Assignment 5: Data Visualization

Camila Rodriguez

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
#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("here")
#install.packages("cowplot")

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
```

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

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

```
library(cowplot)
```

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

```
getwd()
```

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

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

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


#2


class(Litter$collectDate)
```

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

```
class(Nutrients$sampledate)
```

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

```
Litter$collectDate<- as.Date(Litter$collectDate, format='%Y-%m-%d')
Nutrients$sampledate<- as.Date(Nutrients$sampledate, format='%Y-%m-%d')
```

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
 mytheme<- theme_classic(base_size =12)+
  theme(panel.grid.major=element_line(color="grey"), legend.position = "bottom",
        legend.text = element_text(size = 9),
        legend.title = element_text(size = 9, face="bold"),legend.background = element_rect(color = "bla
        axis.text.x = element_text(size=10, hjust = 1))
```

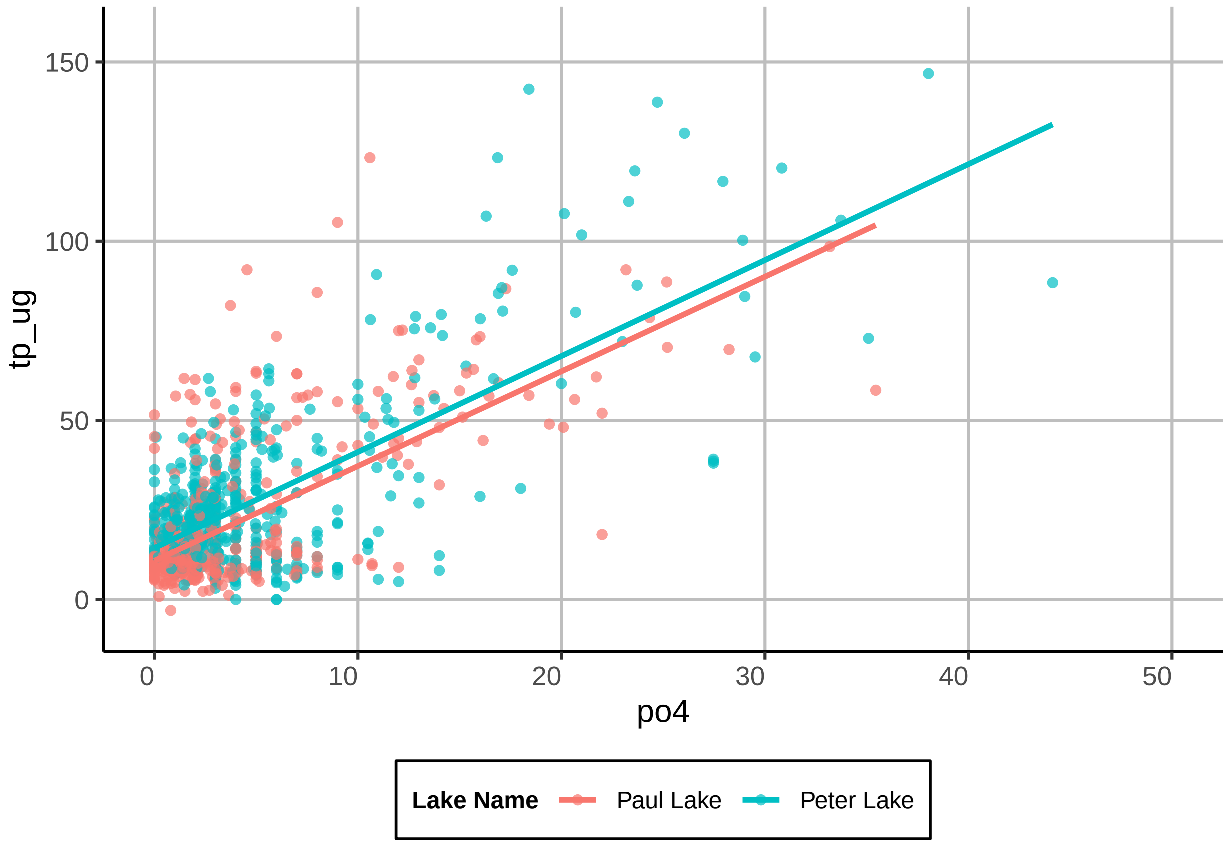
Create graphs

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

#create data frame with ggplot to plot po4 and tp_ug, then print to print plot

Totalphosphorous_by_phosphate<- ggplot(Nutrients, aes(x=po4, y=tp_ug,
                                                      color=lakename))+
  geom_point(alpha = 0.7, size = 1.3)+ geom_smooth(method = lm, se=F)+
  xlim(0,50) + labs(color= "Lake Name")+ mytheme

 print(Totalphosphorous_by_phosphate)
```

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



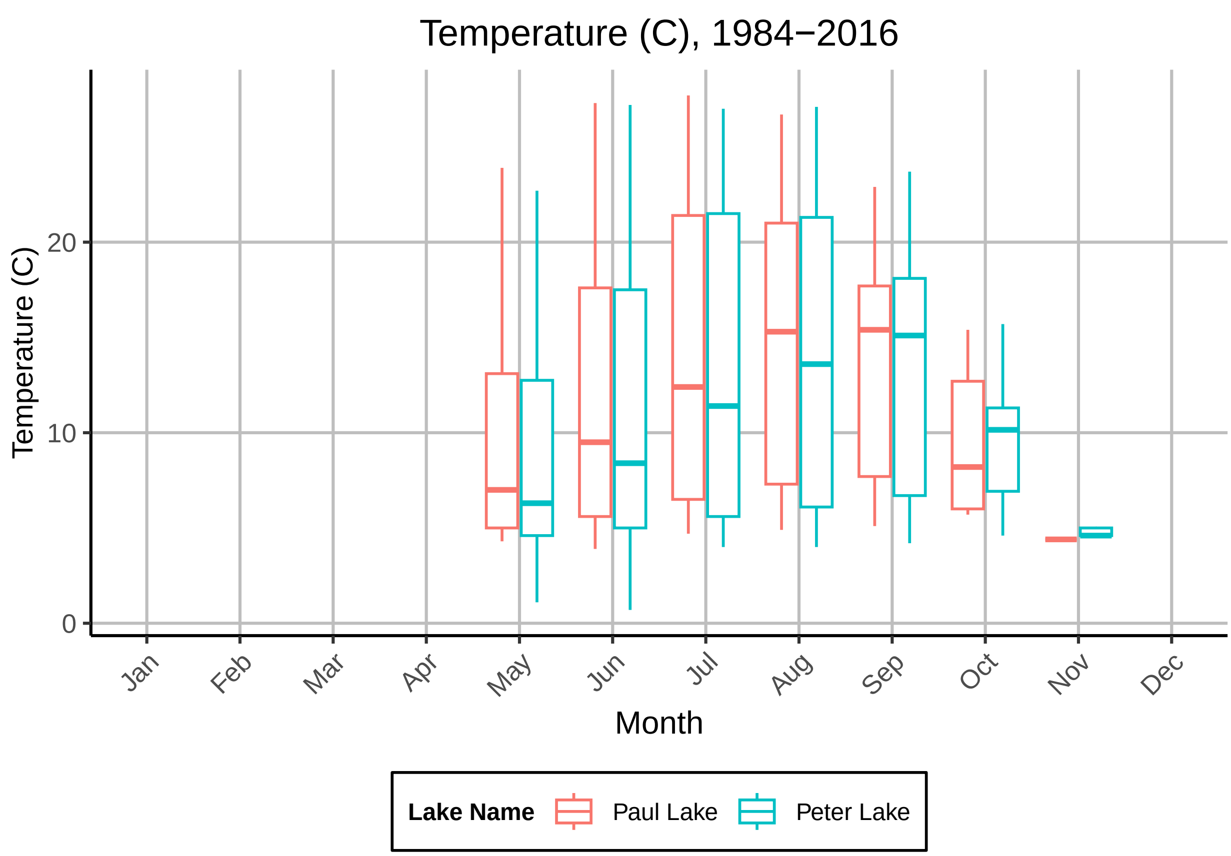
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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 three data frames,print to check, then combine with plot_grid function

Nutrient_temp<- ggplot(Nutrients, aes(x=factor(
  month, levels=1:12,labels=month.abb), y= temperature_C, color=lakename)) +
  geom_boxplot()+ scale_x_discrete(name='Month',drop=FALSE)+mytheme+
  labs(title= "Temperature (C), 1984-2016", y="Temperature (C)",
       color="Lake Name")+ mytheme+
  theme(plot.title = element_text(hjust = 0.5), legend.position ="bottom",
        axis.title.y =element_text(size=11),
        axis.text.x = element_text(angle = 45))

Nutrient_TP<- ggplot(Nutrients, aes(x=factor(month,levels = 1:12,labels =
                                               month.abb),y= tp_ug,
                                    color=lakename))+geom_boxplot()+
```

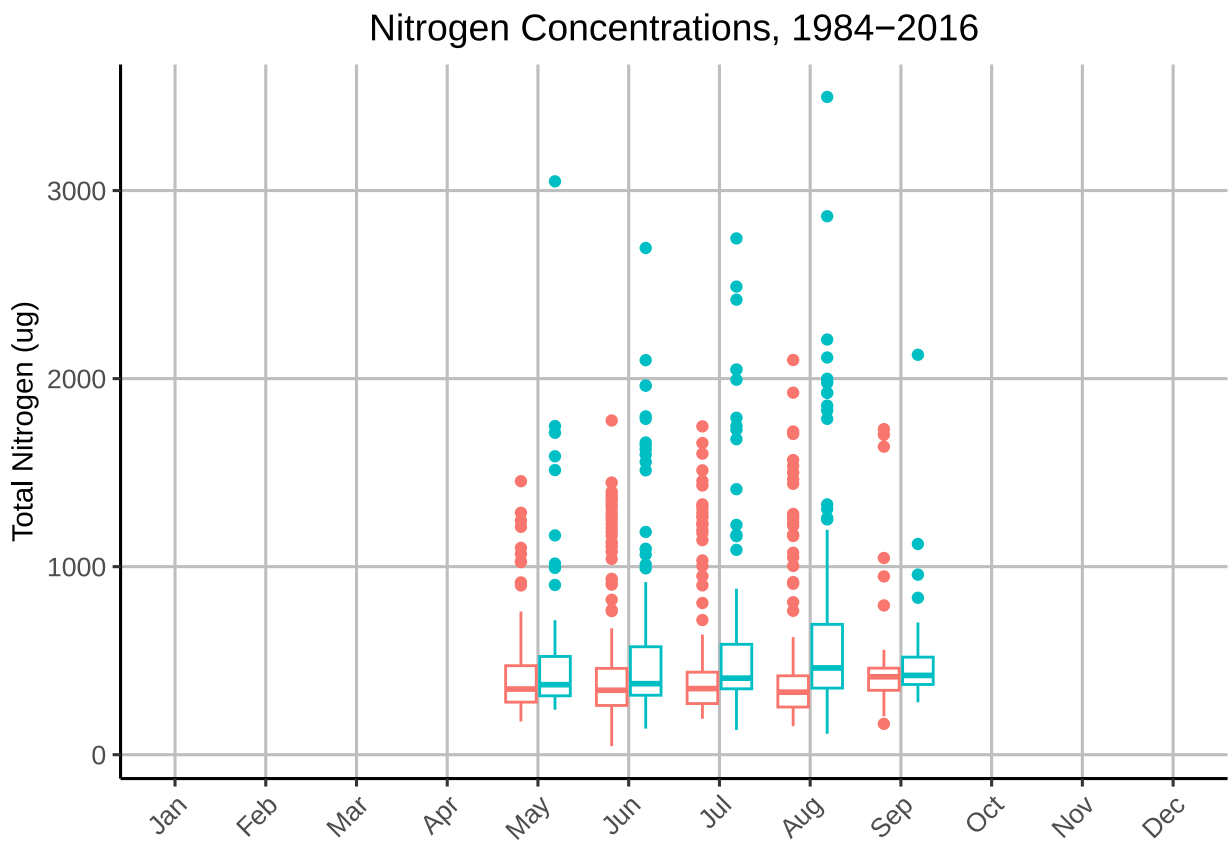
```
  scale_x_discrete(name='Month',drop=FALSE)+mytheme+
  theme(plot.title = element_text(hjust = 0.5), axis.title.x = element_blank(),
        axis.text.x = element_text(angle = 45),
        axis.title.y = element_text(size=11),  legend.position = "none") +
  labs(title="Phosphorous Concentrations, 1984-2016", y="Total Phosphorus (ug)",
       color="Lake Name")


Nutrient_TN<- ggplot(Nutrients, aes(x=factor(month,levels = 1:12,labels =
                                               month.abb) ,y= tn_ug,
                                    color=lakename))+geom_boxplot()+
  scale_x_discrete(name='Month',drop=FALSE)+
  mytheme + theme(plot.title = element_text(hjust = 0.5), legend.position =
  "none", axis.title.x = element_blank(),
  axis.title.y =element_text(size=11),axis.text.x = element_text(angle = 45))+
  labs(title="Nitrogen Concentrations, 1984-2016",y="Total Nitrogen (ug)",
       color="Lake Name")

#edit any axis under labs function,; scale_x_discrete function includes months with missing dates

print(Nutrient_temp)
```

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



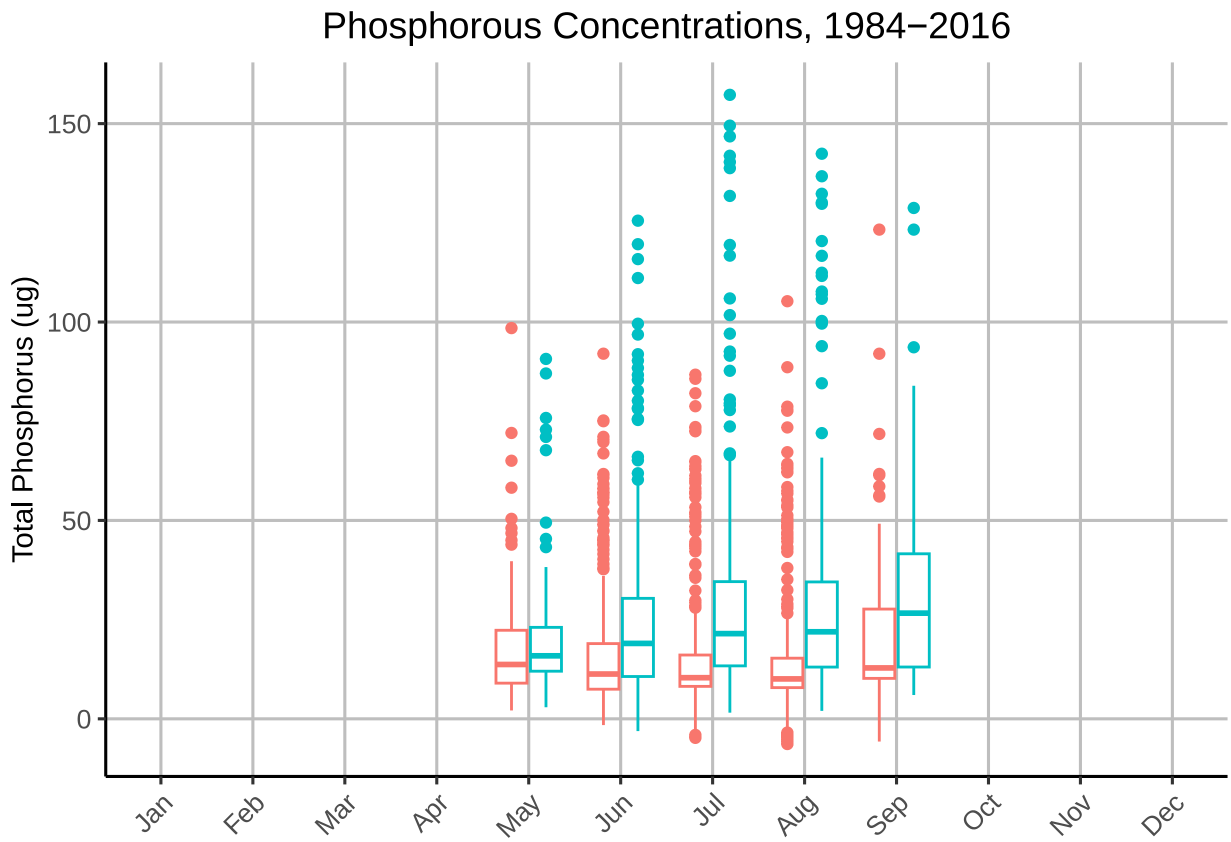
```r
print(Nutrient_TN)
```

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




```r
print(Nutrient_TP)
```

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



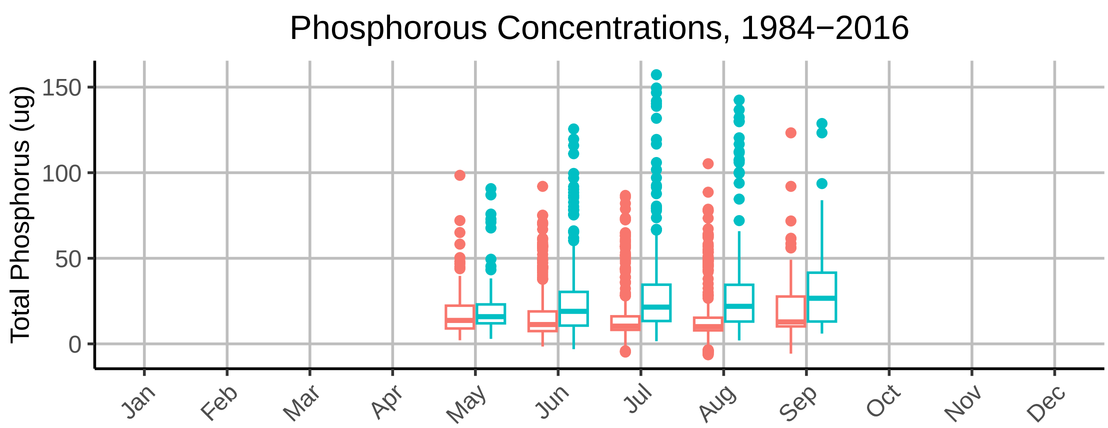

```
library(cowplot)
```

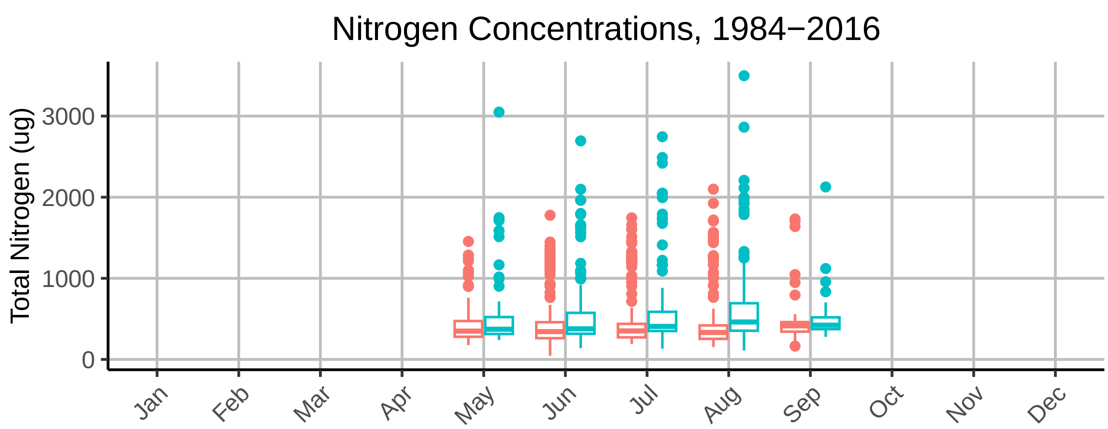
```
#combine three plots using plot_grid; using seperate code chunk to adjust knitting errors when printing

plot_grid(Nutrient_TP, Nutrient_TN, Nutrient_temp, nrow=3, align='h')
```

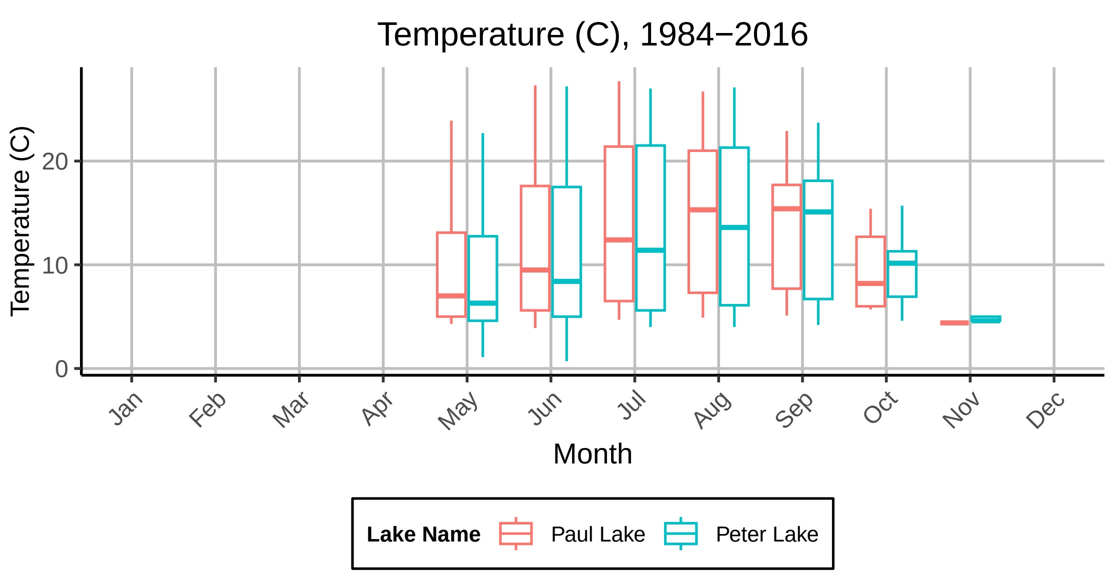
```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

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

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


Phosphorous Concentrations, 1984−2016
Total Phosphorus (ug)
0
50
100
150
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


Nitrogen Concentrations, 1984−2016
Total Nitrogen (ug)
0
1000
2000
3000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

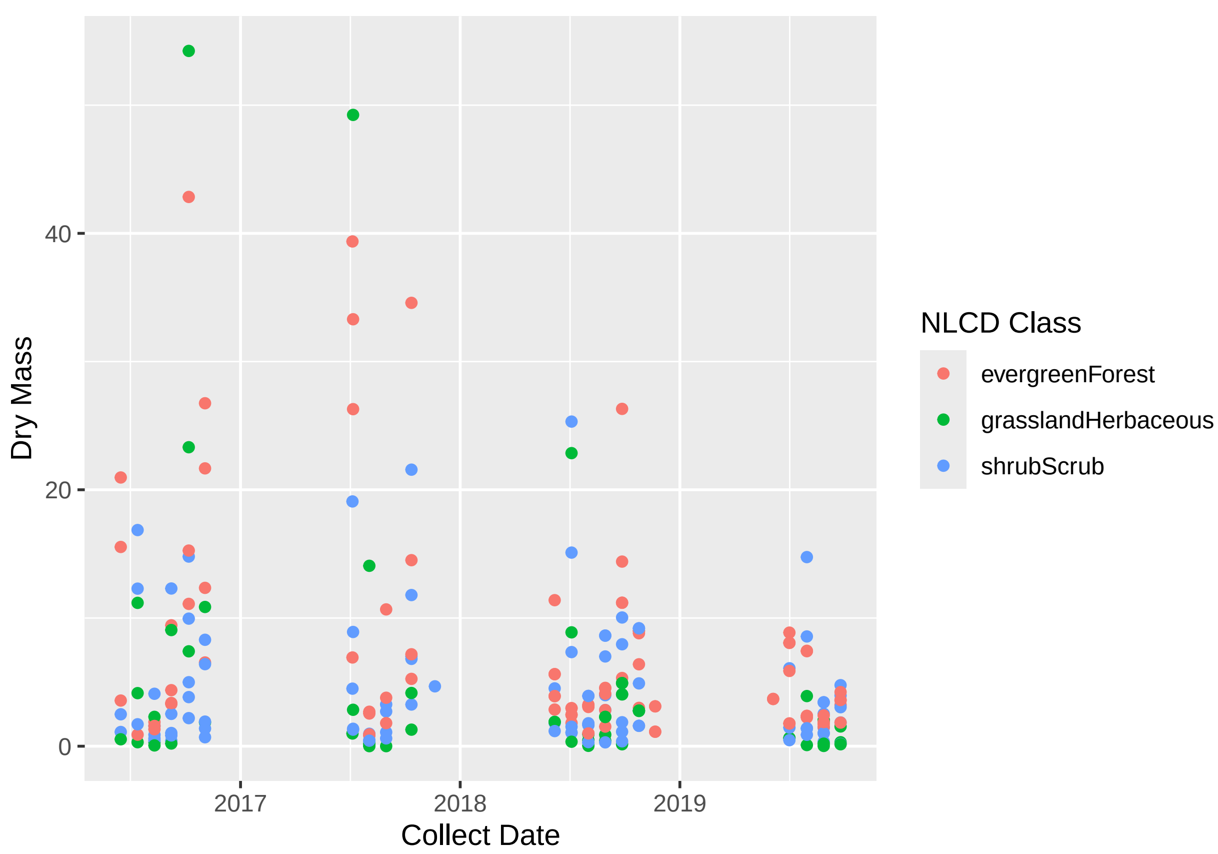

Temperature (C), 1984−2016
Temperature (C)
0
10
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
Lake Name
Paul Lake
Peter Lake

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

> Answer: The values are highest in the summer, most likely due to higher rates of production from sun and heatduring the summer seasons. The total phosphorous and nitrogen is higher in Peter Lake due to the fact that it's lareger in size.

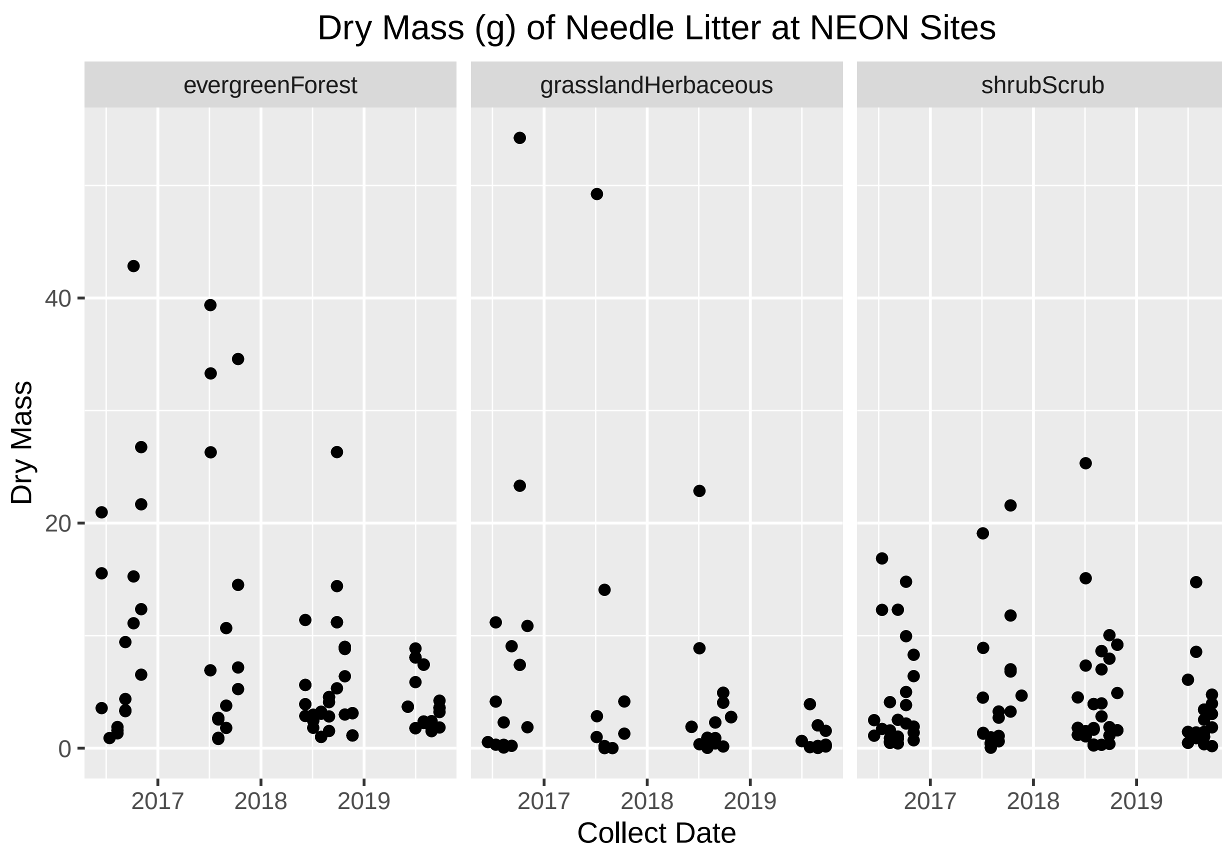
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

needles_data<- subset(Litter, functionalGroup== "Needles")

#print(needles_data)


ggplot(needles_data, aes(x =collectDate, y = dryMass, color= nlcdClass))+
  labs(x="Collect Date",   y="Dry Mass", color= "NLCD Class")+ geom_point()
```




```
#create data frame by first subsetting, then apply ggplot to subset
```

```r
#7


needles_data_faceted <-
  ggplot(needles_data, aes(x = collectDate, y =dryMass)) + geom_point() +
  facet_wrap(vars(nlcdClass), ncol = 3) +
  theme (plot.title = element_text(hjust = 0.5))+
  labs (title= "Dry Mass (g) of Needle Litter at NEON Sites",
        x= "Collect Date", y= "Dry Mass")

#use facet_wrap to combine graphs into one output, adjust axis and titles with labs function
print(needles_data_faceted)
```



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

> Answer:For Plot #7 we are able to observe the data more easily by year and not all clustered together. We are able to more easily see trends by class over time, as well as more clearly see outlier values within classes. We can also see trends in months across various years more clearly. However, With plot #6 we can more easily compare extreme values different classes of litter and compare the groups in general side by side, which is harder to do in plot #7.