Assignment 5: Data Visualization

Student Name

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.

```
#1
#install.packages(tidyverse)
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 1.9.4     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 errors
```

```r
#install.packages("lubridate")
library(lubridate)
#install.packages("here")
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2025
```

```r
#install.packages("cowplot")
library(cowplot)
```

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

```r
#verifying home directory
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```r
here::here()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```r
dir.exists("/home/guest/EDA_Spring2025/Data/Processed_KEY")
```

```
## [1] TRUE
```

```r
#Searching Files
list.files("/home/guest/EDA_Spring2025/Data/Processed_KEY")
```

```
##  [1] "EPAair_O3_NC2018_processed.csv"
##  [2] "EPAair_O3_NC2019_processed.csv"
##  [3] "EPAair_O3_PM25_NC1819_Processed.csv"
##  [4] "EPAair_PM_NC2018_processed.csv"
##  [5] "EPAair_PM_NC2019_processed.csv"
##  [6] "NEON_NIWO_Litter_mass_trap_Processed.csv"
##  [7] "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"
##  [8] "NTL-LTER_Lake_ChemistryPhysics_PeterPaul_Processed.csv"
##  [9] "NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv"
## [10] "NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv"
## [11] "NTL-LTER_Lake_Summaries_PeterPaul_Processed.csv"
## [12] "USGS_Site02085000_Flow_Processed.csv"
```

```r
#Opening the Data
LTER_PeterPaul_Processed <- read.csv("/home/guest/EDA_Spring2025/Data/Processed_KEY/NTL-LTER_Lake_Chemis

NEON_NIWOmass_trap_Processed <- read.csv("/home/guest/EDA_Spring2025/Data/Processed_KEY/NEON_NIWO_Litte

#2

lapply(list(LTER_PeterPaul_Processed,NEON_NIWOmass_trap_Processed), str)
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : Factor w/ 1103 levels "1984-05-27","1984-05-28",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 9 8 9 11 7 7 4 4 4 4 ...
##  $ trapID          : Factor w/ 15 levels "NIWO_040_139",..: 11 10 11 13 9 9 5 5 5 5 ...
##  $ collectDate     : Factor w/ 24 levels "2016-06-16","2016-07-14",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ functionalGroup : Factor w/ 8 levels "Flowers","Leaves",..: 6 5 8 6 4 2 2 6 7 8 ...
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 1 1 1 ...
##  $ subplotID       : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : Factor w/ 3 levels "evergreenForest",..: 3 1 3 1 3 3 2 2 2 2 ...
##  $ plotType        : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum   : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...

## [[1]]
## NULL
##
## [[2]]
## NULL
```

```r
#Sample date is misunderstood as a factor it is necessary to change the format to date

LTER_PeterPaul_Processed$sampledate <- as.Date(LTER_PeterPaul_Processed$sampledate,
                                               format = "%Y-%m-%d")

NEON_NIWOmass_trap_Processed$collectDate <- as.Date(NEON_NIWOmass_trap_Processed$collectDate,
```

```
                                                    format = "%Y-%m-%d")
lapply(list(LTER_PeterPaul_Processed,NEON_NIWOmass_trap_Processed), str)
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : Date, format: "1984-05-27" "1984-05-27" ...
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 9 8 9 11 7 7 4 4 4 4 ...
##  $ trapID          : Factor w/ 15 levels "NIWO_040_139",..: 11 10 11 13 9 9 5 5 5 5 ...
##  $ collectDate     : Date, format: "2016-06-16" "2016-06-16" ...
##  $ functionalGroup : Factor w/ 8 levels "Flowers","Leaves",..: 6 5 8 6 4 2 2 6 7 8 ...
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 1 1 1 ...
##  $ subplotID       : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : Factor w/ 3 levels "evergreenForest",..: 3 1 3 1 3 3 2 2 2 2 ...
##  $ plotType        : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum   : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
```

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

library(ggplot2)
# I define my custom theme with updated colors
Angelica_custom_theme <- function() {
theme_minimal() +
theme(
# Customize the plot background
plot.background = element_rect(fill = "#f8f9fa", color = NA),,
# Customize the plot title
plot.title = element_text(color = "darkblue", size = 16, face = "bold"),
# Customize the axis labels
 axis.title = element_text(color = "#005f73", size = 14, face = "italic"),,
# Customize axis ticks/gridlines
axis.text = element_text(color = "#4a4a4a", size = 12),
 axis.line = element_line(color = "#468faf"),
 axis.ticks = element_line(color = "#e63946"),
# Customize the legend
legend.title = element_text(color = "darkblue", size = 12, face = "bold"),
legend.text = element_text(color = "darkred", size = 10)
)}
#I customized the look of all of the previous to see different colors
# I set my custom theme as the default theme
theme_set(Angelica_custom_theme())
```

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.

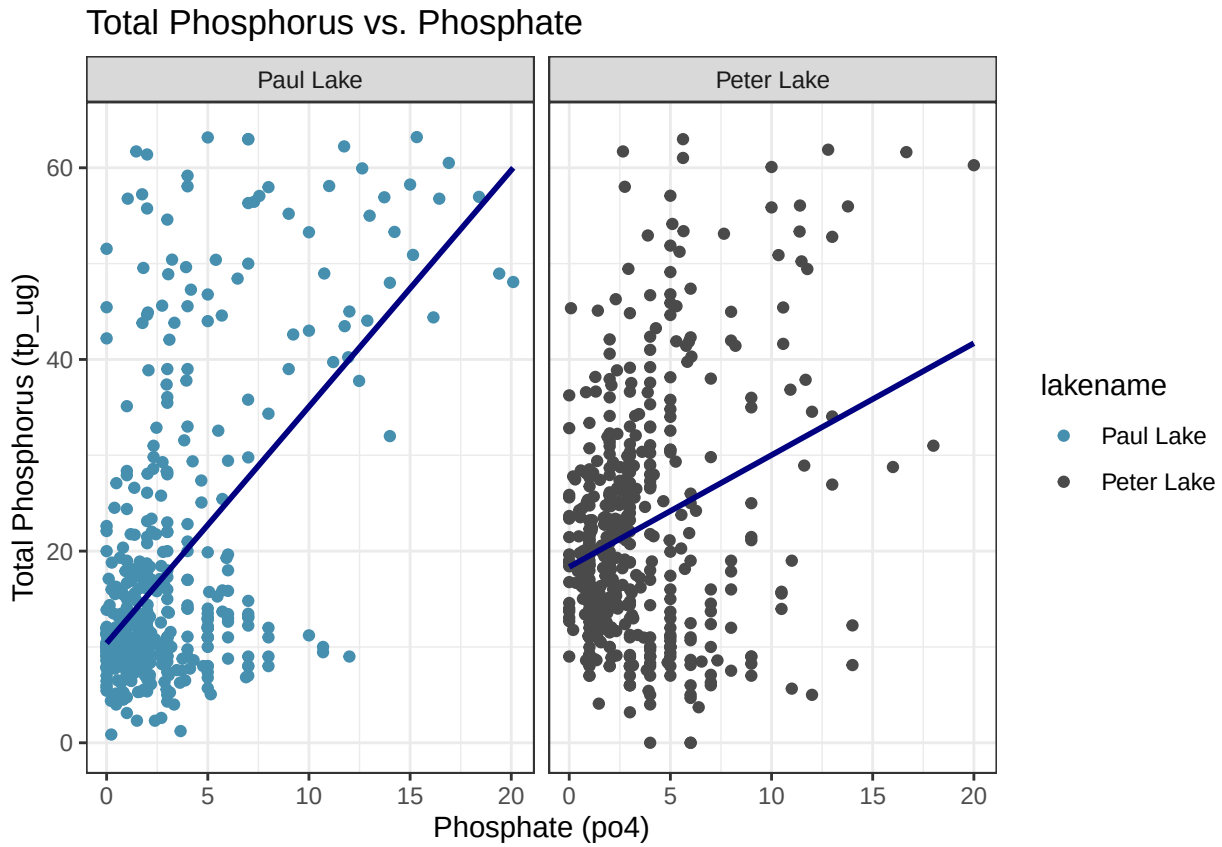
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4

# To see the dplyr library
library(dplyr)
#In this code I swil separate plots for Peter and Paul lakes

TP_vrs_PO4_plot2 <- ggplot(LTER_PeterPaul_Processed, aes(x = po4, y = tp_ug)) +
  # Change point color and Define color within aes() only
  geom_point(aes(color = lakename)) +

  # Regression line with deep teal color
  geom_smooth(method = "lm", formula = y ~ x, color = "navy", se = FALSE) +

  # Labels
  labs(
    title = "Total Phosphorus vs. Phosphate",
    x = "Phosphate (po4)",
    y = "Total Phosphorus (tp_ug)"
```

```
  ) +

  # Limit axes to 95th percentile to avoid extreme outliers
  scale_y_continuous(limits = c(0, quantile(LTER_PeterPaul_Processed$tp_ug, 0.95, na.rm = TRUE))) +
  scale_x_continuous(limits = c(0, quantile(LTER_PeterPaul_Processed$po4, 0.95, na.rm = TRUE))) +

  # Define custom colors outside aes()
  scale_color_manual(values = c("Paul Lake" = "#468faf", "Peter Lake" = "#4a4a4a")) +

  # Black and white theme
  theme_bw() +

  # Separate graphs for each lake
  facet_wrap(~ lakename, ncol = 2)

print(TP_vrs_PO4_plot2)
```

```
## Warning: Removed 22012 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 22012 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



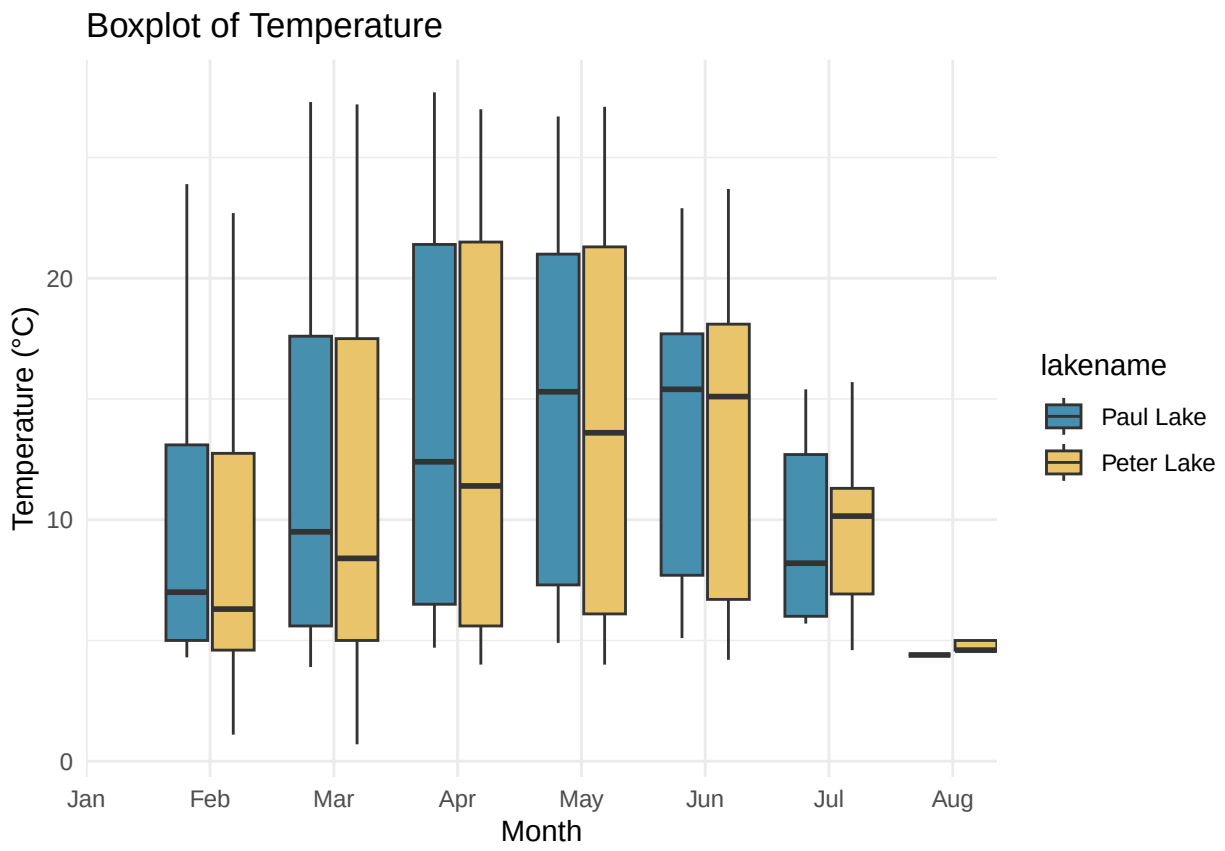
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as

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`.

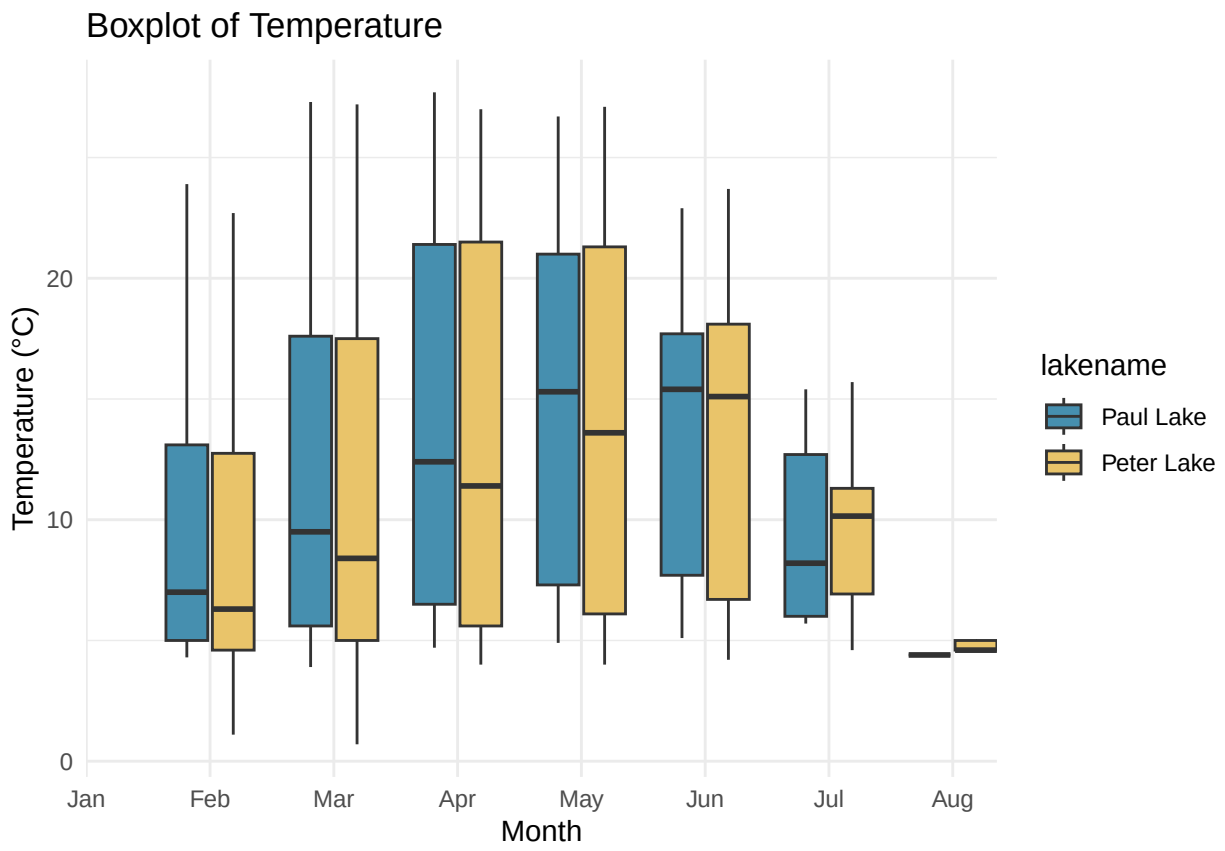
```
#5

# Loading  required libraries
library(ggplot2)
library(cowplot)

# Changing months
all_months <- data.frame(month = unique(LTER_PeterPaul_Processed$month))

# Now I define a vector of abbreviated month labels
month_labels <-
c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
# Merge the original data with the data frame containing all months
LTER_PeterPaul_Processed <- merge(all_months, LTER_PeterPaul_Processed, by = "month", all.x = TRUE)
# Convert the merged month column to factors with abbreviated month labels
LTER_PeterPaul_Processed$month <- factor(LTER_PeterPaul_Processed$month, levels = 1:12, labels = month_l
# Next, I modify my boxplot code

boxplot_temp <- ggplot(LTER_PeterPaul_Processed, aes
(x = factor(month, levels = unique(month)),
y = temperature_C, fill = lakename)) +
geom_boxplot() +
labs(title = "Boxplot of Temperature", y = "Temperature (°C)", x= "Month") +
theme_minimal() +
scale_fill_manual(values = c("Paul Lake" = "#468faf", "Peter Lake" = "#e9c46a"))+
scale_x_discrete(expand = c(0, 0), labels = month_labels)

print(boxplot_temp)
```

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



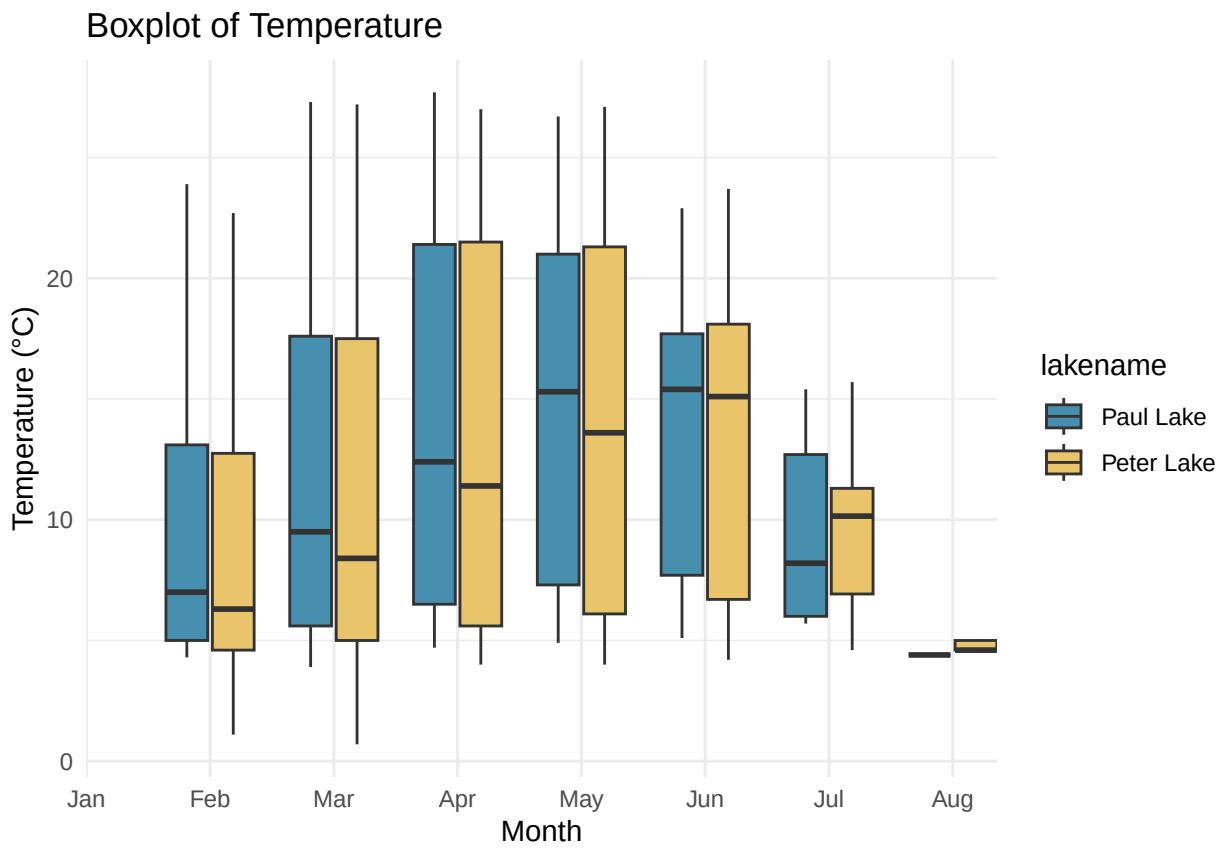

```
boxplot_tp <- ggplot(LTER_PeterPaul_Processed,
aes(x = factor(month, levels = unique(month)),
y = tp_ug, fill = lakename)) +
geom_boxplot() +
labs(title = "Boxplot of Total Phosphorus (TP)", y = "Total P", x= "Month") +
theme_minimal() +
scale_fill_manual(values = c("Paul Lake" = "#468faf", "Peter Lake" = "#e9c46a"))+
scale_x_discrete(expand = c(0, 0), labels = month_labels)

print(boxplot_temp)
```

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




```
boxplot_tn <- ggplot(LTER_PeterPaul_Processed,
aes(x = factor(month, levels = unique(month)),
y = tn_ug, fill = lakename)) +
geom_boxplot() +
labs(title = "Boxplot of Total Nitrogen (TN)", y = "Total N", x= "Month") +
theme_minimal() +
scale_fill_manual(values = c("Paul Lake" = "#468faf", "Peter Lake" = "#e9c46a"))+
scale_x_discrete(expand = c(0, 0), labels = month_labels)

print(boxplot_temp)
```

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



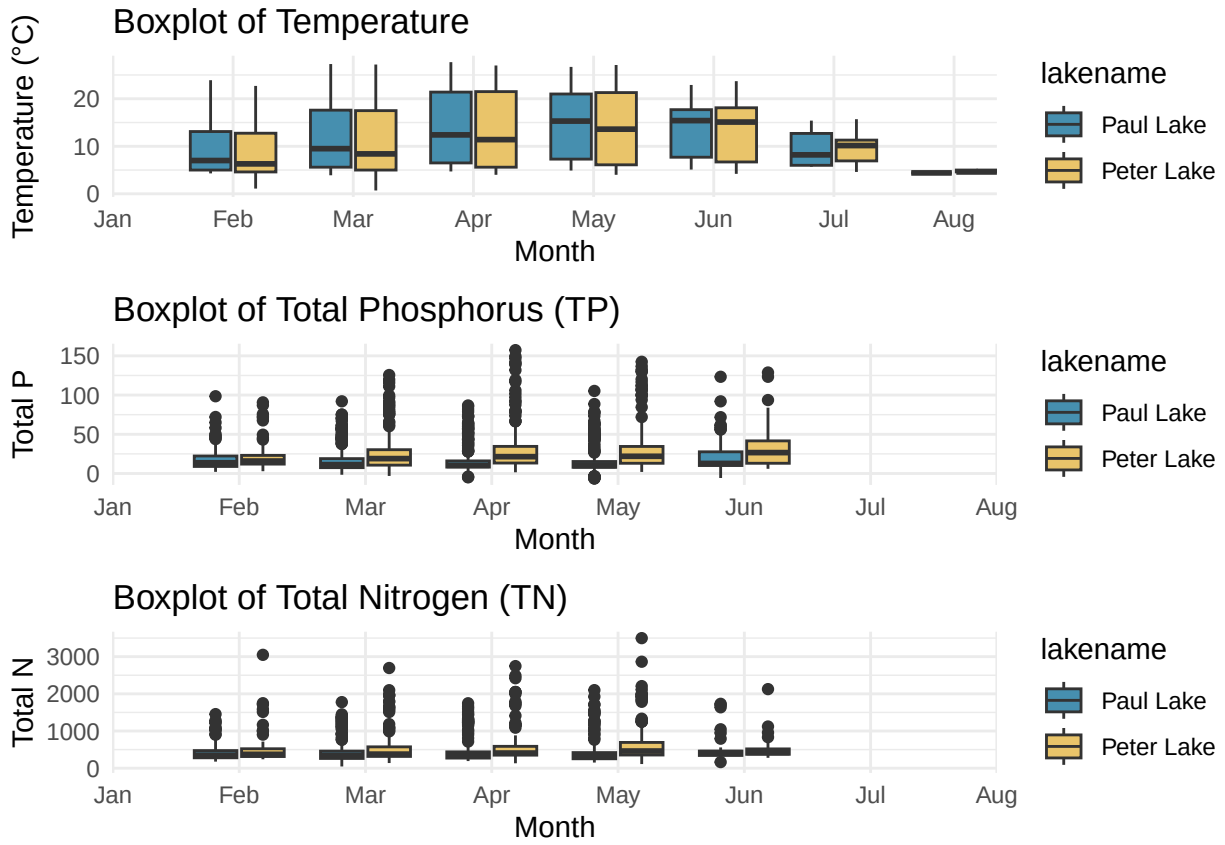

```
# Here I combine the three boxplots into a single cowplot with one legend and aligned axes
combined_plot <- plot_grid(boxplot_temp, boxplot_tp,
boxplot_tn, ncol = 1, align = "v",
rel_heights = c(1, 1, 1))
```

```
## 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()`).
```

```
print(combined_plot)
```



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

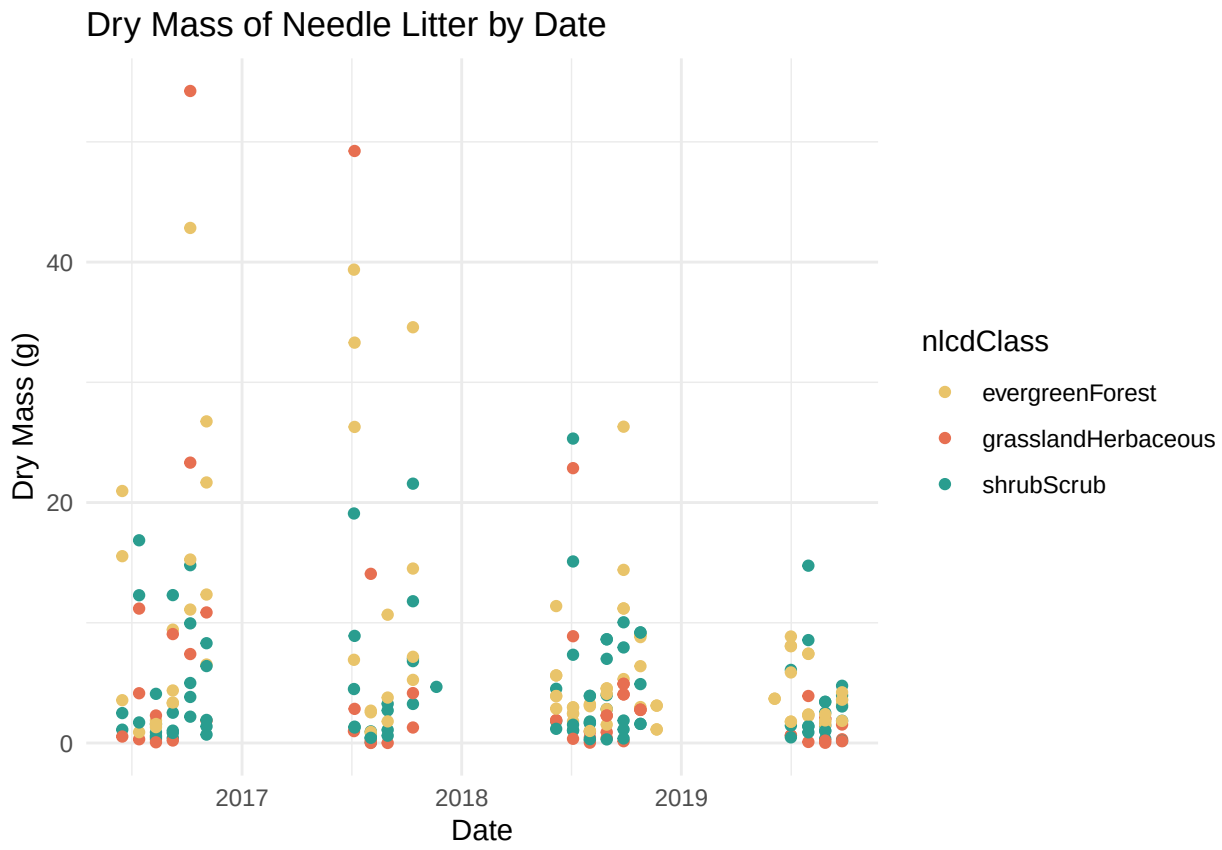
Answer: March, April, and May experience the highest temperatures, which also affect the total phosphorus and nitrogen levels in both lakes.

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)

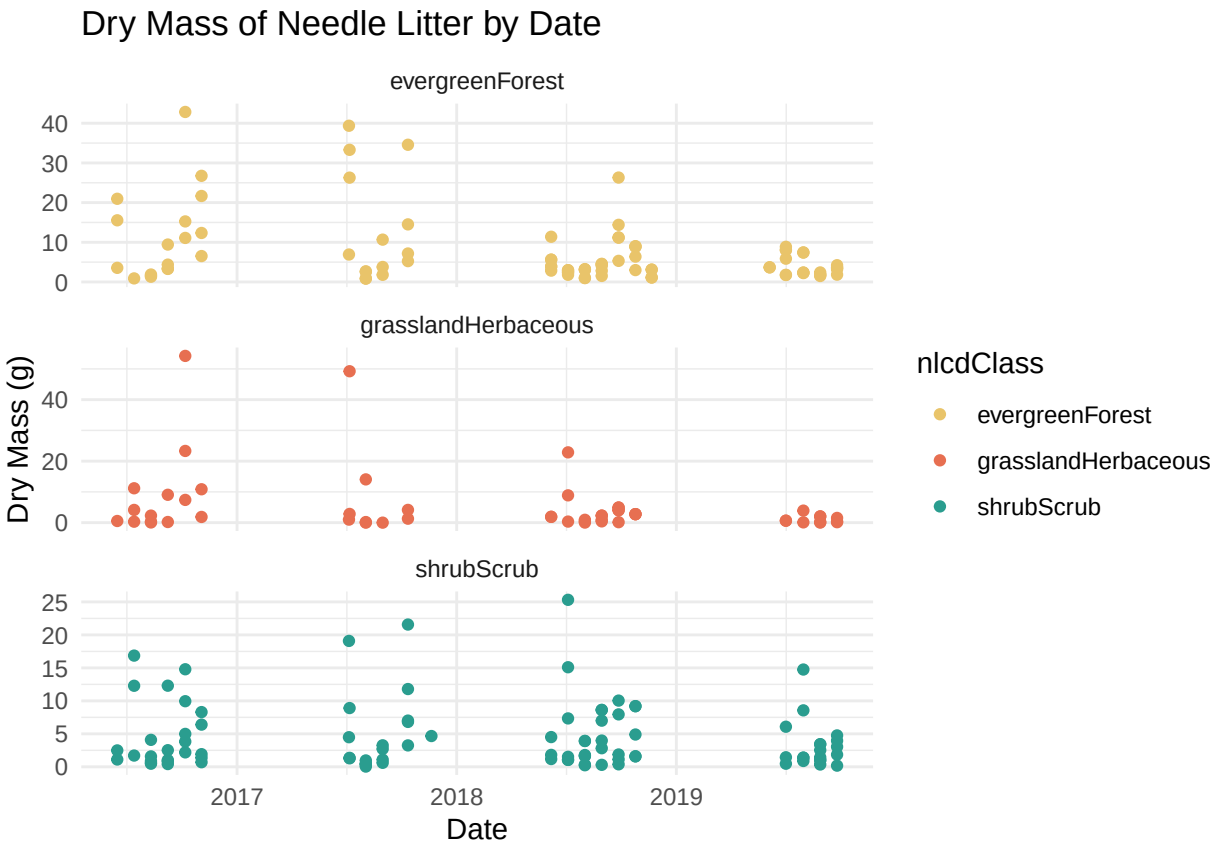
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

# To filter the dataset and include only the "Needles" functional group

needles_display <- NEON_NIWOmass_trap_Processed %>%
filter(functionalGroup == "Needles")
# Now I create the plot
ggplot(needles_display, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
geom_point() +
labs(
title = "Dry Mass of Needle Litter by Date",
x = "Date",
y = "Dry Mass (g)"
) +
```

```
theme_minimal()+
scale_color_manual(values = c("evergreenForest" = "#e9c46a", "shrubScrub" = "#2a9d8f","grasslandHerbace
```




```
#7

needles_display2 <- NEON_NIWOmass_trap_Processed %>%
filter(functionalGroup == "Needles")
# Now I create the plot with facets
ggplot(needles_display2, aes(x = collectDate, y = dryMass)) +
geom_point(aes(color = nlcdClass)) +
labs(
title = "Dry Mass of Needle Litter by Date",
x = "Date",
y = "Dry Mass (g)"
) +
facet_wrap(~ nlcdClass, scales = "free_y", ncol = 1) +
# Separate by nlcdClass into facets
theme_minimal()+
  scale_color_manual(values = c("evergreenForest" = "#e9c46a", "shrubScrub" = "#2a9d8f","grasslandHerba
```



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

Answer:It depends on the project's goal because the separate NLCD classes help understand each of the variables independently. However, the only "Needles" display is to compare and connect how they behave in the three nIcd Classes.