Assignment 5: Data Visualization

Brian Mulu Mutua

Fall 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. 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 Initial setup
#Checking working directory
getwd()
```

```
## [1] "C:/Users/bmm100/Documents/EDE_Fall2023"
```

```
#Loading necessary libraries
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 error
```

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

```
## here() starts at C:/Users/bmm100/Documents/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
#Reading processed data files
processedNTL.LTER.data <-
↪  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
processedNiwot.Ridge.data <-
↪  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2 Making sure R is reading dates as date format
#Checking date format of the loaded data for the North Temperate Lakes LTER Data
glimpse(processedNTL.LTER.data)
```

```
## Rows: 23,008
## Columns: 15
## $ lakename        <chr> "Paul Lake", "Paul Lake", "Paul Lake", "Paul Lake", "P~
## $ year4           <int> 1984, 1984, 1984, 1984, 1984, 1984, 1984, 1984, 1984, ~
## $ daynum          <int> 148, 148, 148, 148, 148, 148, 148, 148, 148, 148, 148,~
## $ month           <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ sampledate      <chr> "1984-05-27", "1984-05-27", "1984-05-27", "1984-05-27"~
## $ depth           <dbl> 0.00, 0.25, 0.50, 0.75, 1.00, 1.50, 2.00, 3.00, 4.00, ~
## $ temperature_C   <dbl> 14.5, NA, NA, NA, 14.5, NA, 14.2, 11.0, 7.0, 6.1, 5.5,~
## $ dissolvedOxygen <dbl> 9.5, NA, NA, NA, 8.8, NA, 8.6, 11.5, 11.9, 2.5, 1.6, 0~
## $ irradianceWater <dbl> 1750.0, 1550.0, 1150.0, 975.0, 870.0, 610.0, 420.0, 22~
## $ irradianceDeck  <dbl> 1620, 1620, 1620, 1620, 1620, 1620, 1620, 1620, 1620, ~
## $ tn_ug           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ tp_ug           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ nh34            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ no23            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ po4             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```
#Adjusting date format using lubridate
processedNTL.LTER.data$sampledate <- ymd(processedNTL.LTER.data$sampledate)
#Using class function to show that date format has been updated successfully
class(processedNTL.LTER.data$sampledate)
```

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

```
#Checking date format of the loaded data for the Niwot Ridge Long-Term Ecological
↪  Research (LTER) station
glimpse(processedNiwot.Ridge.data)
```

```
## Rows: 1,692
## Columns: 13
## $ plotID            <chr> "NIWO_062", "NIWO_061", "NIWO_062", "NIWO_064", "NIWO~
## $ trapID            <chr> "NIWO_062_050", "NIWO_061_169", "NIWO_062_050", "NIWO~
## $ collectDate       <chr> "2016-06-16", "2016-06-16", "2016-06-16", "2016-06-16~
## $ functionalGroup   <chr> "Seeds", "Other", "Woody material", "Seeds", "Needles~
## $ dryMass           <dbl> 0.000, 0.270, 0.120, 0.000, 1.110, 0.000, 0.000, 0.00~
## $ qaDryMass         <chr> "N", "N", "N", "N", "Y", "N", "N", "N", "N", "N", "N"~
## $ subplotID         <int> 31, 41, 31, 32, 32, 32, 40, 40, 40, 40, 40, 31, 31, 3~
## $ decimalLatitude   <dbl> 40.05114, 40.04762, 40.05114, 40.04737, 40.04872, 40.~
## $ decimalLongitude  <dbl> -105.5858, -105.5861, -105.5858, -105.5840, -105.5872~
## $ elevation         <dbl> 3477.0, 3413.4, 3477.0, 3373.2, 3446.4, 3446.4, 3509.~
## $ nlcdClass         <chr> "shrubScrub", "evergreenForest", "shrubScrub", "everg~
## $ plotType          <chr> "tower", "tower", "tower", "tower", "tower", "tower",~
## $ geodeticDatum     <chr> "WGS84", "WGS84", "WGS84", "WGS84", "WGS84", "WGS84",~
```

```
processedNiwot.Ridge.data$collectDate <- ymd(processedNiwot.Ridge.data$collectDate)
#Using class function to show that date format has been updated successfully
class(processedNiwot.Ridge.data$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 Building personal theme
brian_theme_A05 <- theme_base()  +
  theme(
    line =  element_line(colour = "grey10"),
    rect =  element_rect(colour = "grey10", fill = "Honeydew"),
    text =  element_text(colour = "grey10",size = 12),

    # Modified inheritance structure of text element
    plot.title = element_text(family = "Helvetica",
                              face = "bold",
                              size = 20,colour = "grey10"),
    axis.title.x = element_text(family="Helvetica",size = 12,colour = "grey10",face =
    ↪   "bold"),
    axis.title.y = element_text(family="Helvetica",size = 12,colour = "grey10",face =
    ↪   "bold"),
    axis.text = element_text(family="Helvetica",size = 12,colour = "grey10"),

    # Modified inheritance structure of line element
    axis.ticks = element_blank(),
    panel.grid.major =  element_line(colour = "grey80"),
    panel.grid.minor =  element_blank(),
    panel.border = element_rect(colour = "grey80"),

    # Modified inheritance structure of rect element
    plot.background = element_rect(fill = "Honeydew",colour = NA),
    panel.background =  element_rect(fill = "Honeydew",colour = "grey80"),
    legend.key =  element_rect(fill="Honeydew"),

    # Modifiying legend.position
    legend.position = "bottom",
    legend.background = element_rect(colour = "grey10"),
    legend.text = element_text(colour = "grey10")

    #complete = TRUE
    )

#Setting personal theme as default theme
theme_set(brian_theme_A05)
```

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()`).

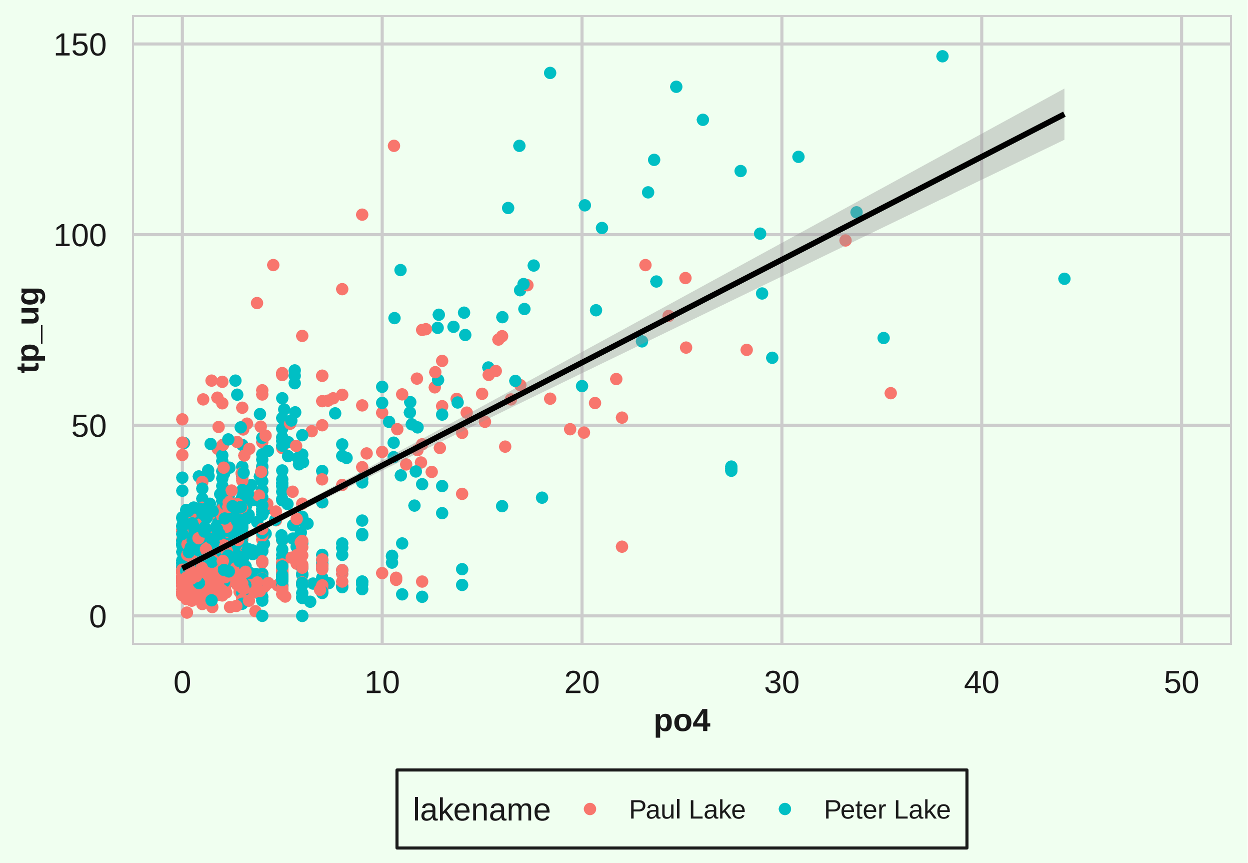
```r
#4 Setting up plot
NTL.LTER.Question4Plot <- processedNTL.LTER.data %>%
  ggplot(aes(x=po4,y=tp_ug,color=lakename)) + geom_point() + xlim(0,50) + ylim(0,150) +
  ↪  geom_smooth(method=lm, color="black")

#Printing out plot
NTL.LTER.Question4Plot
```

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



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: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```r
#5 Setting up the boxplots
#Checking how month variables are stored
class(processedNTL.LTER.data$month)
```

```
## [1] "integer"
```

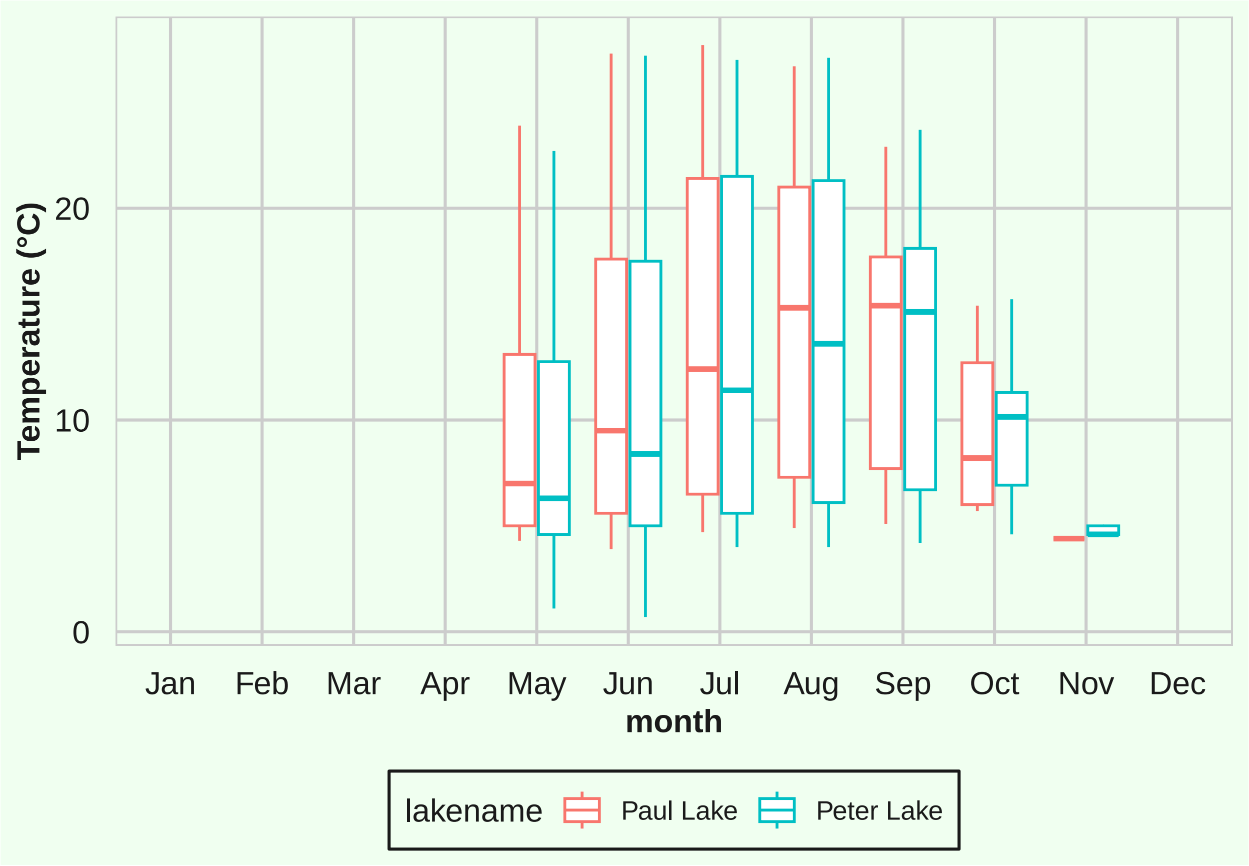
```r
#Since month is stored as an integer, it needs to be converted to a factor to probably
↪  plot the months on the x axis. This is done in the respective pipes for the different
↪  visualisations outlined below.

#Setting up boxplots of temperatures in the different months of the year
temperatureBoxplot.NTL.LTER <- processedNTL.LTER.data %>%

  ↪  ggplot(aes(x=factor(month,levels=1:12,labels=month.abb),y=temperature_C,color=lakename))+geom_boxp
  ↪  (\u00B0C)")

#Displaying temperature boxplot
temperatureBoxplot.NTL.LTER
```

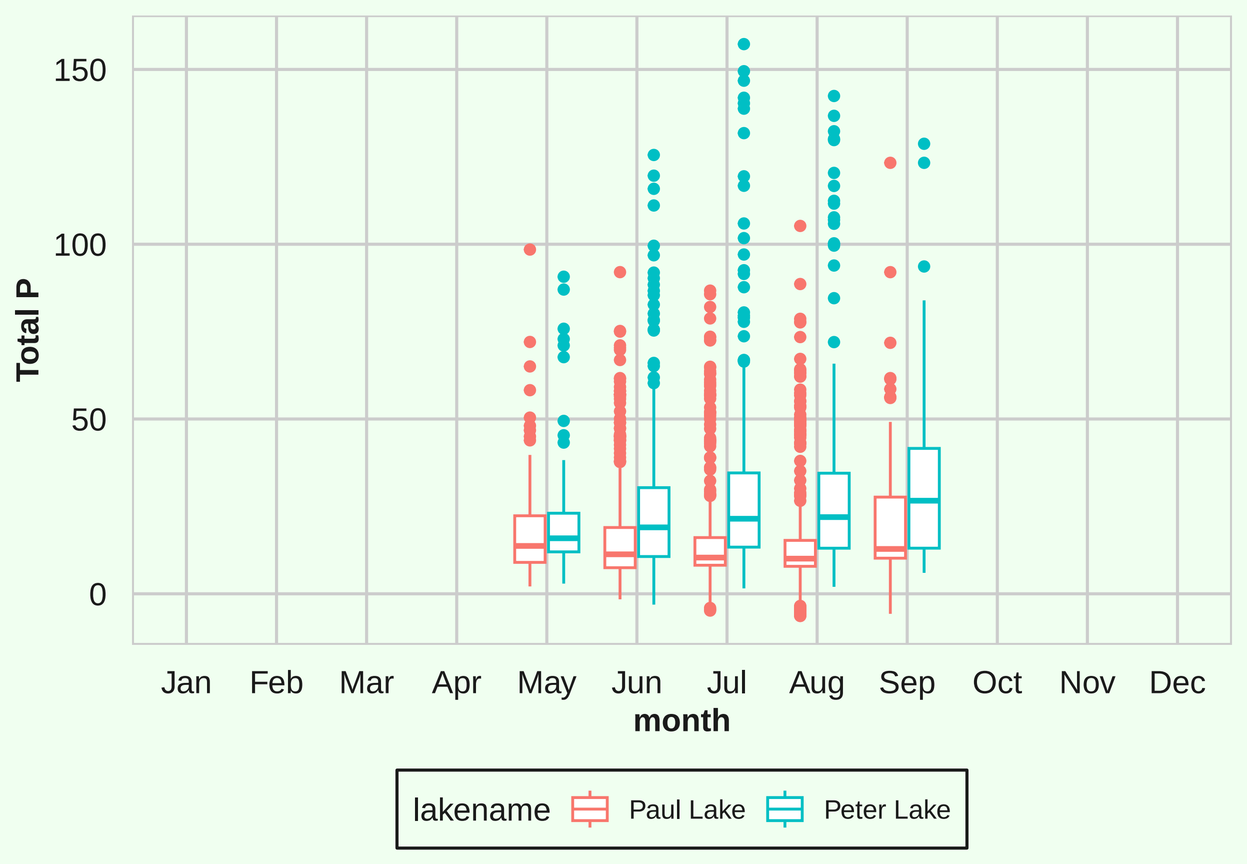
```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).
```



```
#Setting up boxplots of TP in the different months of the year
TPBoxplot.NTL.LTER <- processedNTL.LTER.data %>%
                                                                                           
   ↪   ggplot(aes(x=factor(month,levels=1:12,labels=month.abb),y=tp_ug,color=lakename))+geom_boxplot()+s
   ↪   P")

#Displaying temperature boxplot
TPBoxplot.NTL.LTER
```

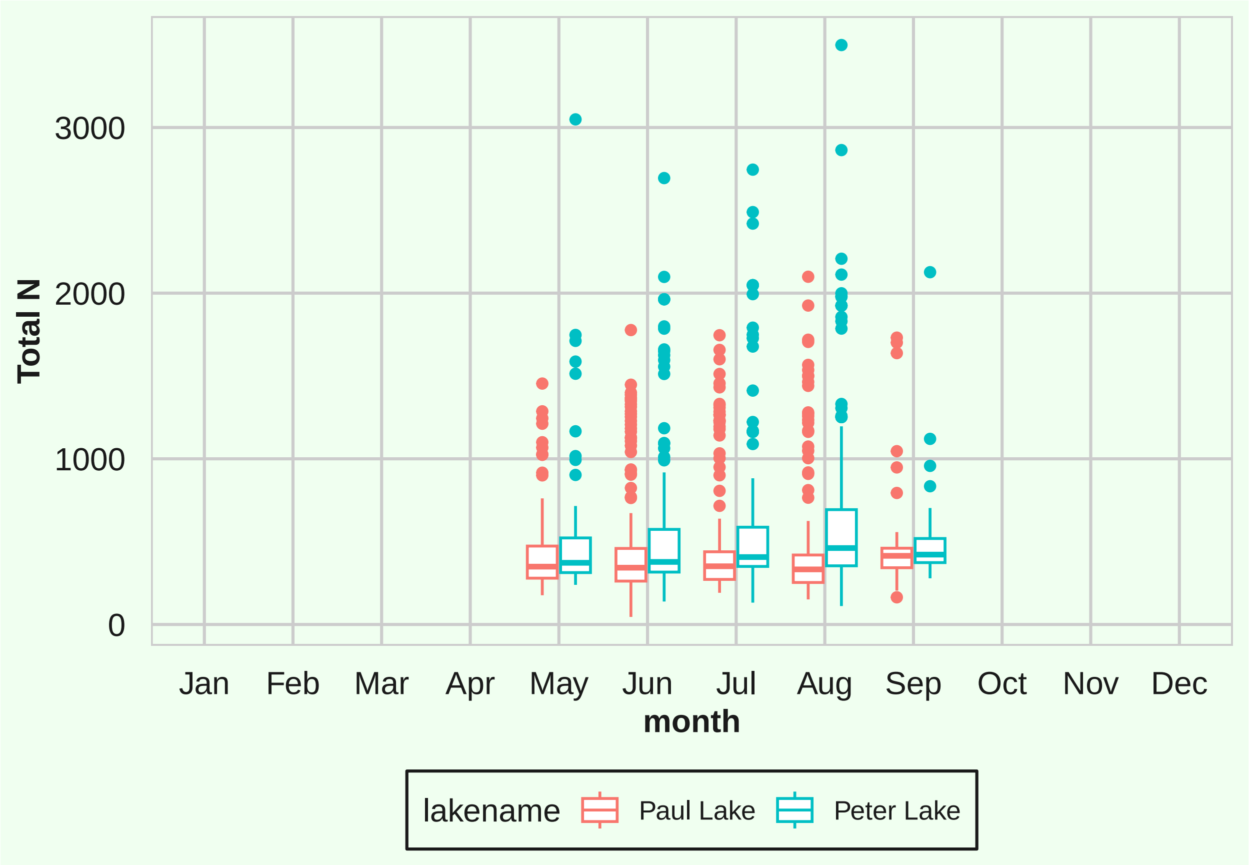
```
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```



```
#Setting up boxplots of TN in the different months of the year
TNBoxplot.NTL.LTER <- processedNTL.LTER.data %>%
                                                                                           
   ↪   ggplot(aes(x=factor(month,levels=1:12,labels=month.abb),y=tn_ug,color=lakename))+geom_boxplot()+s
   ↪   N")

#Displaying temperature boxplot
TNBoxplot.NTL.LTER
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

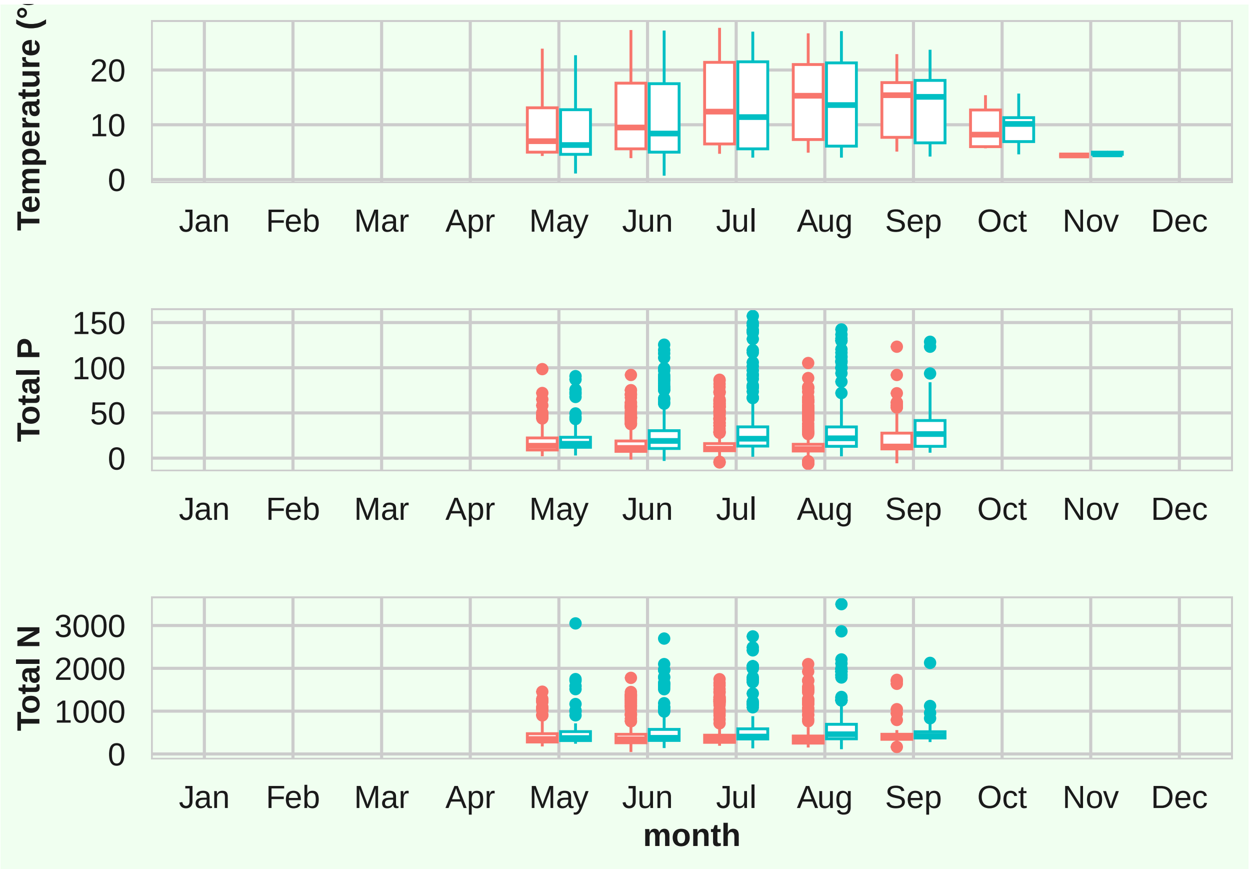


```r
#Consolidating the 3 boxplots created above
plot_grid(temperatureBoxplot.NTL.LTER+theme(legend.position ="none",axis.title.x =
↪  element_blank()),TPBoxplot.NTL.LTER+theme(legend.position ="none",axis.title.x =
↪  element_blank()),TNBoxplot.NTL.LTER+theme(legend.position ="none"),nrow = 3,align =
↪  "hv")
```

```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

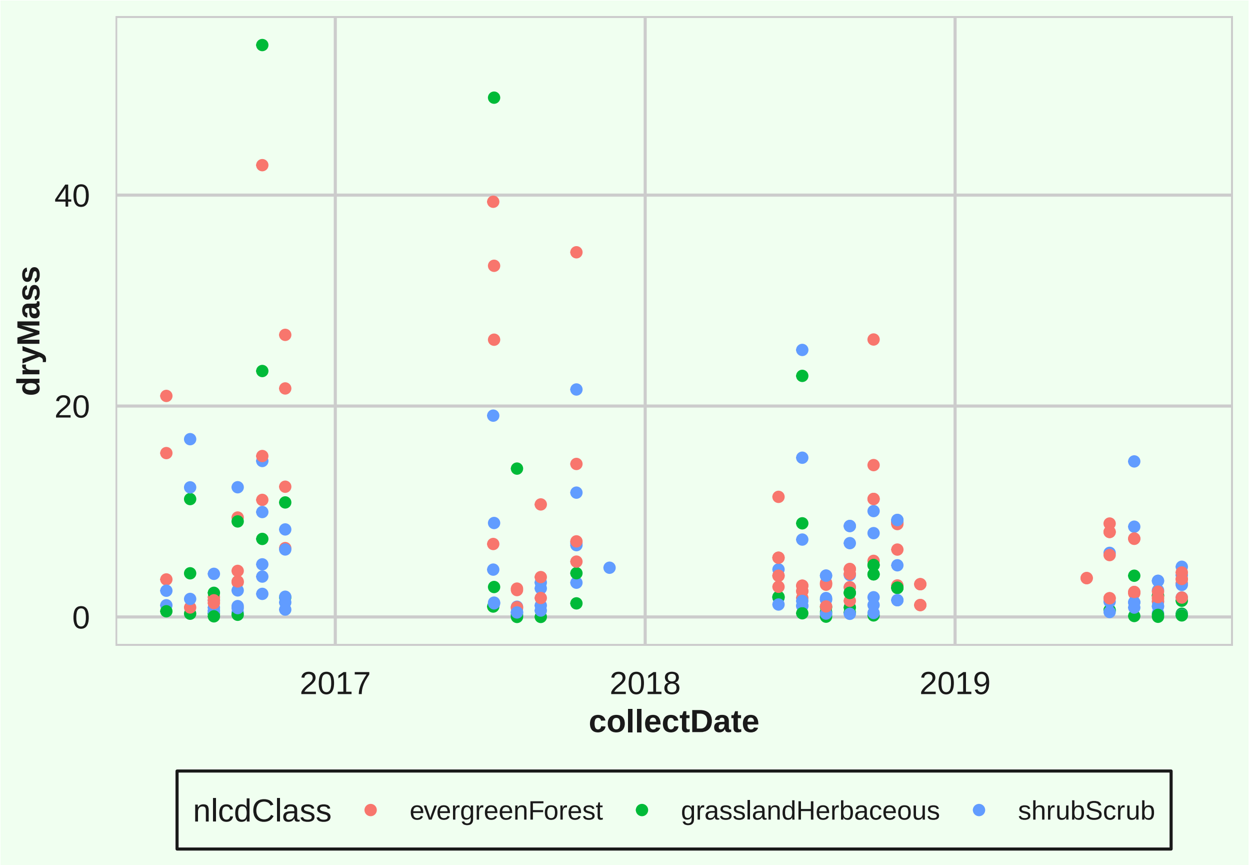


Question: What do you observe about the variables of interest over seasons and between lakes?

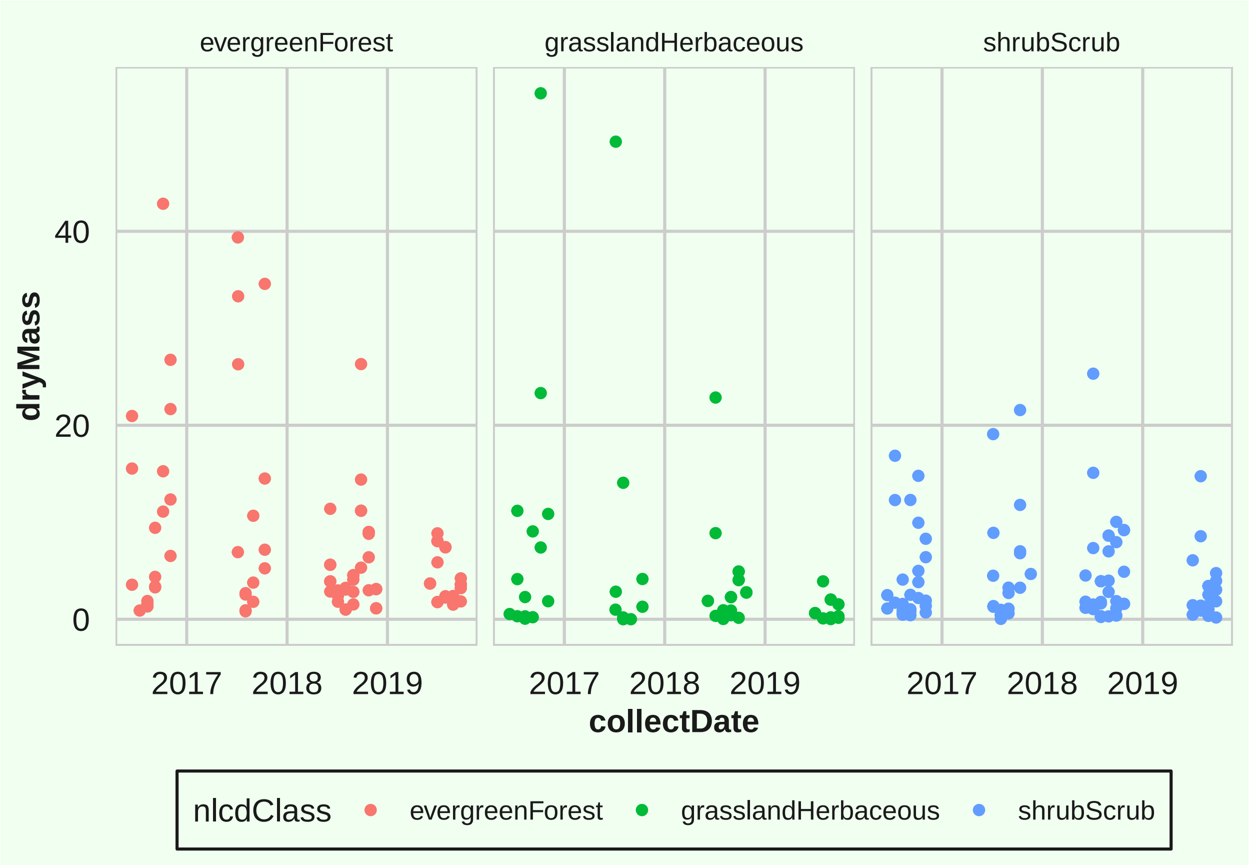
> Answer: From the boxplots, we observe the following concerning the variables: 1. The median temperature in both lakes rises between the months of May and August then drops down in the months of October and November. The temperature distributions in the months of May to July are also positively skewed and the temperatures in both lakes fall within a similar range. 2. Total P and N values are fairly concentrated with several outliers.

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 Plotting a subset of the litter dataset displaying only the "Needles" functional group
NiwotRideDryMassbyDate.plot <- processedNiwot.Ridge.data %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x=collectDate,y=dryMass,color=nlcdClass)) + geom_point()
NiwotRideDryMassbyDate.plot
```



```
#7 Plotting the same data separated into facets
NiwotRideDryMassbyDate.facetedplot <- processedNiwot.Ridge.data %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x=collectDate,y=dryMass,color=nlcdClass)) + geom_point() +
  ↪  facet_wrap(vars(nlcdClass),nrow = 1) + theme(legend.position = "bottom")
NiwotRideDryMassbyDate.facetedplot
```



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

> Answer: Plot 7 is more effective than Plot 6 as one can more easily infer the patterns in dryMass for the different nlcd classes.