Assignment 5: Data Visualization

Gretchen Barbera

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
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 errors
```

```r
library (lubridate)
library(cowplot)
```

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

```r
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```r
library(readr)
library(ggplot2)
library(dplyr)
library(tidyr)
here()
```

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

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

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



#2

#class(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate)
#initially reads as factor format


NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate <-
  as.Date(
    NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate, format = "%m/%d/%y"
    )

class(
  NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$sampledate
  )
```

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

```r
#now it is date

#class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

```
#factor originally

NEON_NIWO_Litter_mass_trap_Processed$collectDate <- as.Date(
    NEON_NIWO_Litter_mass_trap_Processed$collectDate)

class(NEON_NIWO_Litter_mass_trap_Processed$collectDate)
```

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

```
#now it is in "Date" format
```

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

my_theme <-
  theme(
    plot.title = element_text(
      size= 16,
      face= "bold",
      color = "darkblue",
      hjust = 5
    ),
    axis.title.x = element_text(size = 12, face = "italic", color = "darkblue"),
    axis.title.y = element_text(size = 12, face = "italic", color= "darkblue"),
    axis.text = element_text(size= 12, color = "lightgreen"),
    axis.ticks = element_line(color = "lightgreen"),
    panel.grid.major = element_line(color = "lightblue", size = 0.5),
    panel.grid.minor = element_blank(),
    plot.background = element_rect(fill= "white"),
    panel.background = element_rect(fill="white"),
    legend.key = element_rect(fill = "white"),
    legend.position = "right",
    complete = TRUE
)
```

```
## 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.
```

```
theme_set(my_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()`).

```
#4
#
# #ggplot(NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed, aes(
#   x= po4,
#   y= tp_ug,
#   color = lakename)) +
#   geom_point(size= .5) +
#   geom_smooth(method = "lm", se = FALSE) +
#   labs(title= "Total Phosphorus By Phosphate",
#        x= "Phosphate (PO4)",
#        y= "Total Phosphorus (Tp_ug)",
# color = "lakename")+
#    coord_fixed(ratio = 1) +
#   xlim(-0.2,400)+
#   ylim(-10,160) +
#   my_theme

#When i did this I got a lot of warning messages that there are NAs/infinite values
#It looks terrible? I need to filter out
#the NA values because there is
#a lot of them and I think they are ruining my plot?

summary(
  NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$po4
  )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -0.233   1.000   2.324   5.919   5.000 373.836   21822
```

```
summary(
  NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed$tp_ug
  )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -6.349   9.194  14.401  22.159  27.746 157.250   20729
```

```
#I adjusted the limits based on the values I got from the summary
```

```
NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed<-
  NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed %>%
  filter(
    is.finite(po4) & is.finite(tp_ug)
    )



NTL_clean <-
NTL_LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed %>%
  filter(!is.na(po4) & !is.na(tp_ug))
summary(NTL_clean)
```

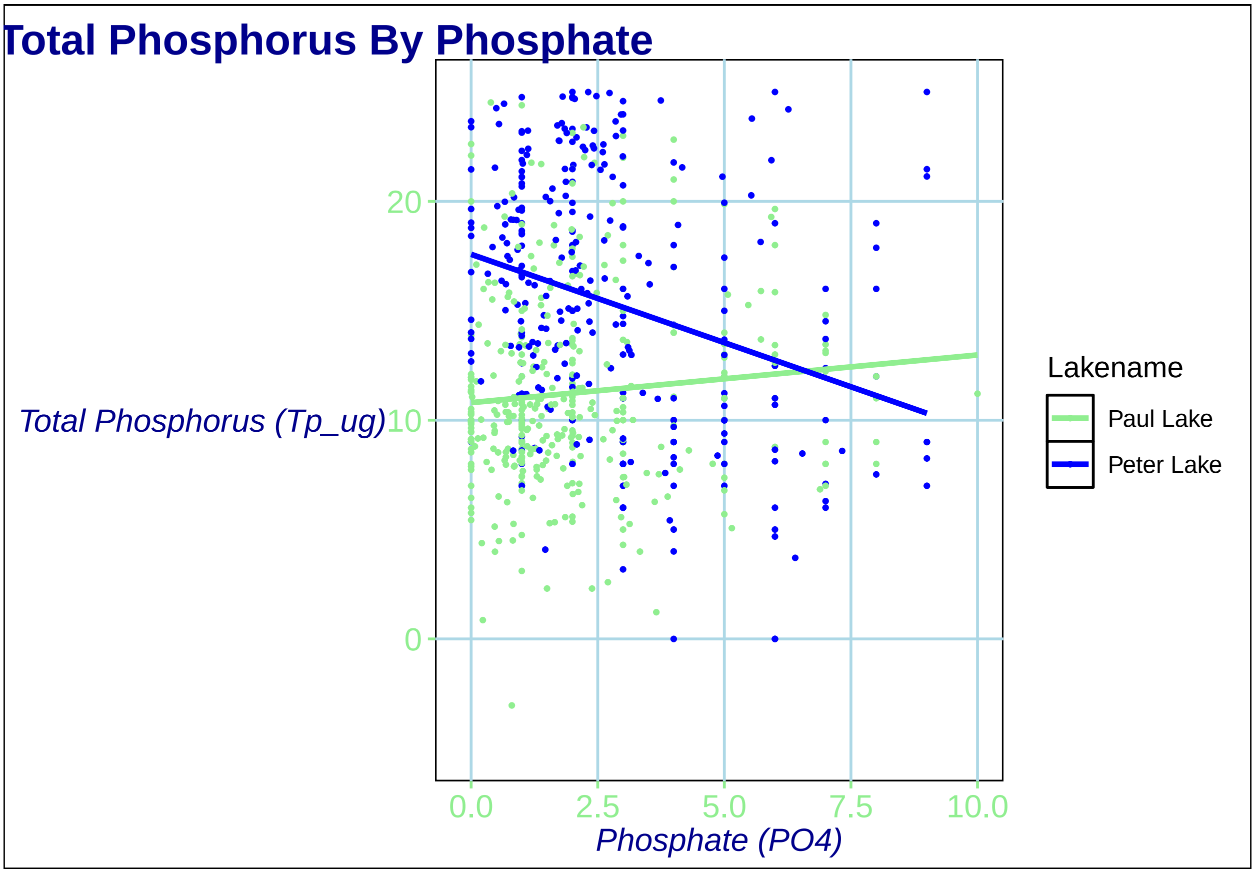
```
##        lakename       year4          daynum          month         sampledate
##  Paul Lake :537   Min.   :1991   Min.   :139.0   Min.   :5.000   Min.   :NA
##  Peter Lake:525   1st Qu.:1992   1st Qu.:167.0   1st Qu.:6.000   1st Qu.:NA
##                   Median :1994   Median :193.0   Median :7.000   Median :NA
##                   Mean   :1994   Mean   :192.7   Mean   :6.825   Mean   :NaN
##                   3rd Qu.:1995   3rd Qu.:218.0   3rd Qu.:8.000   3rd Qu.:NA
##                   Max.   :2013   Max.   :250.0   Max.   :9.000   Max.   :NA
##                                                                  NA's   :1062
##      depth        temperature_C   dissolvedOxygen  irradianceWater
##  Min.   : 0.000   Min.   : 4.10   Min.   : 0.000   Min.   :   0.8
##  1st Qu.: 0.800   1st Qu.: 5.20   1st Qu.: 0.500   1st Qu.:  79.5
##  Median : 2.525   Median :16.70   Median : 7.600   Median : 250.0
##  Mean   : 3.602   Mean   :14.28   Mean   : 6.196   Mean   : 441.7
##  3rd Qu.: 5.400   3rd Qu.:21.30   3rd Qu.: 9.400   3rd Qu.: 775.5
##  Max.   :12.000   Max.   :27.30   Max.   :20.000   Max.   :1550.0
##                   NA's   :601     NA's   :601      NA's   :731
##  irradianceDeck       tn_ug             tp_ug              nh34
##  Min.   :  41.0   Min.   :  45.67   Min.   : -3.039   Min.   :   0.000
##  1st Qu.: 360.5   1st Qu.: 348.66   1st Qu.: 10.013   1st Qu.:   6.916
##  Median : 736.0   Median : 432.06   Median : 16.331   Median :  12.485
##  Mean   : 664.2   Mean   : 631.19   Mean   : 23.903   Mean   : 115.863
##  3rd Qu.: 976.0   3rd Qu.: 696.40   3rd Qu.: 28.745   3rd Qu.:  28.977
##  Max.   :1473.0   Max.   :3497.70   Max.   :146.782   Max.   :2713.684
##  NA's   :731      NA's   :283                         NA's   :68
##       no23             po4
##  Min.   :  0.000   Min.   :  0.000
##  1st Qu.:  1.660   1st Qu.:  1.000
##  Median :  2.862   Median :  2.352
##  Mean   : 19.855   Mean   :  4.579
##  3rd Qu.:  7.440   3rd Qu.:  5.000
##  Max.   :866.249   Max.   :373.836
##  NA's   :37
```

```
summary(NTL_clean$po4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.352   4.579   5.000 373.836
```

```r
#now the NAs are gone and my data is easier to read

# ggplot(NTL_clean, aes(
#   x= po4,
#   y= tp_ug,
#   color = lakename)) +
#   geom_point(size= .5) +
#   geom_smooth(method = "lm", se = FALSE) +
#   labs(title= "Total Phosphorus By Phosphate",
#        x= "Phosphate (PO4)",
#        y= "Total Phosphorus (Tp_ug)",
#        color= "Lakename") +
#   xlim(-0.2,400)+
#   ylim(-10,160) +
#     scale_color_manual(values = c( "Paul Lake"= "lightgreen", "Peter Lake"= "blue"))+
#   my_theme

#unique(NTL_clean$lakename)

#why is it so skinny? I am going to change the parameters so it makes my data
#look better- but uses they numbers originally because I looked at the summary
#values for the phospohorus and did based on the max and min- I will readjust
#so we can see the numbers better

 ggplot(NTL_clean, aes(
  x= po4,
  y= tp_ug,
  color = lakename)) +
  geom_point(size= .5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title= "Total Phosphorus By Phosphate",
       x= "Phosphate (PO4)",
       y= "Total Phosphorus (Tp_ug)",
       color= "Lakename") +
  xlim(-0.2,10)+
  ylim(-5,25) +
    scale_color_manual(values = c( "Paul Lake"= "lightgreen", "Peter Lake"= "blue"))+
  my_theme
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 342 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).

## Warning: Removed 342 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



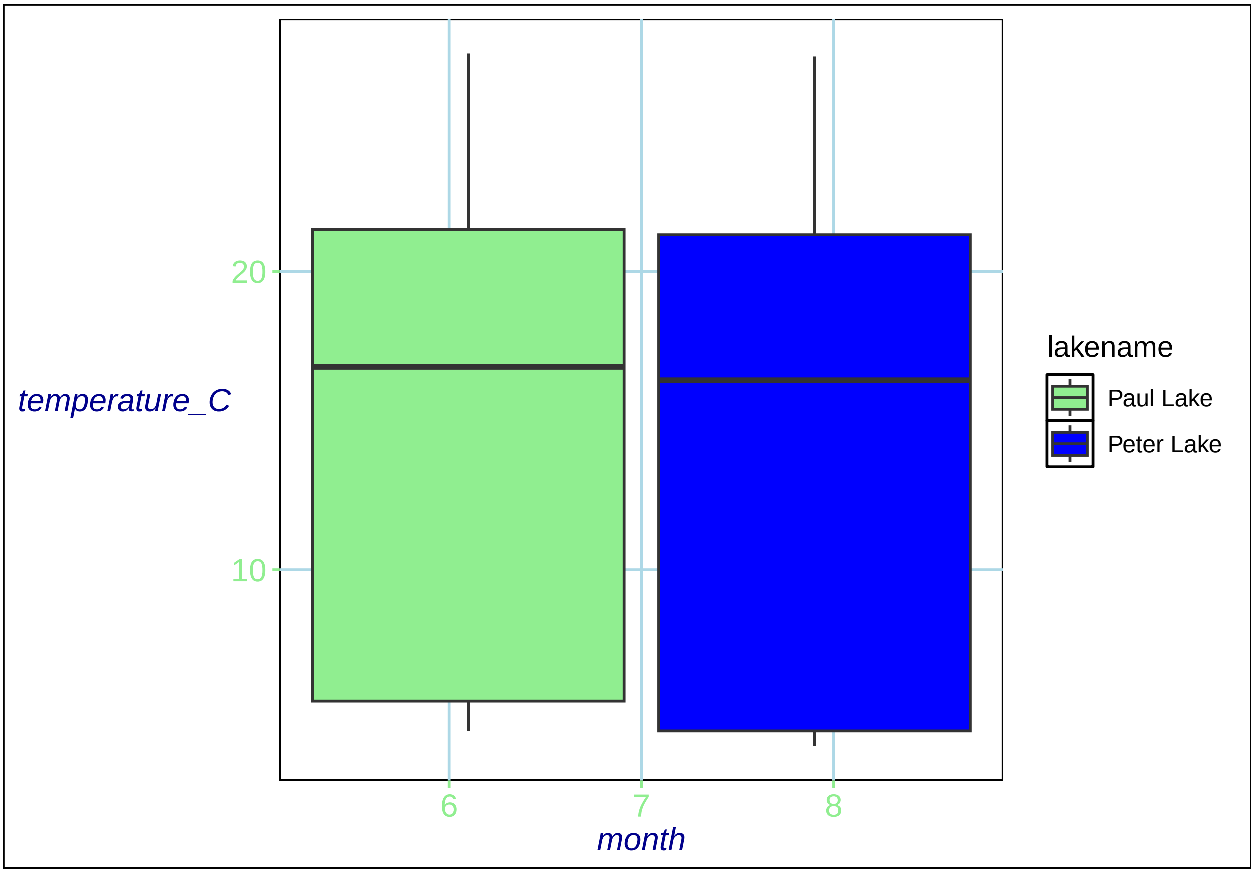
```
# summary(NTL_clean$tp_ug)- using the summary functions
 #i readjusted my parameters so we could see my data
```

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.

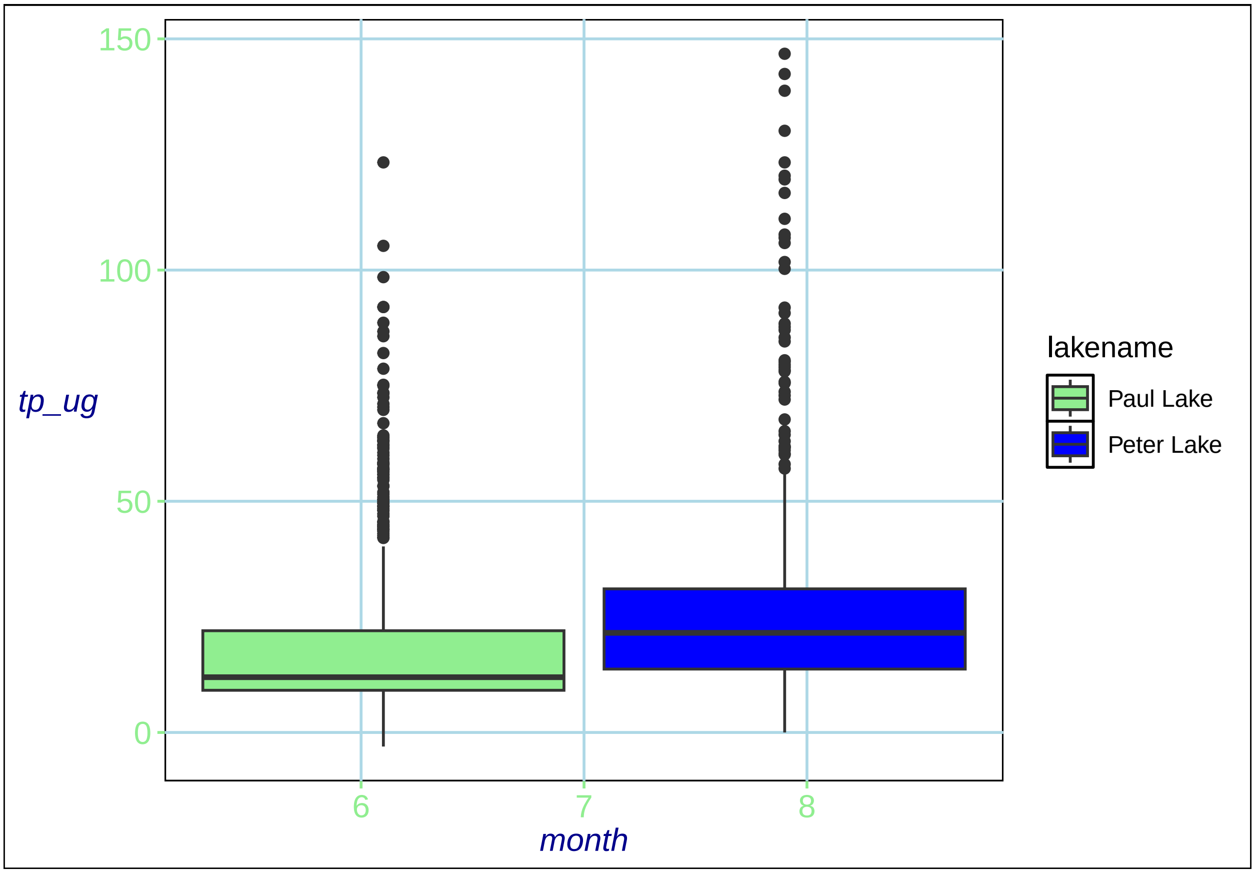
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a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

```
#5

BP1<-
  ggplot(NTL_clean, aes(
  x= month,
  y= temperature_C,
  fill= lakename
)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Peter Lake" = "blue", "Paul Lake"= "lightgreen")) +
  my_theme
print(BP1)
```

```
## Warning: Removed 601 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

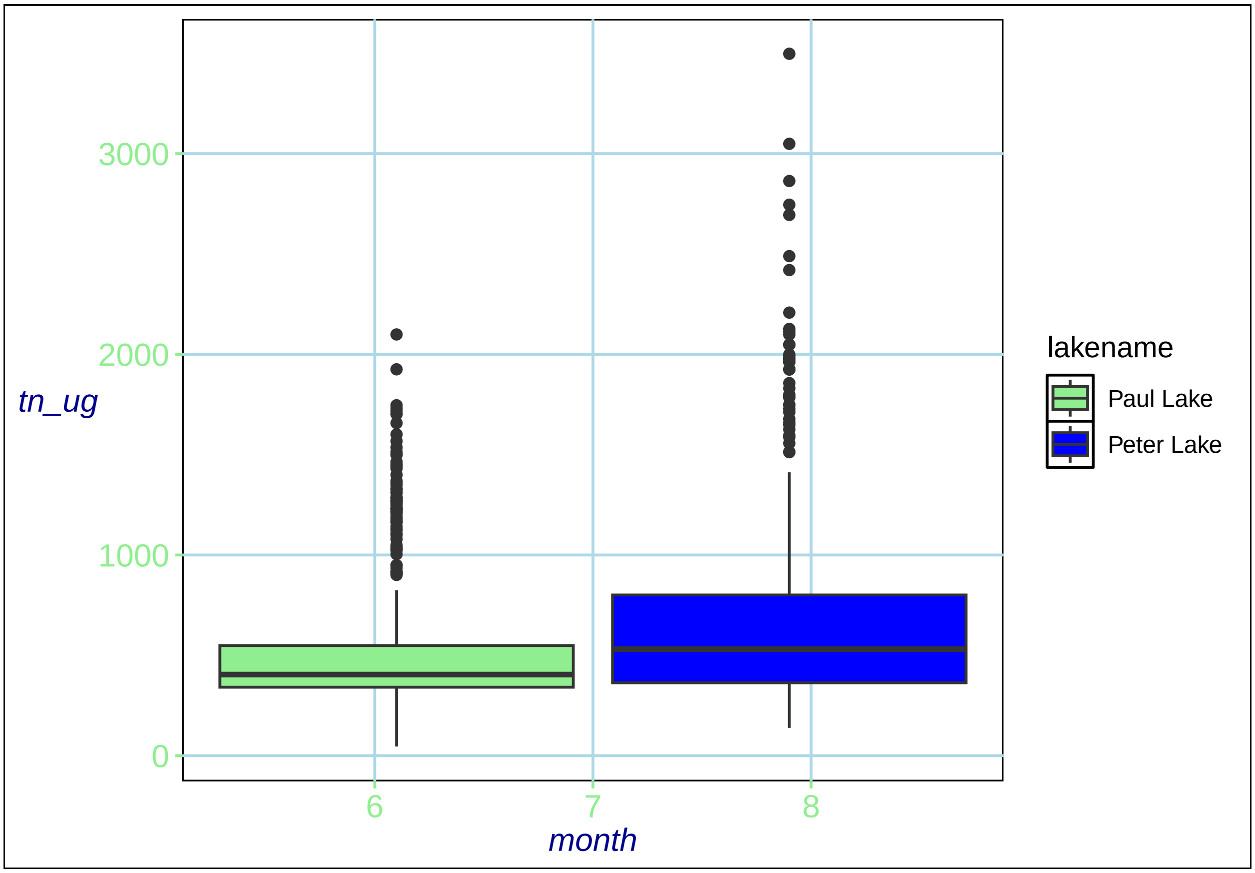


```r
#I hate the pink and blue so I changed the colors to be specific
#I also intentionally left off the title of the graph so it would look better with the cowplot

BP2 <- ggplot(NTL_clean, aes(
  x= month,
  y= tp_ug,
  fill= lakename
)) +
  geom_boxplot()+
  scale_fill_manual(values = c("Peter Lake" = "blue", "Paul Lake"= "lightgreen"))+
  my_theme
print(BP2)
```



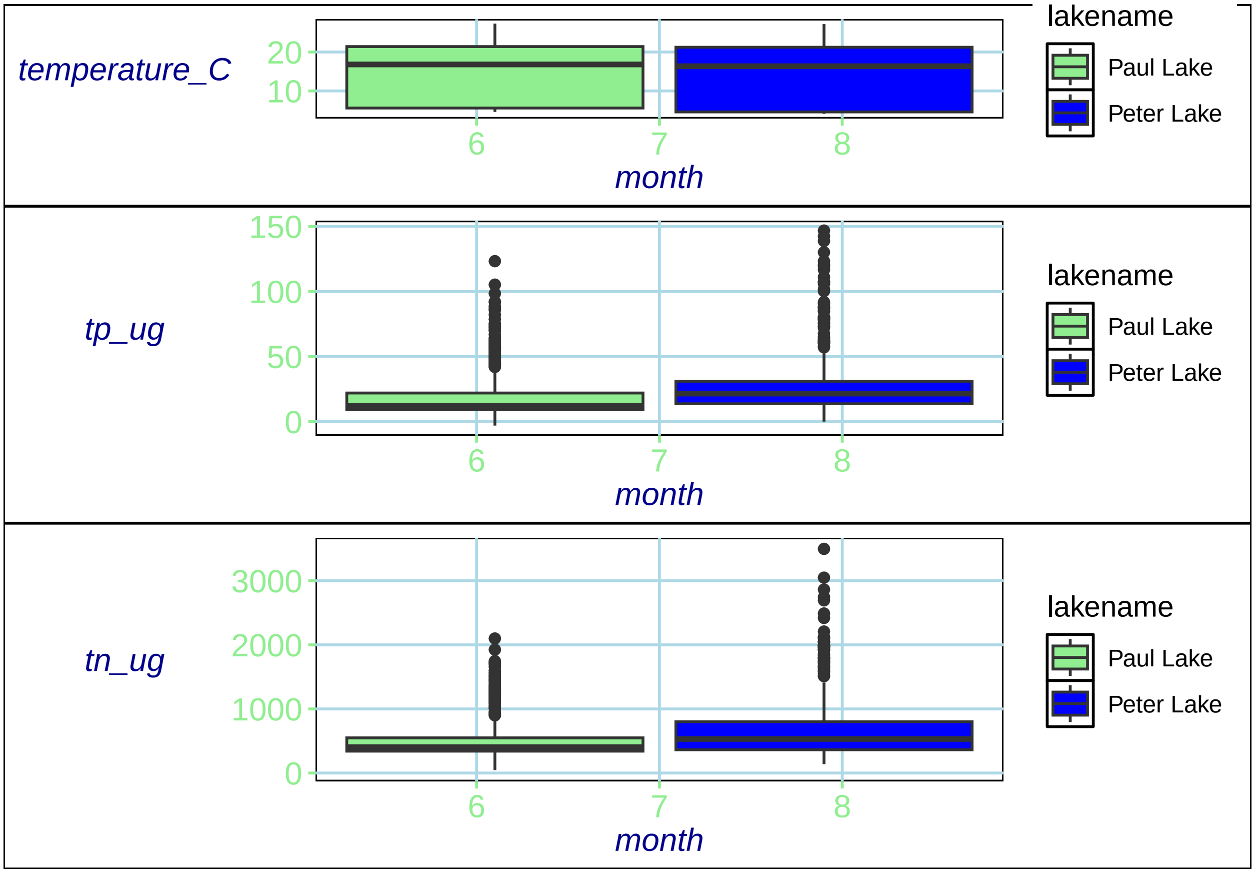
```
BP3 <- ggplot(NTL_clean, aes(
  x= month,
  y= tn_ug,
  fill= lakename
)) +
  geom_boxplot()+
  scale_fill_manual(values = c("Peter Lake" = "blue", "Paul Lake"= "lightgreen")) +
  my_theme
print(BP3)
```

```
## Warning: Removed 283 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



```
plot_grid(BP1, BP2, BP3, nrow=3, align = 'v', rel_heights = c(.7,1.1,1.2))
```

```
## Warning: Removed 601 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

## Warning: Removed 283 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



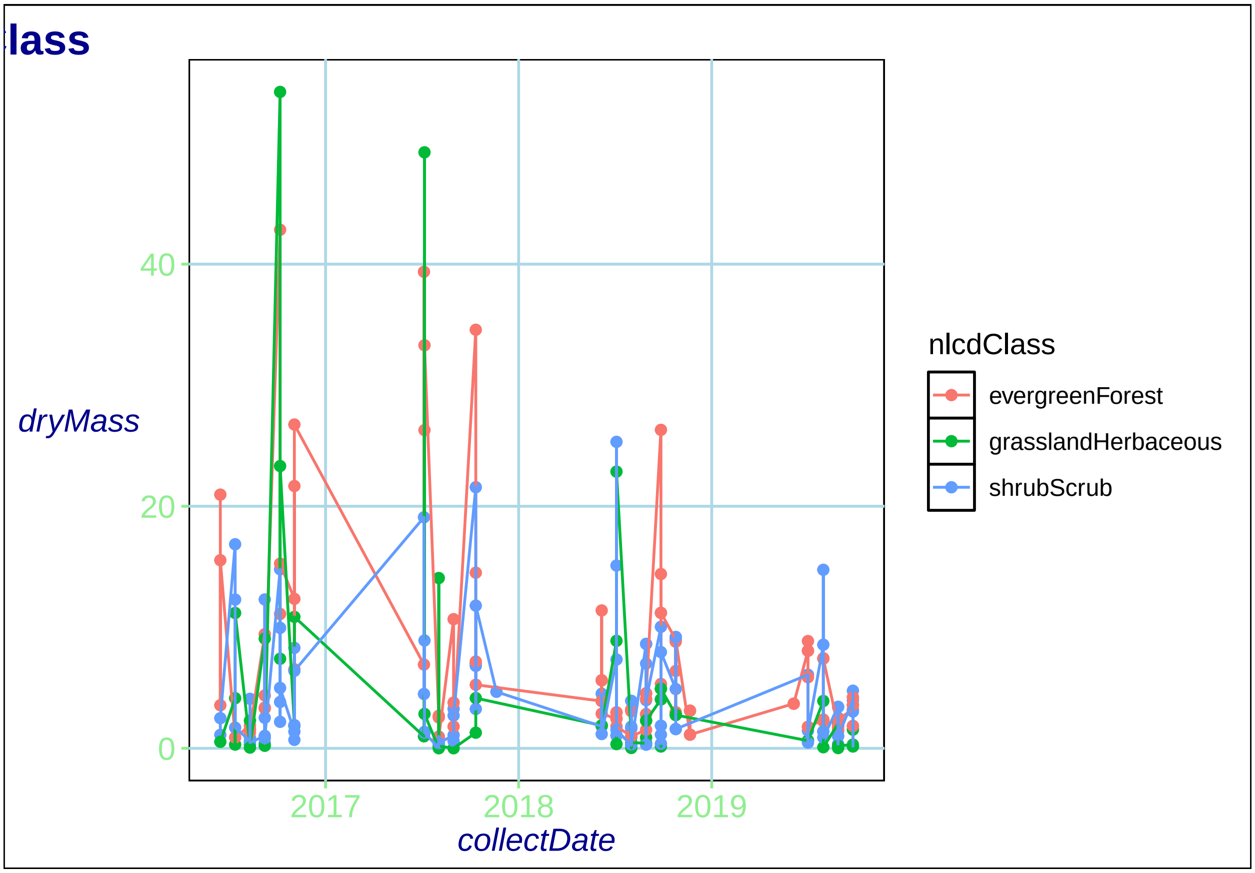
```
#it's not the cutest looking bot plot...
#i adjusted the sizes of the boxplots so we can see h
```

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

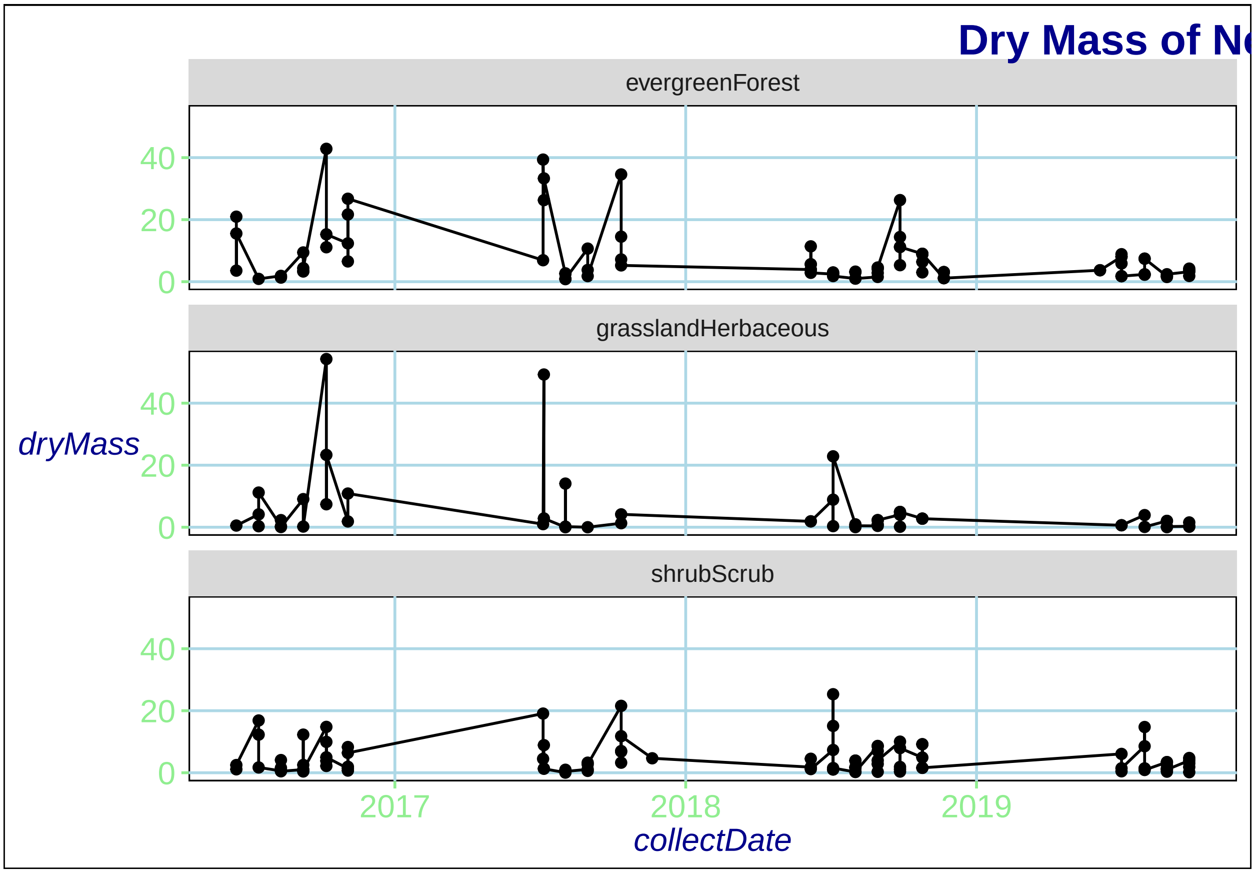
> Answer: The concentrations of tp_ug and tn_up follow similar trends. They are both higher in Peter Lake and are present between the 7th and 9th months respectively. Looking at the temperature, there between the 7th and 8th months are when we see the temperature increase in Peter Lake. However, the temperatures between Peter and Paul Lakes are similar, there is a higher concentration of both tp_ug and tn_up in Peter Lake.

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

needle_litter <-
  NEON_NIWO_Litter_mass_trap_Processed %>%
  filter(functionalGroup == "Needles")

ggplot(needle_litter,
        aes(
          x= collectDate,
          y= dryMass,
          color = nlcdClass)) +
  geom_point() +
  geom_line() +
  labs(title= "Dry Mass of Needle Litter By Date ans Class")+
  my_theme
```



```
#7



needle_litter <-
  NEON_NIWO_Litter_mass_trap_Processed %>%
  filter(functionalGroup == "Needles")

ggplot(needle_litter, aes(x= collectDate, y= dryMass)) +
  geom_point() +
  geom_line() +
  labs(title= "Dry Mass of Needle Litter By Date ans Class")+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  my_theme
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why? > Answer: They both have their value. Facets are easier to read and understand separately but seeing the different nlcdClasses together made it easier to visualize the other between them. The facets are cleaner to look at and I appreciate that they are separated so I can see and interpret the different dryMass values. Plot #6 was good for comparing values immediately and visualizing the difference in values. For myself, plot #7 is better for understanding the data.