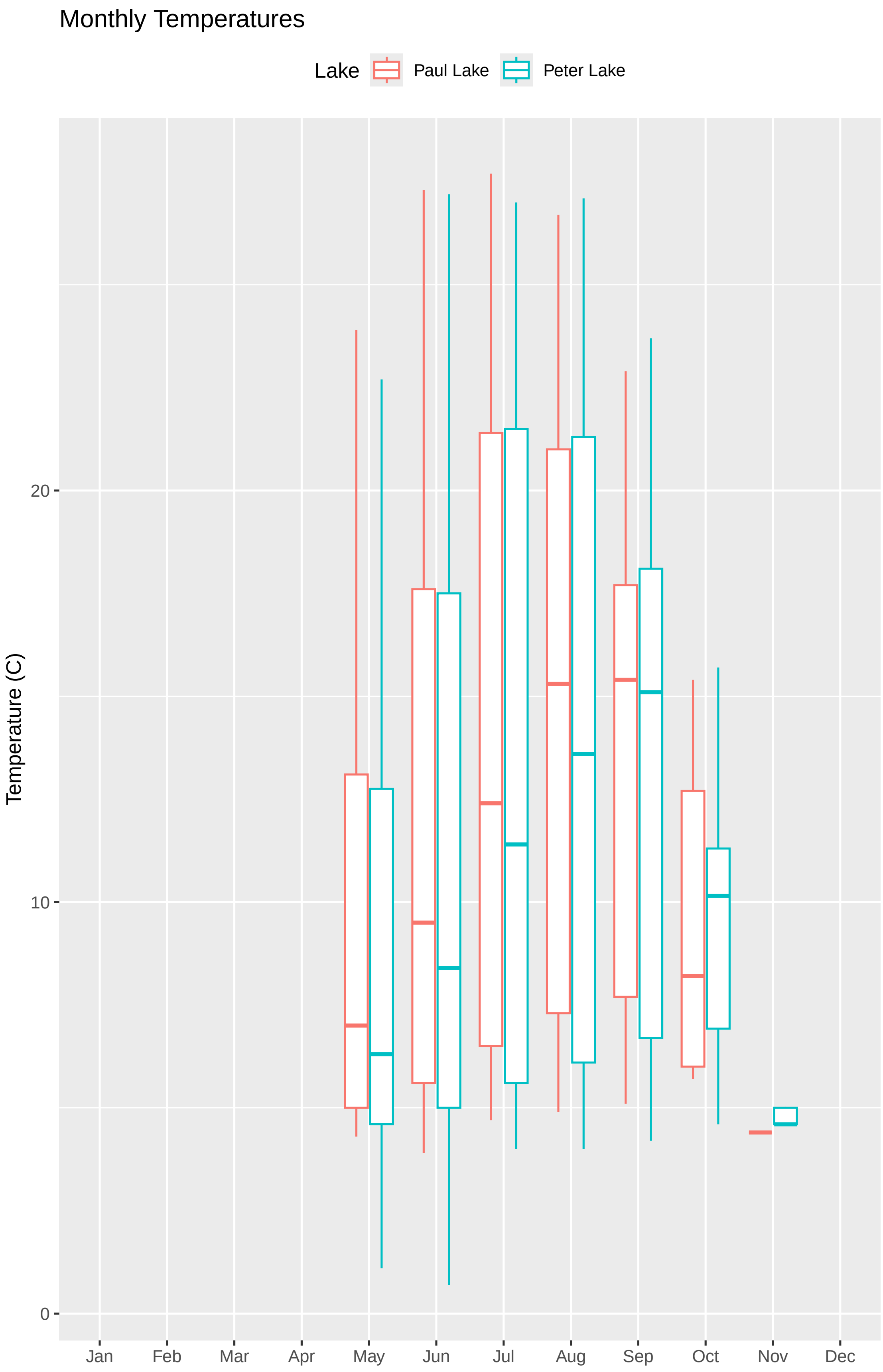
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a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

```
#5

#create temp boxplot
TemperatureBoxplot <-
  ggplot(NutrientsPeterPaul, aes(x = factor(month,
                                            levels = 1:12,
                                            labels = month.abb),
                                 y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = 'Month', drop = FALSE) +
```

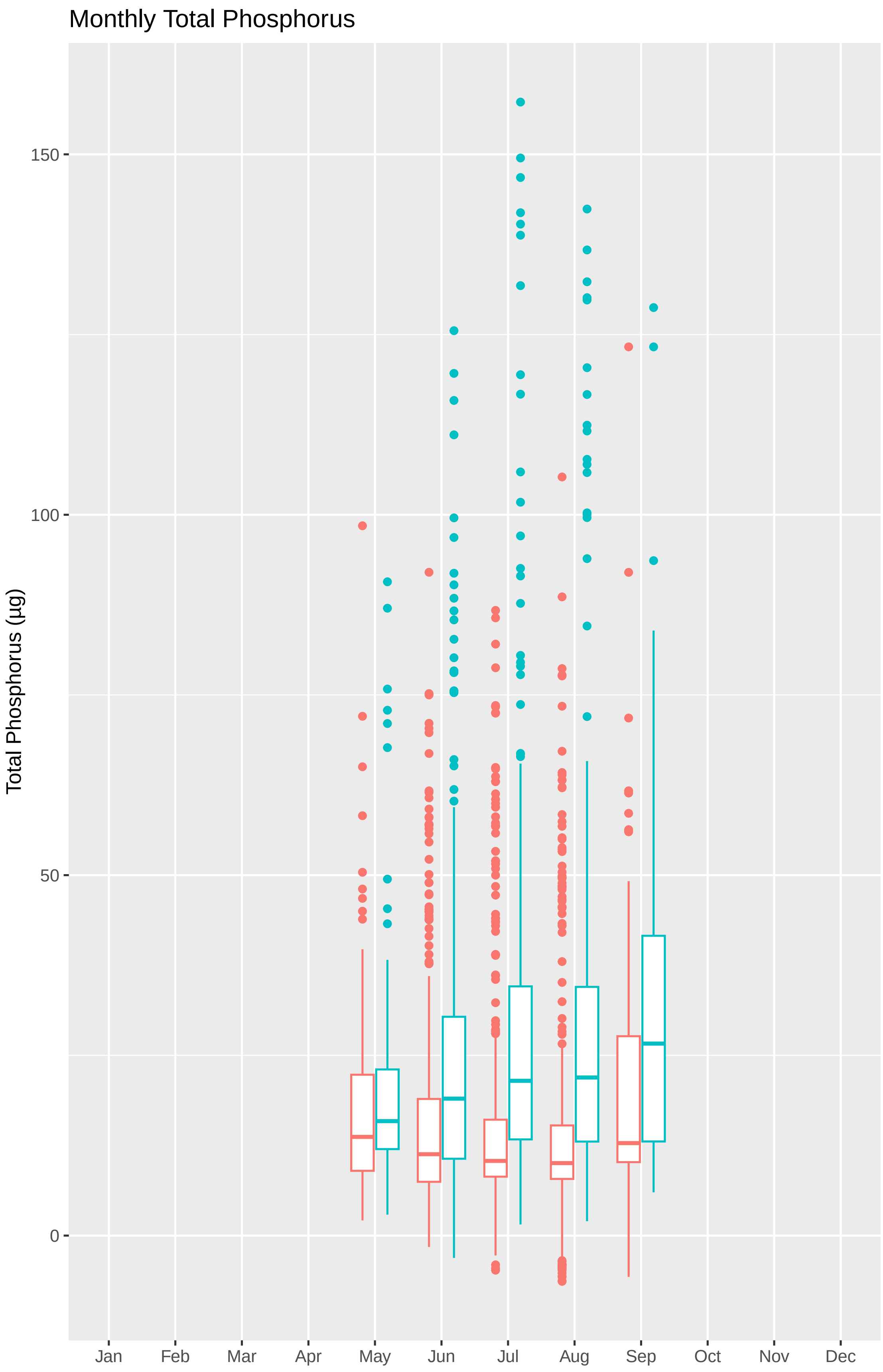
```r
  labs(y = 'Temperature (C)', color = 'Lake', title = 'Monthly Temperatures') +
  theme(legend.position = 'top', axis.title.x = element_blank())
print(TemperatureBoxplot)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


Monthly Temperatures
Lake
Paul Lake
Peter Lake
Temperature (C)
20
10
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

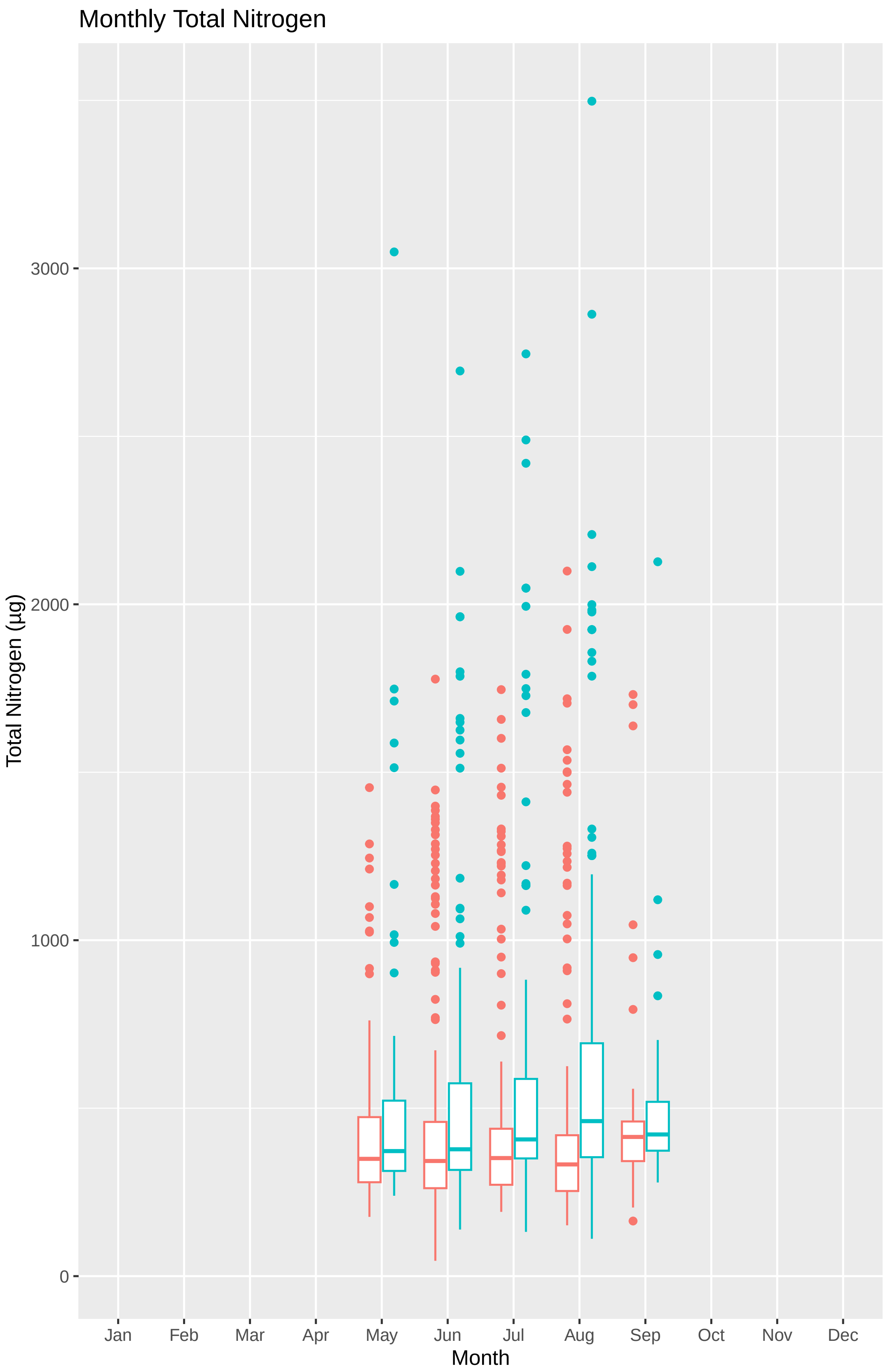
```r
#create TP boxplot
TP_Boxplot <-
  ggplot(NutrientsPeterPaul, aes(x = factor(month,
                                            levels = 1:12,
                                            labels = month.abb),
                                 y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = 'Month', drop = FALSE) +
  labs(y = "Total Phosphorus (µg)", color = 'Lake',
       title = 'Monthly Total Phosphorus') +
  theme(legend.position = 'none', axis.title.x = element_blank())
print(TP_Boxplot)
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


Monthly Total Phosphorus
Total Phosphorus (µg)
150
100
50
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

```r
#create TN boxplot
TN_Boxplot <-
  ggplot(NutrientsPeterPaul, aes(x = factor(month,
                                            levels = 1:12,
                                            labels = month.abb),
                                 y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = 'Month', drop = FALSE) +
  labs(y = 'Total Nitrogen (µg)', color = 'Lake',
       title = 'Monthly Total Nitrogen') +
  theme(legend.position = 'none')
print(TN_Boxplot)
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```


Monthly Total Nitrogen
Total Nitrogen (µg)
3000
2000
1000
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

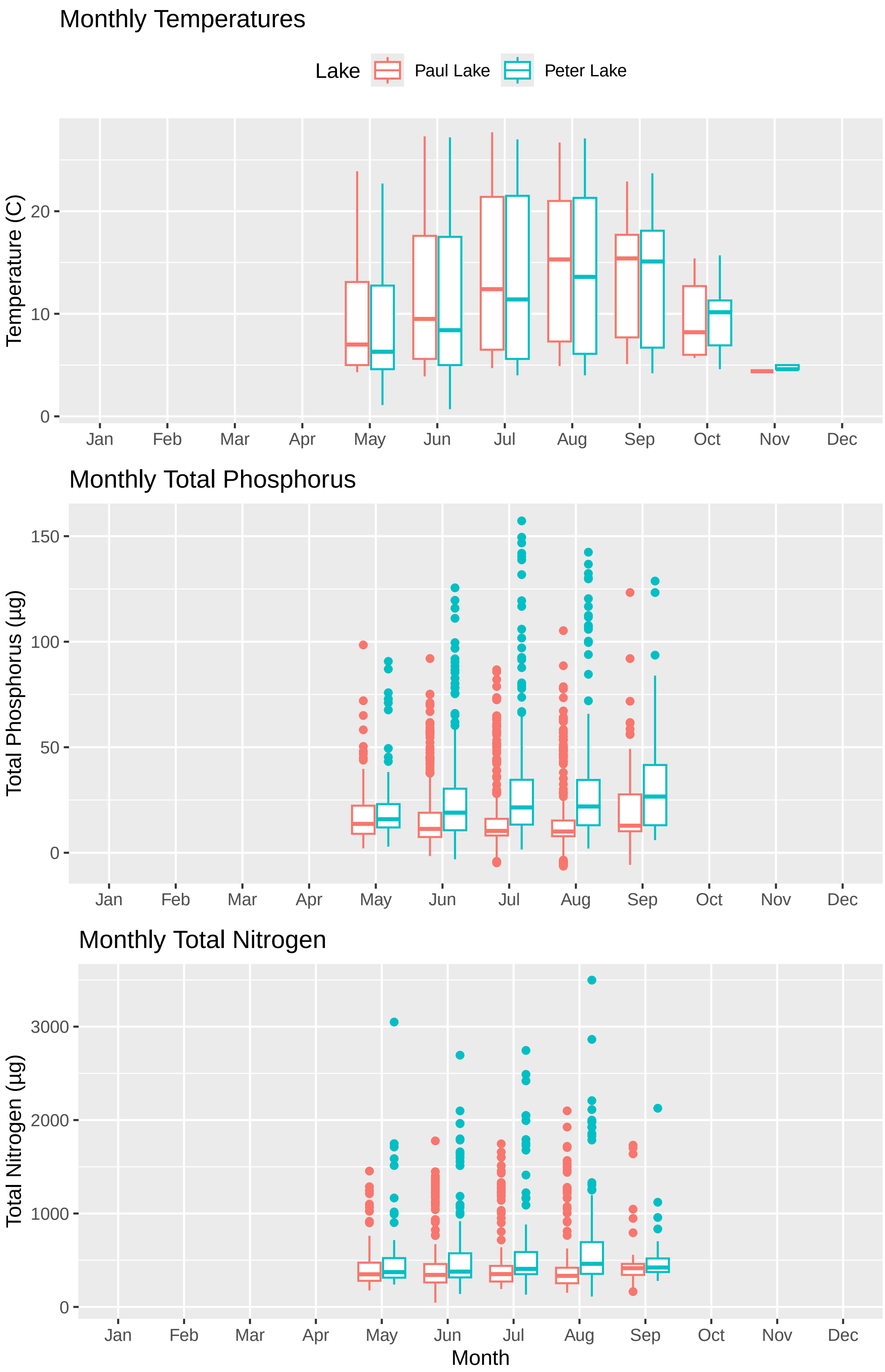
```r
#combine three boxplots
combined_plot <- plot_grid(TemperatureBoxplot, TP_Boxplot,
                           TN_Boxplot, nrow = 3)
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 21583 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
print(combined_plot)
```


Monthly Temperatures
Lake
Paul Lake
Peter Lake
Temperature (C)
0
10
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Monthly Total Phosphorus
Total Phosphorus (µg)
0
50
100
150
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Monthly Total Nitrogen
Total Nitrogen (µg)
0
1000
2000
3000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

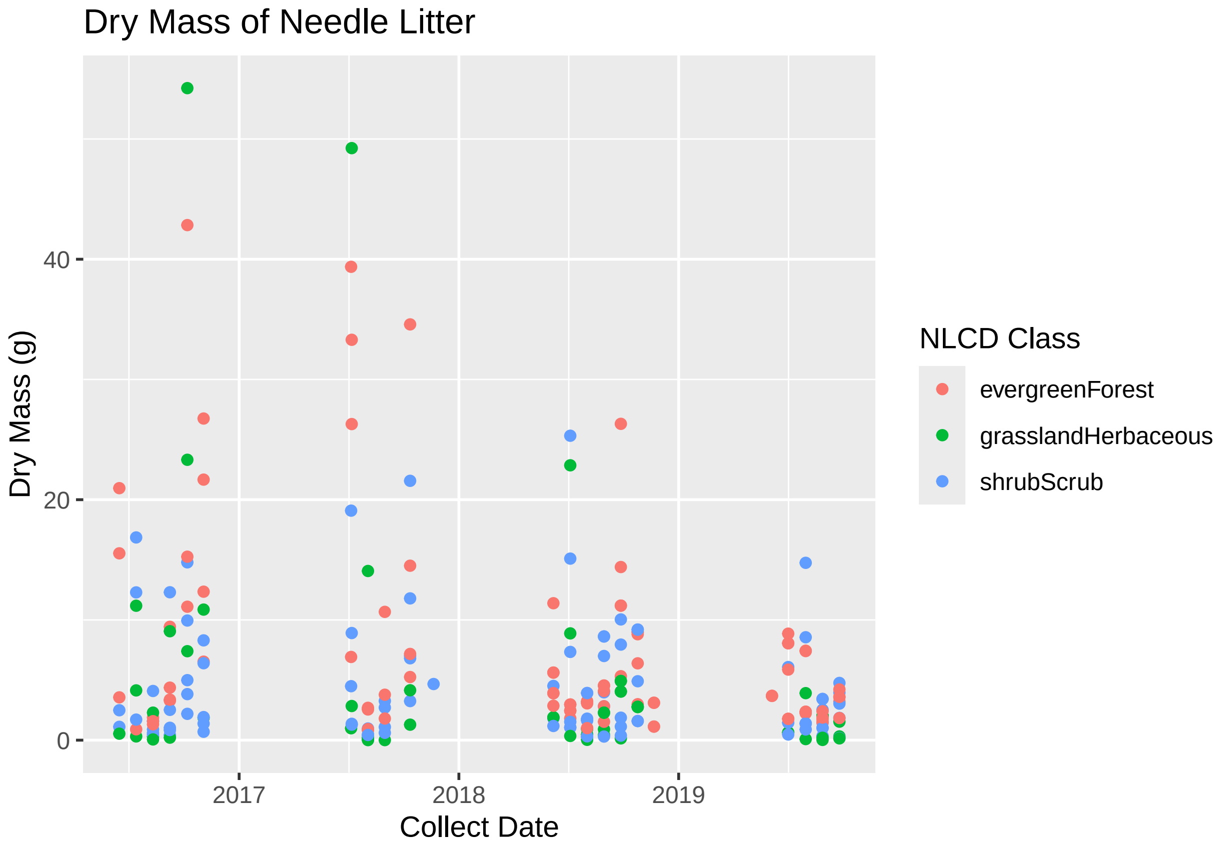
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: For the months in which the lakes are compared, Peter Lake appears to consistently have higher total amounts in nitrogen and phosphorus than Paul Lake, as well as a larger range in total amounts of nitrogen and phosphorus. Regarding temperature, there doesn't seem to be much of a difference between both lakes during the months compared; their ranges look similar through the months.

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.

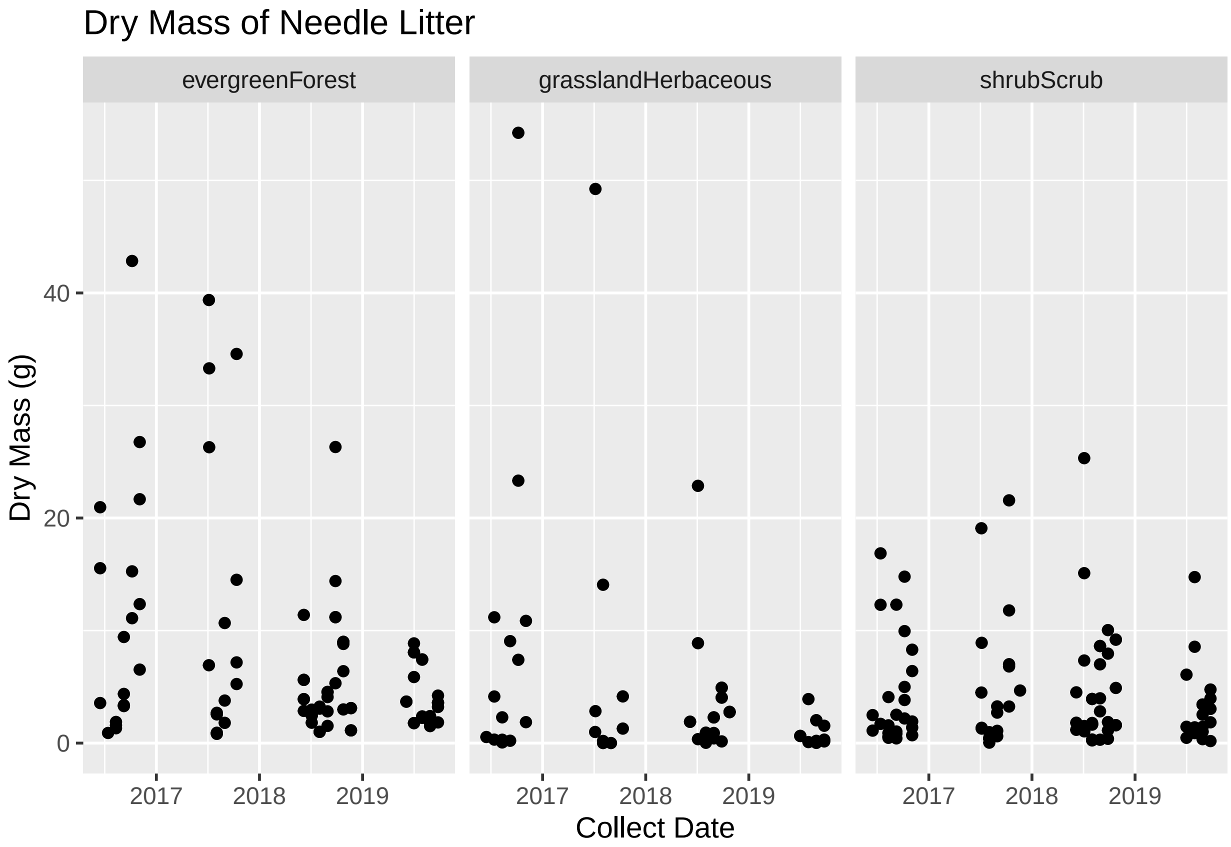
```
#6

LitterNiwotRidge %>%
  filter(functionalGroup == "Needles") %>%
  ggplot() +
  aes(x = collectDate, y = dryMass, color = nlcdClass) +
  geom_point() +
  labs(x = 'Collect Date', y = 'Dry Mass (g)',
       title = 'Dry Mass of Needle Litter', color = 'NLCD Class')
```



```
#7

LitterNiwotRidge %>%
  filter(functionalGroup == "Needles") %>%
  ggplot() +
  aes(x = collectDate, y = dryMass) +
  geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  labs(x = 'Collect Date', y = 'Dry Mass (g)',
       title = 'Dry Mass of Needle Litter')
```



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

Answer: Plot 6 is arguably more effective because you have all the data needed for the analysis in one plot. Its main advantage is that you can easily compare between the different NLCD classes by looking at the distribution of the points and their respective color. However, plot 7 could prove to be more effective if you're looking to compare not between all the classes but only between some of the classes. You would only focus on the plots relevant to your analysis without any extraneous data. Additionally, by focusing on the separated plots individually, you could also come up with useful analysis regarding trends of the individual plot's data.