Assignment 5: Data Visualization

Maeve Arthur

Spring 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. Upload 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) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version).

2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
#loading necessary packages
library(tidyverse); library(lubridate); library(here); library(cowplot)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Loading required package: timechange
##
##
## Attaching package: 'lubridate'
##
##
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
##
##
## here() starts at /home/guest/EDA-Spring2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
getwd()
```

```
## [1] "/home/guest/EDA-Spring2023"
```

```
#reading in data
PeterPaul.chem.nut <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"), string

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

#2
#changing sample date from factor to date
class(PeterPaul.chem.nut$sampledate)
```

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

```
PeterPaul.chem.nut$sampledate <- ymd(PeterPaul.chem.nut$sampledate); class(PeterPaul.chem.nut$sampledat
```

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

```
class(Litter$collectDate)
```

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

```
Litter$collectDate <- ymd(Litter$collectDate); class(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
#building my theme
my_theme <- theme(
    line = element_line(color="black"),
    plot.title = element_text(size = 12),
    legend.position = "right",
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    axis.title = element_text(size = 10)
   )

#setting my theme to apply to the whole project
theme_set(my_theme)
```

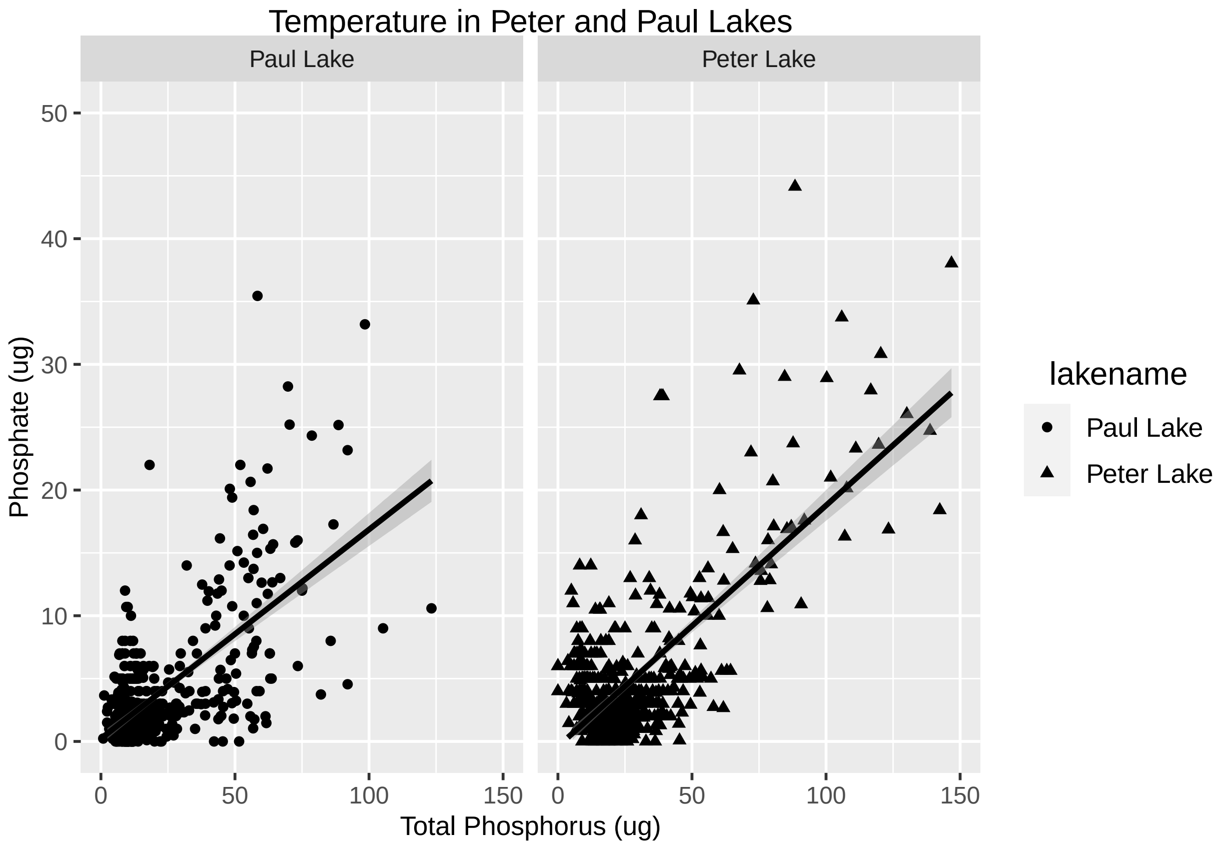
Create graphs

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 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
#creating scatterplot
tp_ug.vs.po4_2 <-
  ggplot(PeterPaul.chem.nut, aes(x = tp_ug, y = po4, shape=lakename)) +
  geom_point() +
  geom_smooth(method = lm, color = "black") +
  xlim(0, 150) +
  ylim(0, 50) +
  facet_wrap(vars(lakename)) +
  labs(
    title="Temperature in Peter and Paul Lakes",
    x = "Total Phosphorus (ug)",
    y = "Phosphate (ug)",
    color = "Lake Name")  #how to I change this legend?

print(tp_ug.vs.po4_2)
```

```
## `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()`).

## Warning: Removed 2 rows containing missing values (`geom_smooth()`).
```

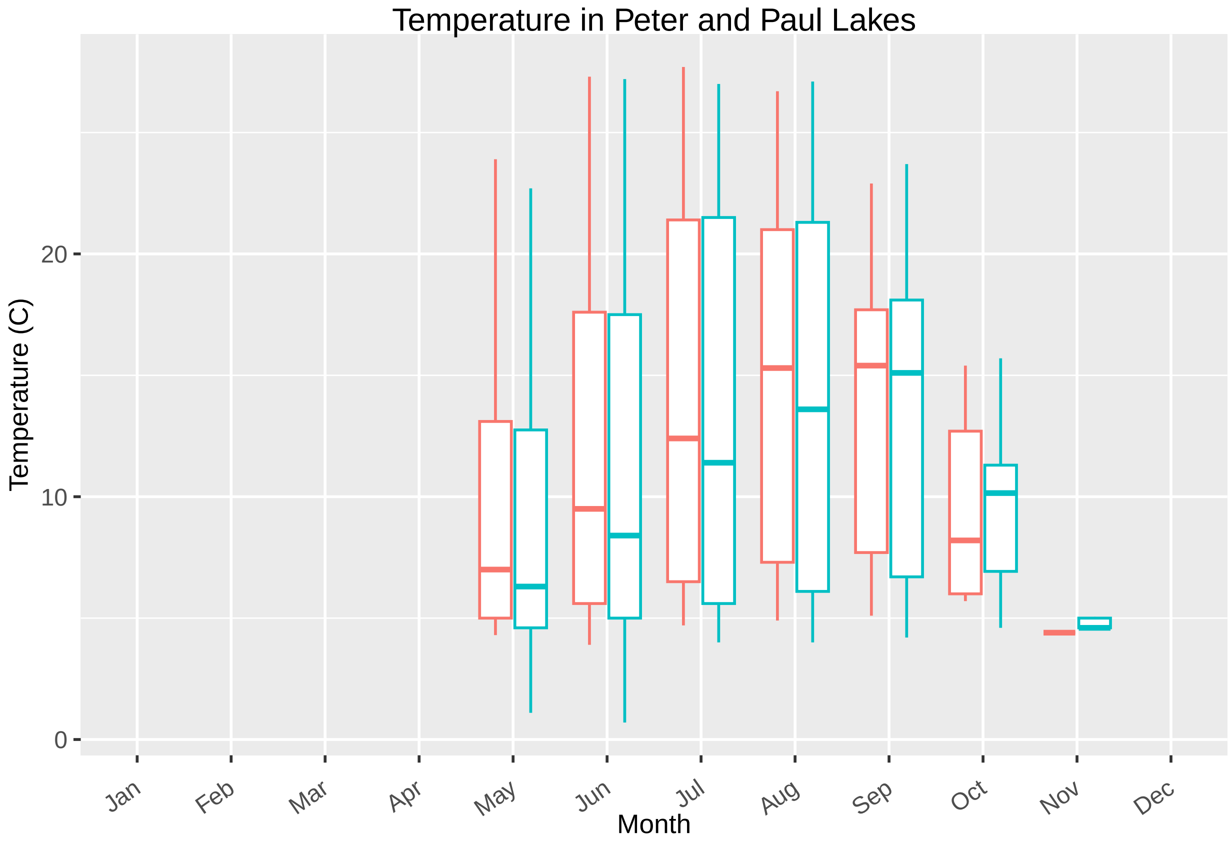


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

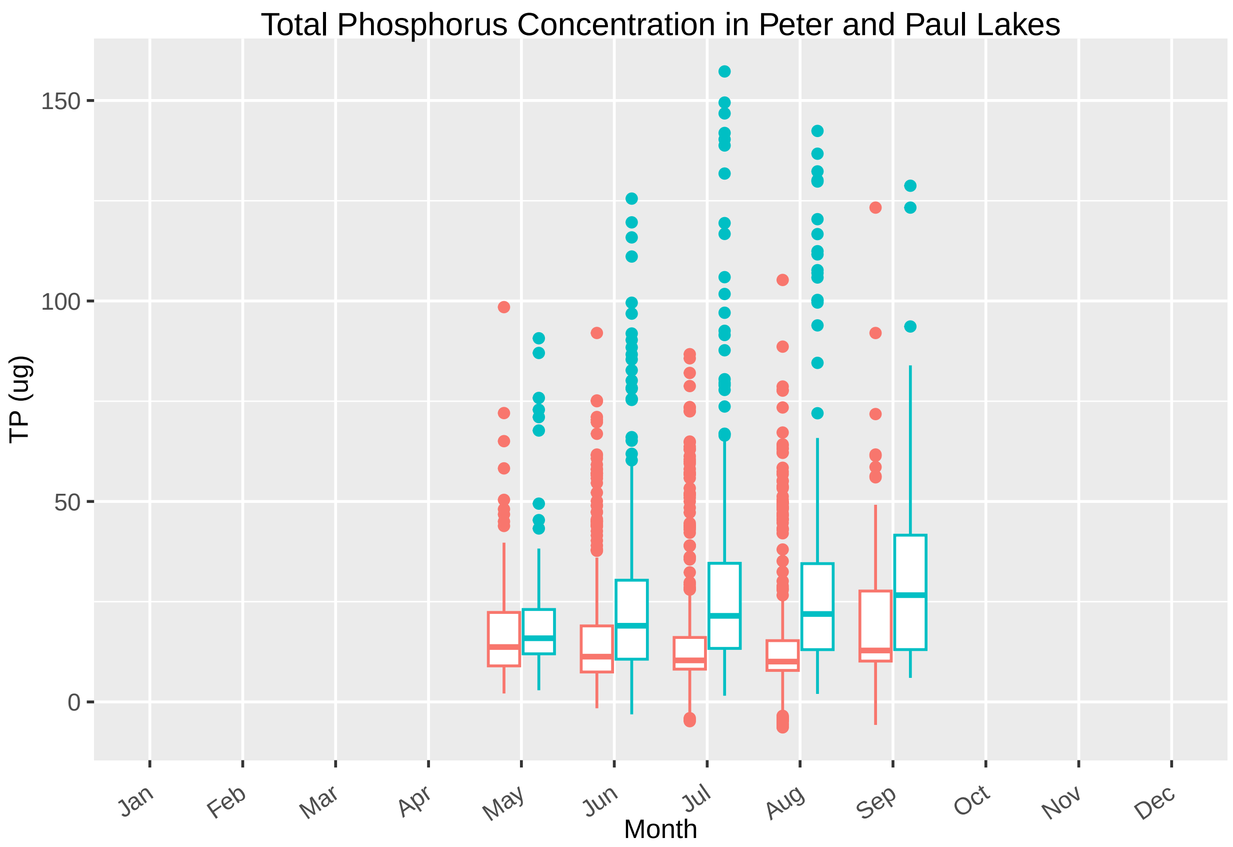
```
#5

#temperature plot
temp.plot <-
  ggplot(PeterPaul.chem.nut, aes(
    x = factor(
    month,
    levels=1:12,
    labels = month.abb),
    y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(drop=FALSE) +
  labs(
    title="Temperature in Peter and Paul Lakes",
    x = "Month",
    y = "Temperature (C)") +
  theme(axis.text.x = element_text(angle = 35,  hjust = 1),
        legend.position = "none")

print(temp.plot)
```

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



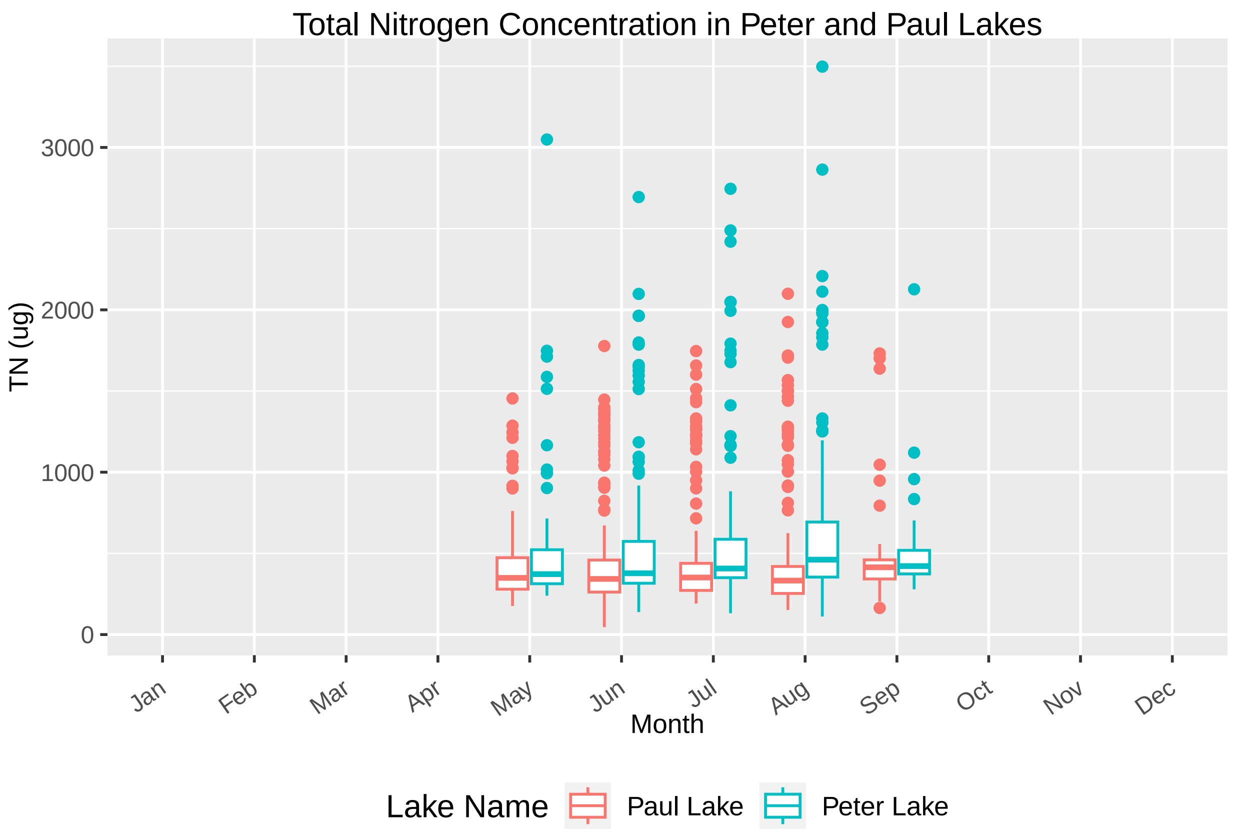
```
#TP plot
TP.plot <-
  ggplot(PeterPaul.chem.nut, aes(
    x = factor(
    month,
    levels=1:12,
    labels = month.abb),
    y = tp_ug)) +
  geom_boxplot(aes(color = lakename))+
  scale_x_discrete(drop=FALSE) +
  labs(
    title="Total Phosphorus Concentration in Peter and Paul Lakes",
    x = "Month",
    y = "TP (ug)",
    color = "Lake name") +
  theme(axis.text.x = element_text(angle = 35,  hjust = 1),
        legend.position = "none")

print(TP.plot)
```

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



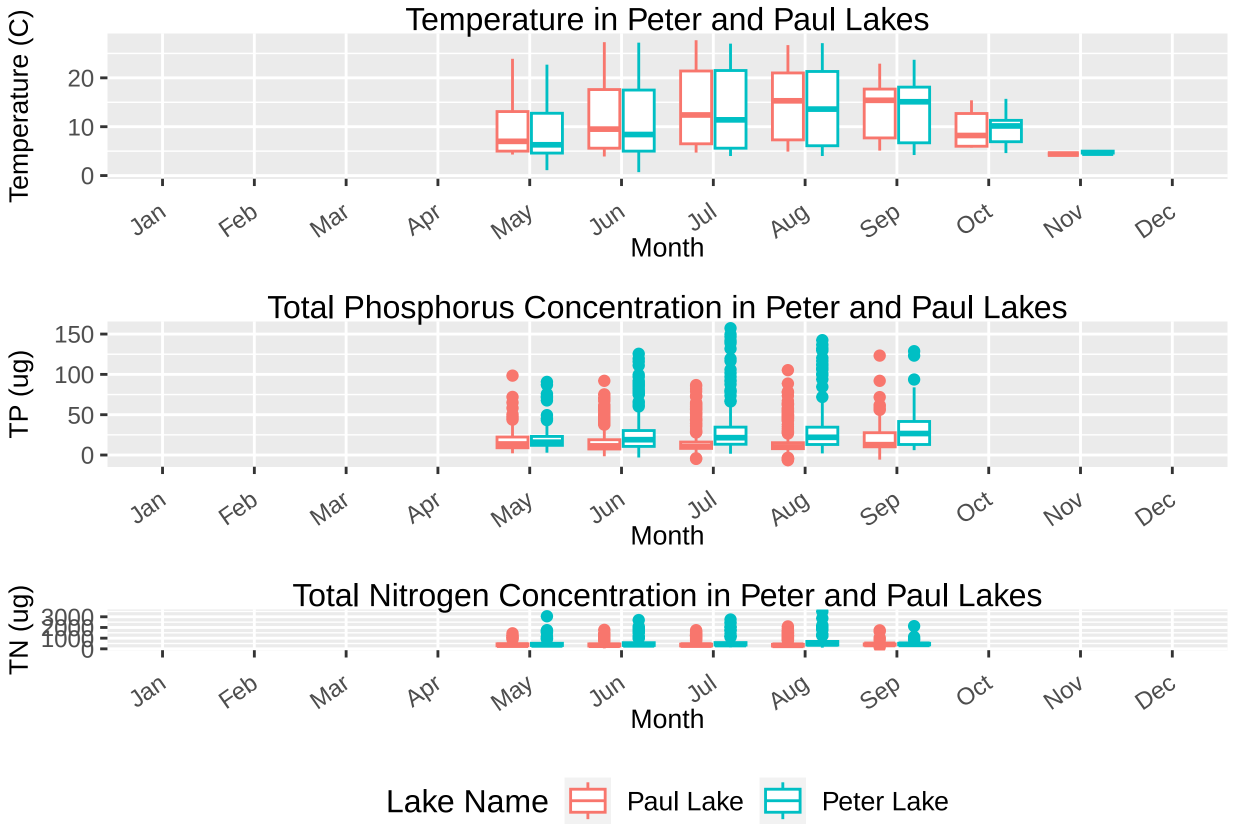
```
#TN plot
TN.plot <-
  ggplot(PeterPaul.chem.nut, aes(
    x = factor(
    month,
    levels=1:12,
    labels = month.abb),
    y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(drop=FALSE) +
  labs(
    title="Total Nitrogen Concentration in Peter and Paul Lakes",
    x = "Month",
    y = "TN (ug)",
    color = "Lake Name") +
  theme(axis.text.x = element_text(angle = 35,  hjust = 1),
        legend.position = "bottom")

print(TN.plot)
```

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



```
#making cowplot
plot_grid(temp.plot, TP.plot, TN.plot,
  align = 'v',
  axis = 'b',
  nrow = 3
  )
```

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

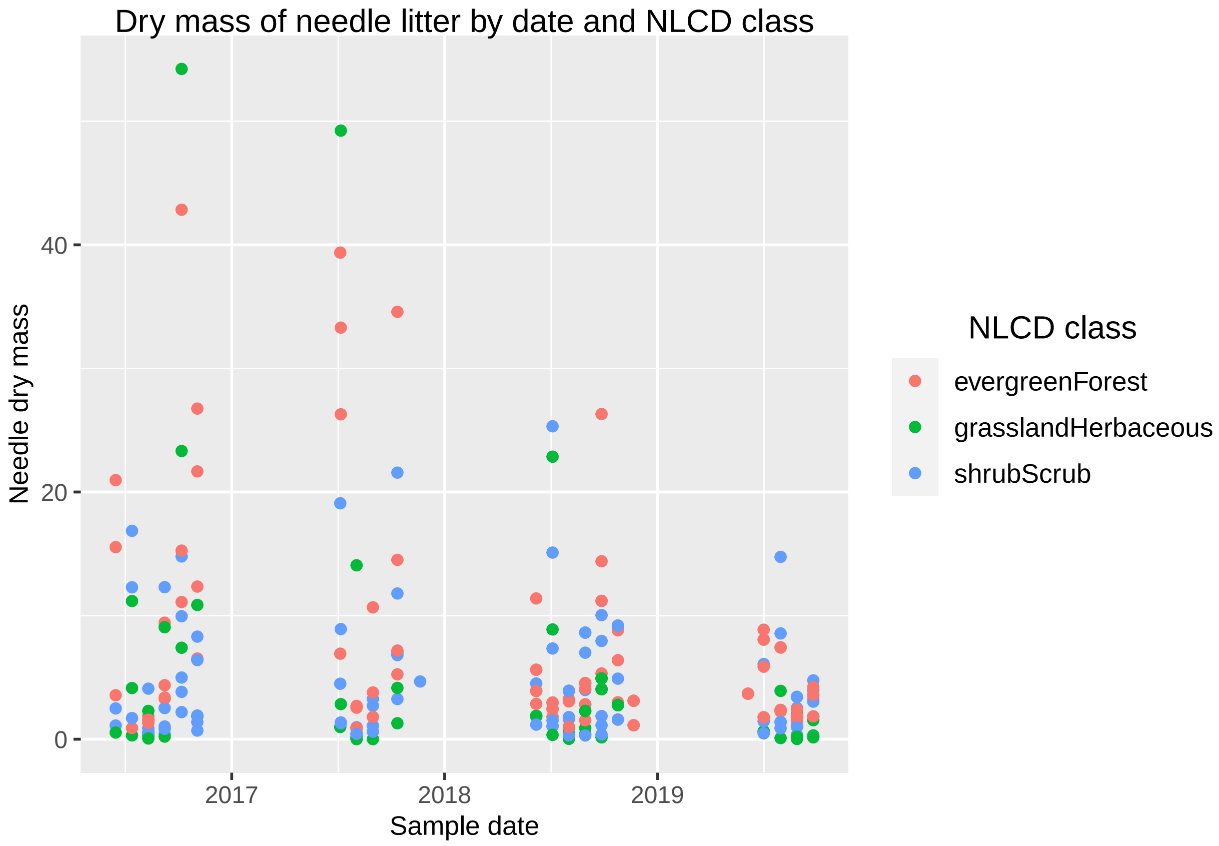


Question: What do you observe about the variables of interest over seasons and between lakes?

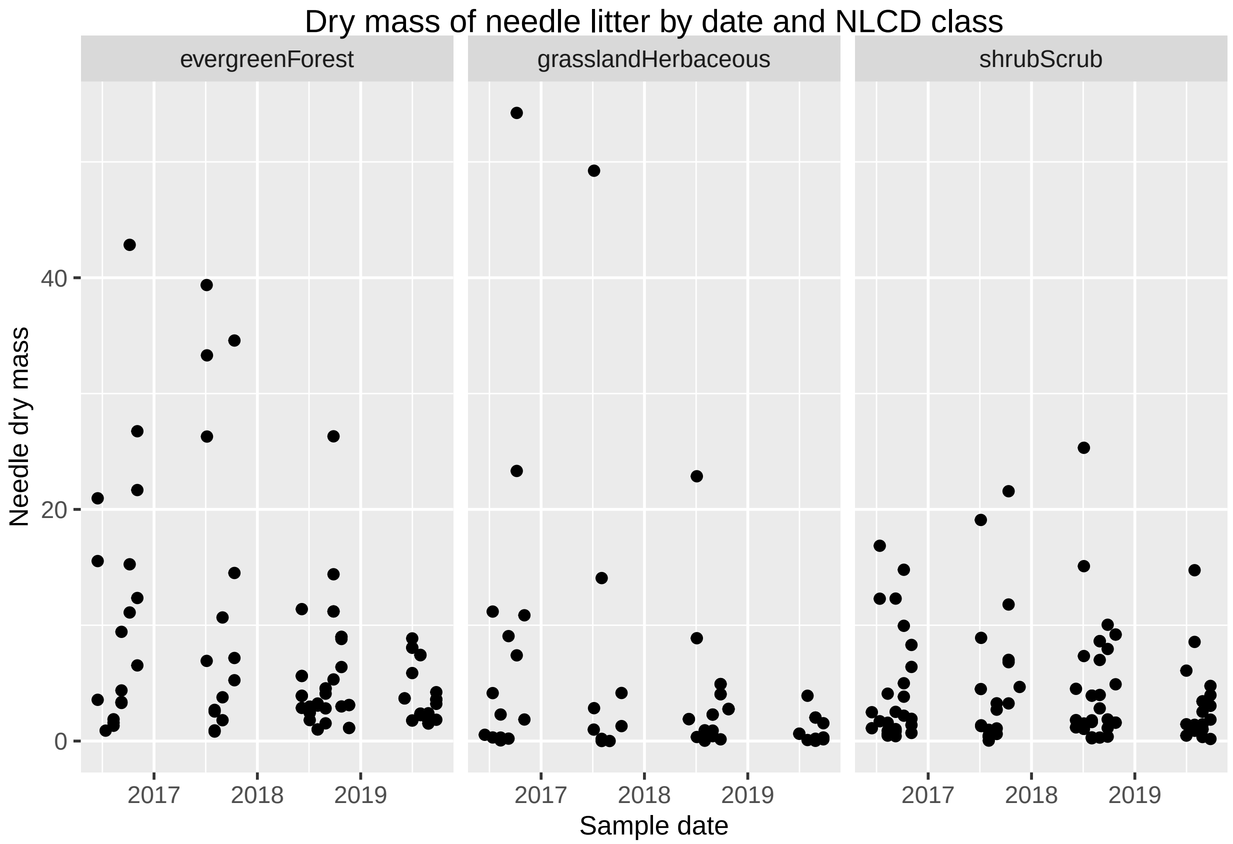
> Answer: In terms of lake temperature, the plot shows that temperature distributions between the two lakes are pretty comparable throughout the observed months. The plot also shows that the mean temperaturre in Paul Lake is consistently higher than the mean temperature in Peter Lake, except in October and November, when the mean temperature in Peter Lake is higher than that of Paul Lake. In terms of phosphorus concentration, the plot shows that both lakes are characterized by a significant amount of outlyinf data. Comparing the two lakes, you can see that the mean concentration in Peter Lake is higher than the mean concentration in Paul Lake during each of the observed months. In Peter Lake, the mean phosphorus concentration increases each month. THis is not the case for Paul Lake, where the concentration is relatively steady but does change month to month. In terms of nitrogen concentration, mean nitrogen levels between the two lakes are comparable and stay relatively consistent between the observation motnhs. In August, there is a slight spike in nitrogen levels 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 dry mass vs sample date, not faceted
litter.plot <- Litter %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(
    title="Dry mass of needle litter by date and NLCD class",
    x = "Sample date",
    y = "Needle dry mass",
    color = "NLCD class")

print(litter.plot)
```



```
#7
#needle dry mass vs data, faceted
litter.plot2 <- Litter %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(
    title="Dry mass of needle litter by date and NLCD class",
    x = "Sample date",
    y = "Needle dry mass",
    color = "NLCD class")

print(litter.plot2)
```



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

Answer: I think plot 7 is more effective. Separating the information into 3 panes makes the graph more visually understandable. When the information is in one panel, the colors of the NLCD classes make the plot visually complicated. It is easier to make comparisons of dry mass between years and between NLCD classes when the plot is split into 3 panels.