Assignment 5: Data Visualization

Ayoung Kim

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 Loading the tidyverse, lubridate, here & cowplot
install.packages("tidyverse")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("here")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("cowplot")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
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 1.9.3     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 errors
```

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

```
## here() starts at /home/guest/EDE_Fall2023
```

```
library(cowplot)
```

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

```
library(ggplot2)

#Verify my home directory
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

```
#Reading datasets
nutrients_chem<-read.csv("./NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
neon_litter<-read.csv("./NEON_NIWO_Litter_mass_trap_Processed.csv")

#2 Change the "sampledate" to date format
# I used as.Date function to change the class of sampledate and collectdate to Date.

#Nutrients and chemistry/physics for Peter and Paul lakes
class(nutrients_chem$sampledate)
```

```
## [1] "character"
```

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

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

```
#Niwot Ridge Litter dataset
class(neon_litter$collectDate)
```

```
## [1] "character"
```

```
neon_litter$collectDate<-as.Date(neon_litter$collectDate,format="%Y-%m-%d")
class(neon_litter$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 Defining mytheme by customizing plot title ad legend
#I defined "mytheme", customizing the legend background/title/position and text size. Also, I assigned
mytheme<-theme_classic(base_size=13)+
  theme(
    legend.background = element_rect(color='grey',fill='beige'),
    legend.title=element_text(color='orange'),
    plot.title=element_text(color='black',size=15),
    legend.position="right",
    legend.text = element_text(size=11)
  )
```

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.

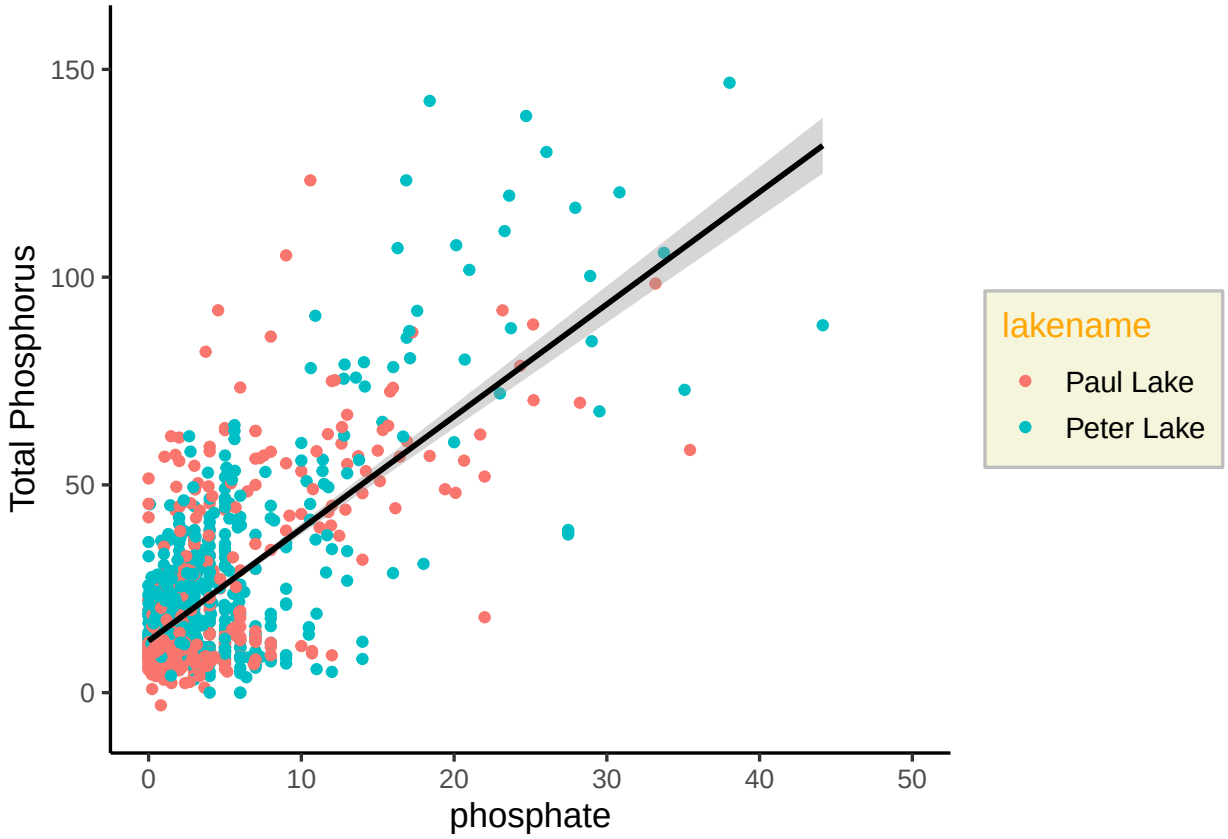
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4 Making a plot and adjust my axes to get rid of the extreme values
#I used po4 for x vlue and tp_ug for y value. I made sperated aesthetics for Peter and Paul lakes by us
ntl_plot<-ggplot(nutrients_chem,aes(x=po4, y=tp_ug, color=lakename))+
  geom_point()+
  mytheme+
  xlim(0,50)+
  labs(x="phosphate",y="Total Phosphorus")+
geom_smooth(method = lm, color='black')
print(ntl_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21947 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

```
#5 Making boxplots and cowplots
#I changed the month to factor
class(nutrients_chem$month)
```

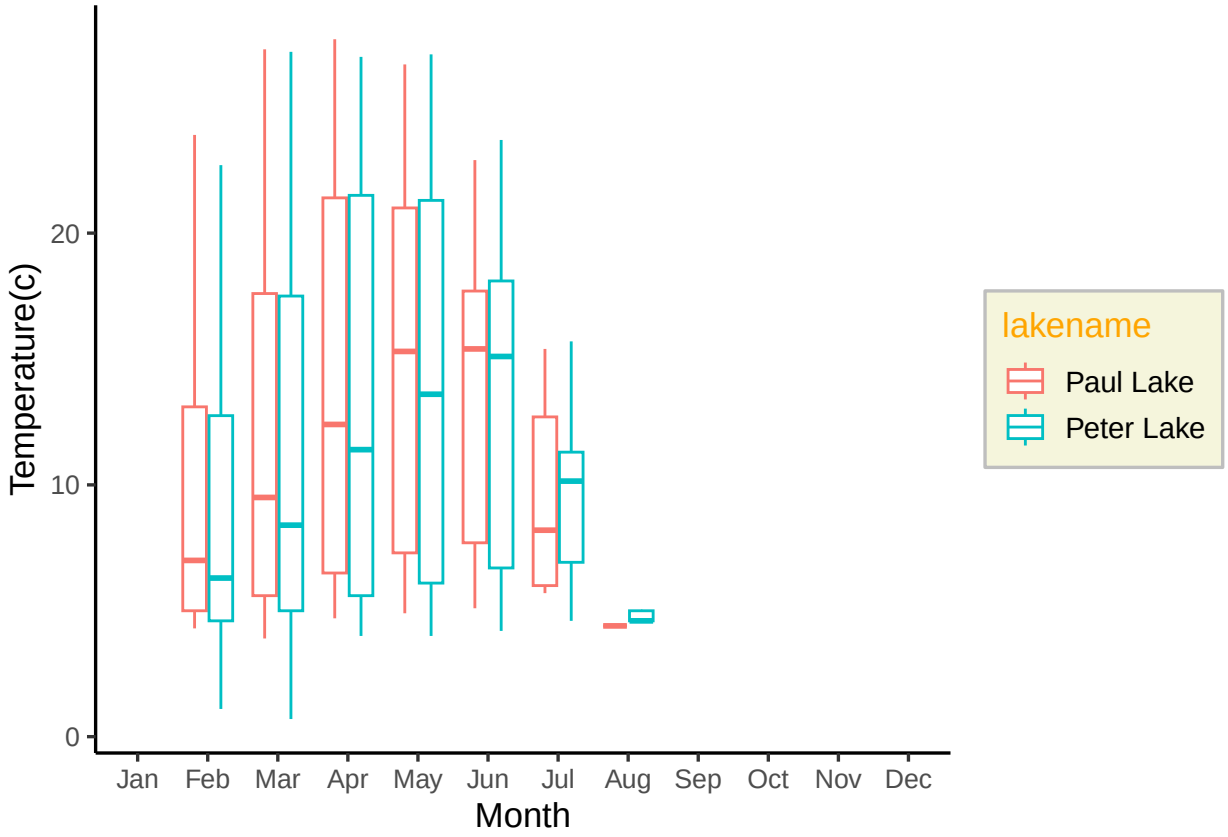
```
## [1] "integer"
```

```
nutrients_chem$month<-factor(nutrients_chem$month)
class(nutrients_chem$month)
```

```
## [1] "factor"
```

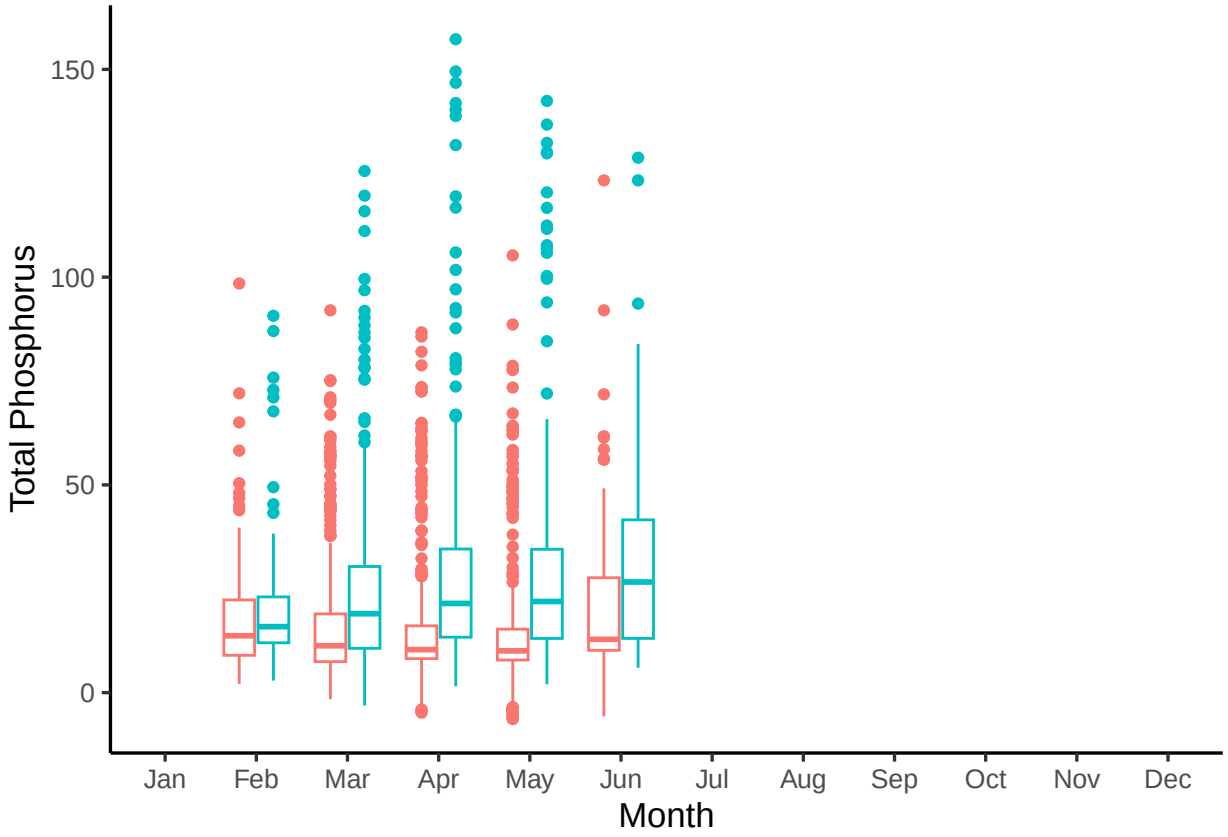
```
#I made three boxplots and one cowplot with mytheme. I used temperature, TP, TN, and month to get three
#Boxplot - (a)Temperature
ntl_plot_temp <-
  ggplot(nutrients_chem, aes(x=month.abb[month], y=temperature_C, color=lakename))+
  geom_boxplot(aes(color=lakename))+
  labs(x="Month",y="Temperature(c)")+ #Changed the label name
  mytheme+
  scale_x_discrete(limits=month.abb)
print(ntl_plot_temp)
```

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



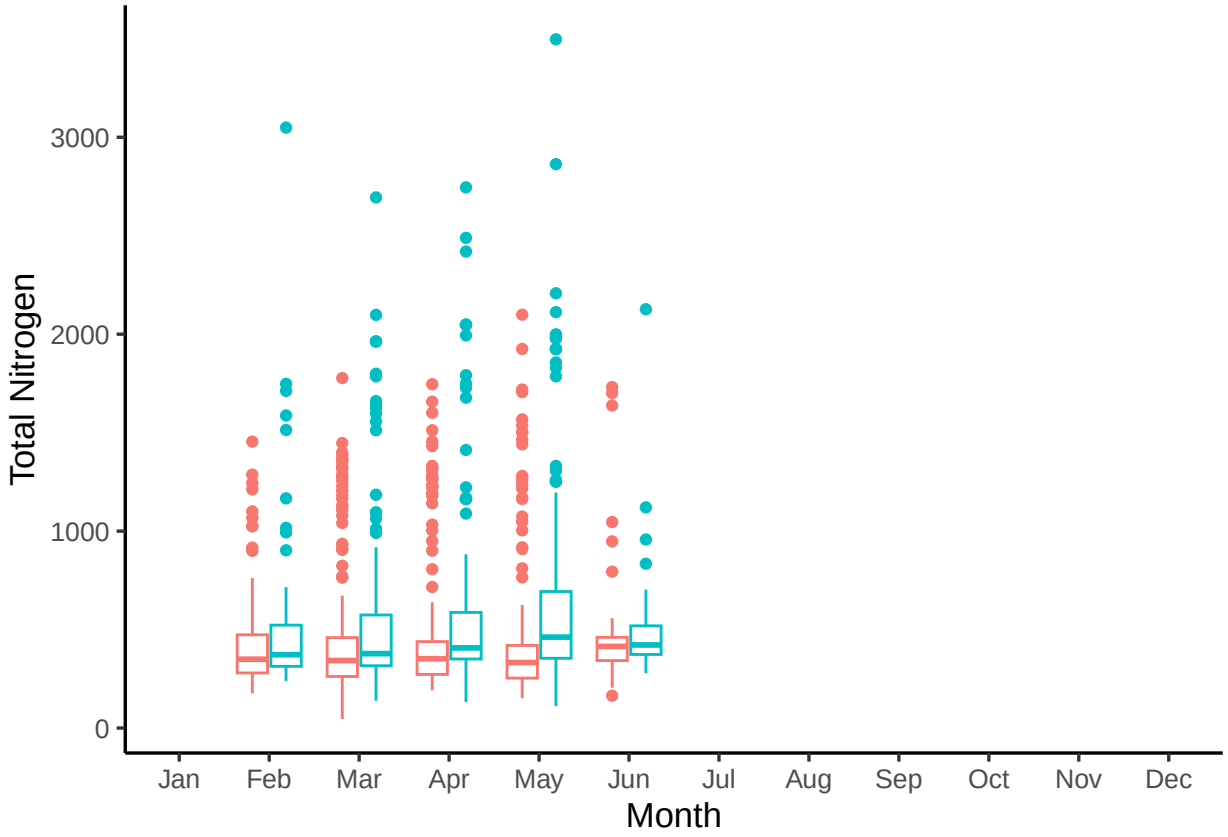

```
#Boxplot - (b)TP (Total Phosphorus)
ntl_plot_TP <-ggplot(nutrients_chem, aes(x=month.abb[month], y=tp_ug, color=lakename))+
  geom_boxplot(aes(color=lakename),show.legend = F)+
  labs(x="Month",y="Total Phosphorus")+
  mytheme+
  scale_x_discrete(limits=month.abb)
print(ntl_plot_TP)
```

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




```
#Boxplot - (c)TN (Total Nitrogen)
ntl_plot_TN<-ggplot(nutrients_chem, aes(x=month.abb[month], y=tn_ug, color=lakename))+
  geom_boxplot(aes(color=lakename),show.legend = F)+
  labs(x="Month",y="Total Nitrogen")+
  mytheme+
  scale_x_discrete(limits=month.abb)
print(ntl_plot_TN)
```

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




```
#A cowplot combining all three graphs
# I made a conbined plot which includes all of the three plots above.
cowplot1 <- plot_grid(ntl_plot_temp, ntl_plot_TP, ntl_plot_TN, ncol = 1, align = "v")
```

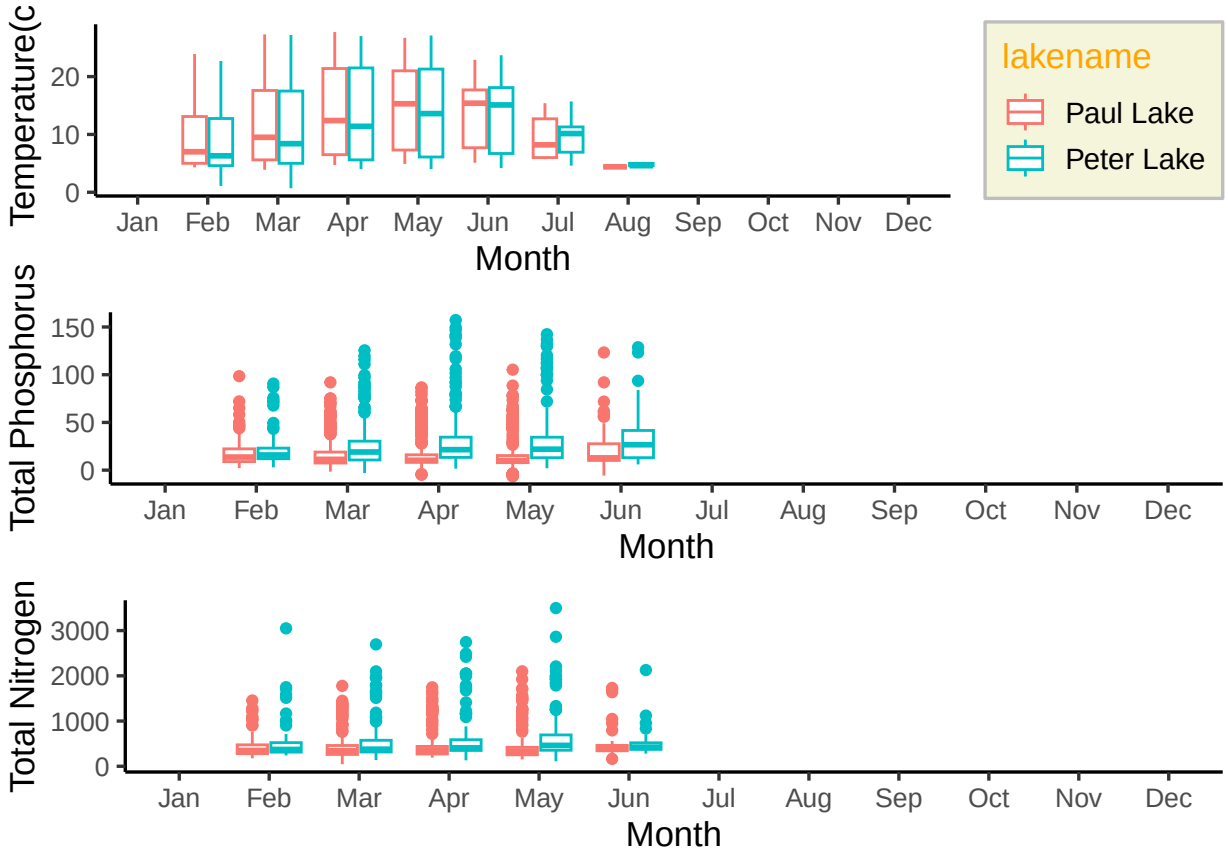
```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).
```

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

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

```
## Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
print(cowplot1)
```



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

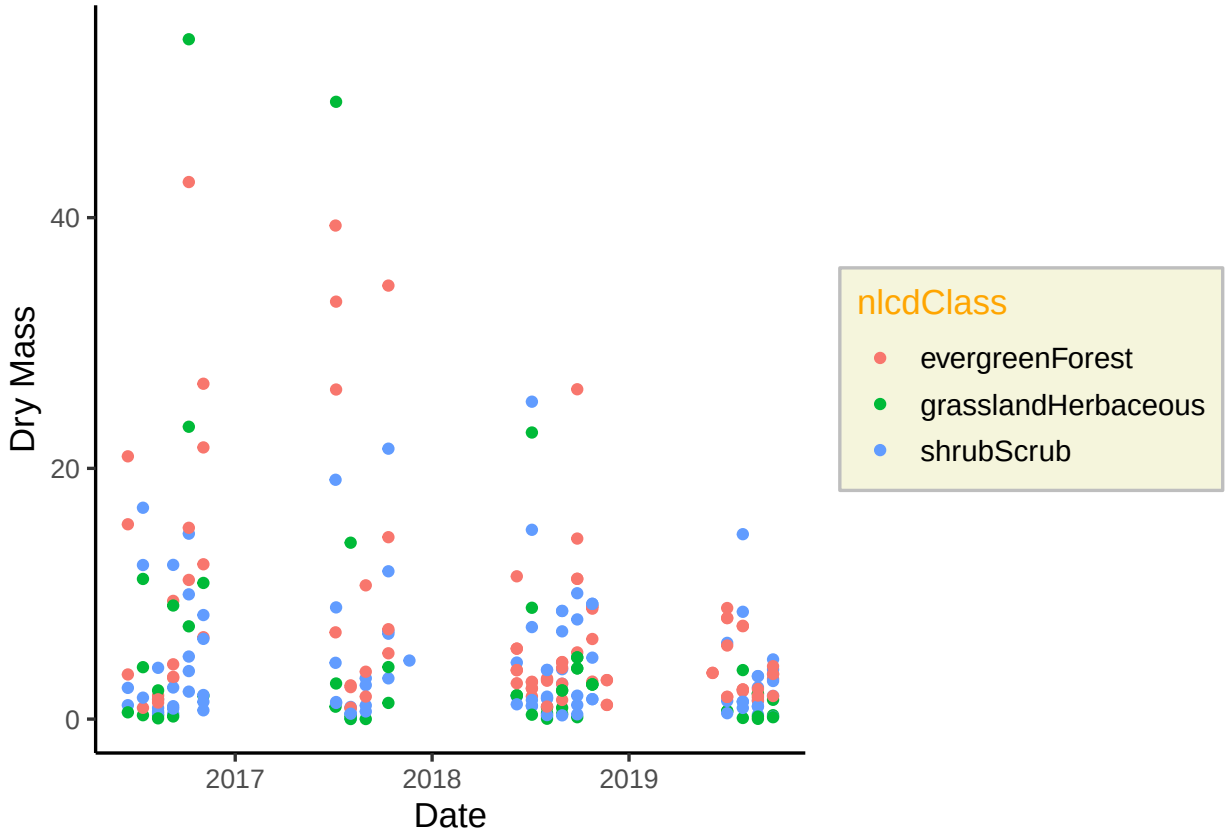
> Answer: Based on the cowplot, I could find that two lakes have quite similar trends in temperature from feburary to August. However, they have quite different values in Total Phosporous and Nitrogen. According to the plot, the Peter lake tends to have more phosphorous and it reacehd the peak in June. The Peter lake tends to have more nitrogen than Paul lake as well, but it reached the peak in May.

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)

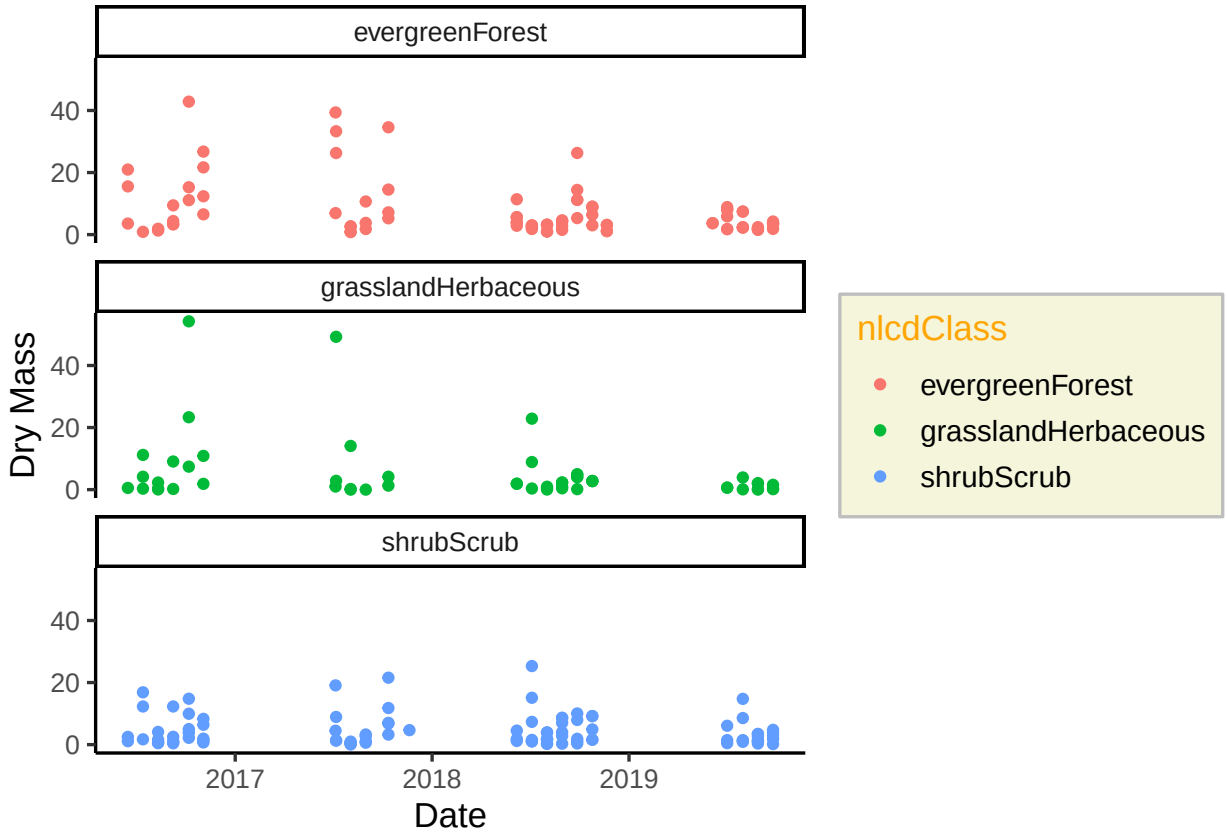
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6 Generating a plot of the dry mass of needle litter by date
#Made a subset of the litter datasets only with the "Needles" functional group
sub_niwot <-subset(neon_litter, functionalGroup=="Needles")

#Generated a plot of the dry mass by date and sepearted by NLCD class with a color aesthetic
niwot_plot1<-ggplot(sub_niwot, aes(x=collectDate, y=dryMass))+
  geom_point(aes(color=nlcdClass))+
  labs(x="Date",y="Dry Mass")+
  mytheme
print(niwot_plot1)
```




```
#7 Same plot with NLCD classes separated into three facets
#I seperated into three facets instead of sepeartin by color
niwot_plot2<-
  ggplot(sub_niwot, aes(x=collectDate, y=dryMass))+
  geom_point(aes(color=nlcdClass))+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  labs(x="Date",y="Dry Mass")+
  mytheme
print(niwot_plot2)
```



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

Answer: I think the plot with NLCD classes separated into three facets are more effecive than the other one, because I can compare the classes more easily and accurately. Compared to the plot in 6, which all classes are shown together, in plot 7, which classes are sepearted in each of the small sub-plot in one plot, I can compare three classes at once.