Assignment 5: Data Visualization

Darpan Barua

Spring 2025

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<DarpanBarua>_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)
library(lubridate)
library(here)
library(cowplot)

# Below reads the processed data files
ntl_lter <- read_csv(here("Processed_KEY", "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))

niwot_ridge <- read_csv(here("Processed_KEY", "NEON_NIWO_Litter_mass_trap_Processed.csv"))


# 2
```

```r
# Below converts date columns to Date format to ensure proper data handling.

ntl_lter <- ntl_lter %>%
    mutate(sampledate = ymd(sampledate))
niwot_ridge <- niwot_ridge %>%
    mutate(collectDate = ymd(collectDate))
```

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

```r
#3


#Below defines a custom ggplot theme.


library(ggplot2)

my_theme <- theme_bw() +
  theme(
    plot.background = element_rect(
      fill = 'linen',
      color = 'Black'
    ),
    legend.background = element_rect(fill = 'linen'),
    axis.line = element_line(
      linewidth = 1,
      color = 'Black'
    ),
    axis.text = element_text(
      family = 'serif'
    ),
    plot.title = element_text(

      face = 'bold',
      color = 'Black',
      family = 'serif'
    ),
    panel.background = element_rect(
      fill = "linen",
      colour = "linen",
      linewidth = 0.5,
      linetype = "solid"
    )
  )
```

```
# Below sets my_theme as the default theme for all plots
theme_set(my_theme)
```

Create graphs

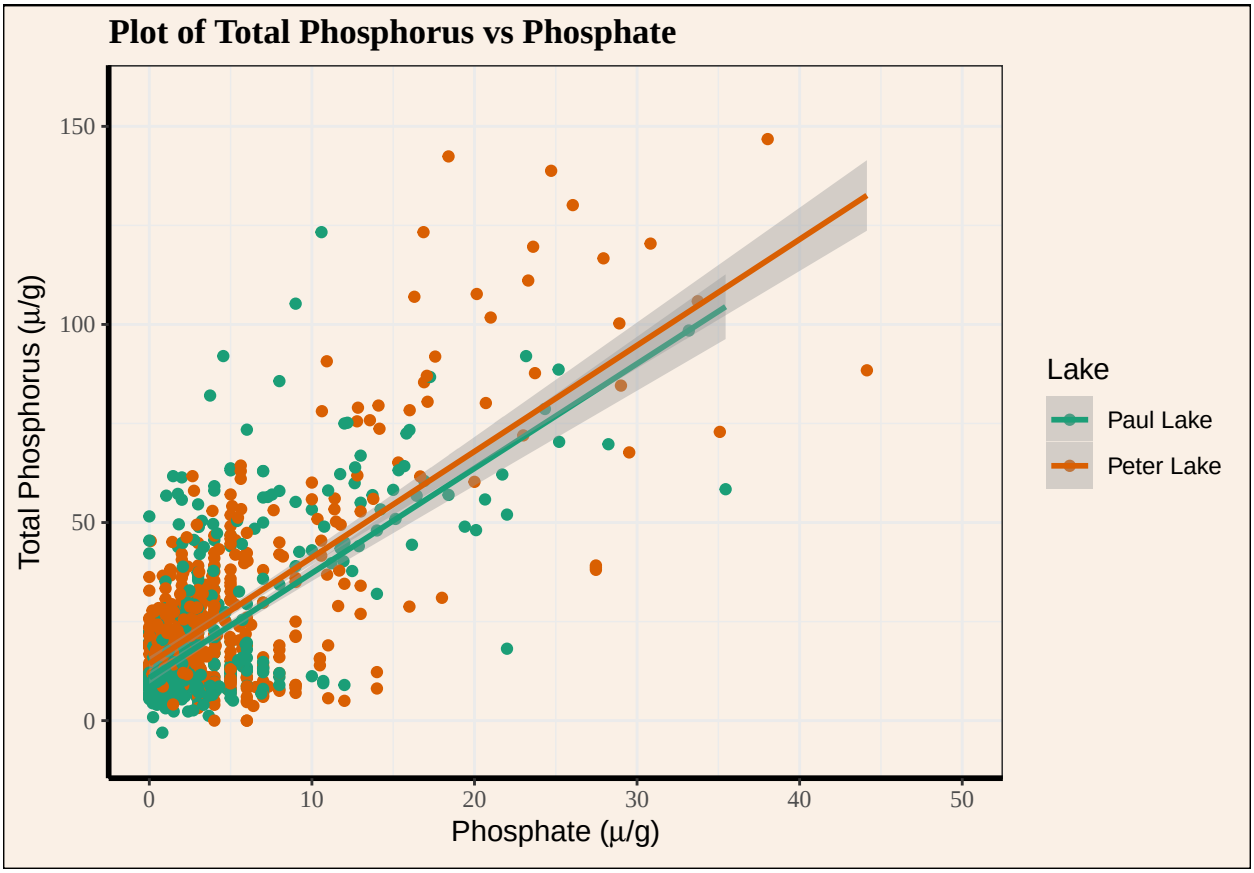
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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


#Below creates a scatter plot.

ggplot(ntl_lter,aes(x=po4,y=tp_ug,color=lakename)) +
geom_point() + #Adds scatter plot points

scale_color_brewer(palette="Dark2")+ #Uses a color palette for distinction
geom_smooth(method='lm')+ #Adds a linear trend line
xlim(0,50)+ #This limits x-axis range
labs(
x = expression(paste('Phosphate (',mu,'/g',')' )),
y = expression(paste('Total Phosphorus (',mu,'/g)')),

title = 'Plot of Total Phosphorus vs Phosphate',

color='Lake')
```

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

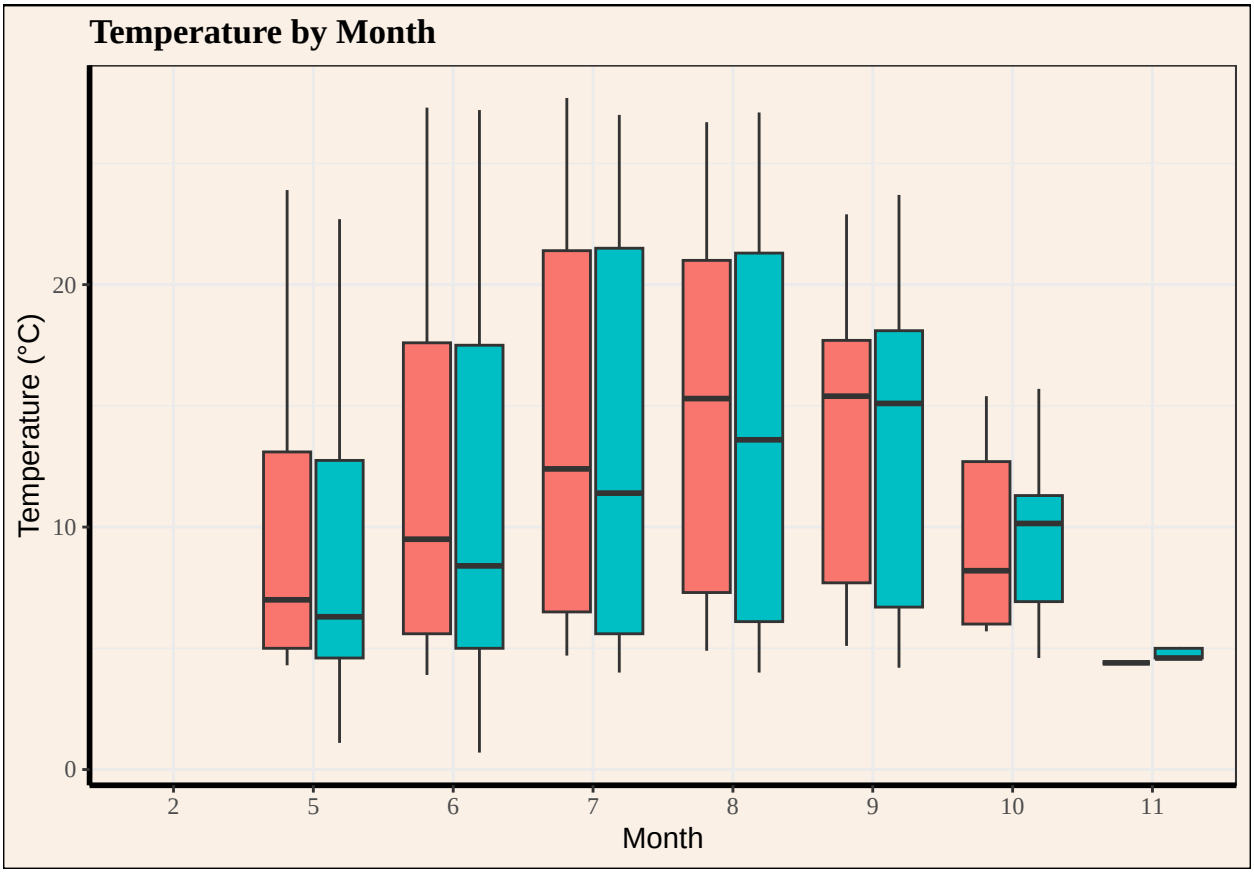


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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`.

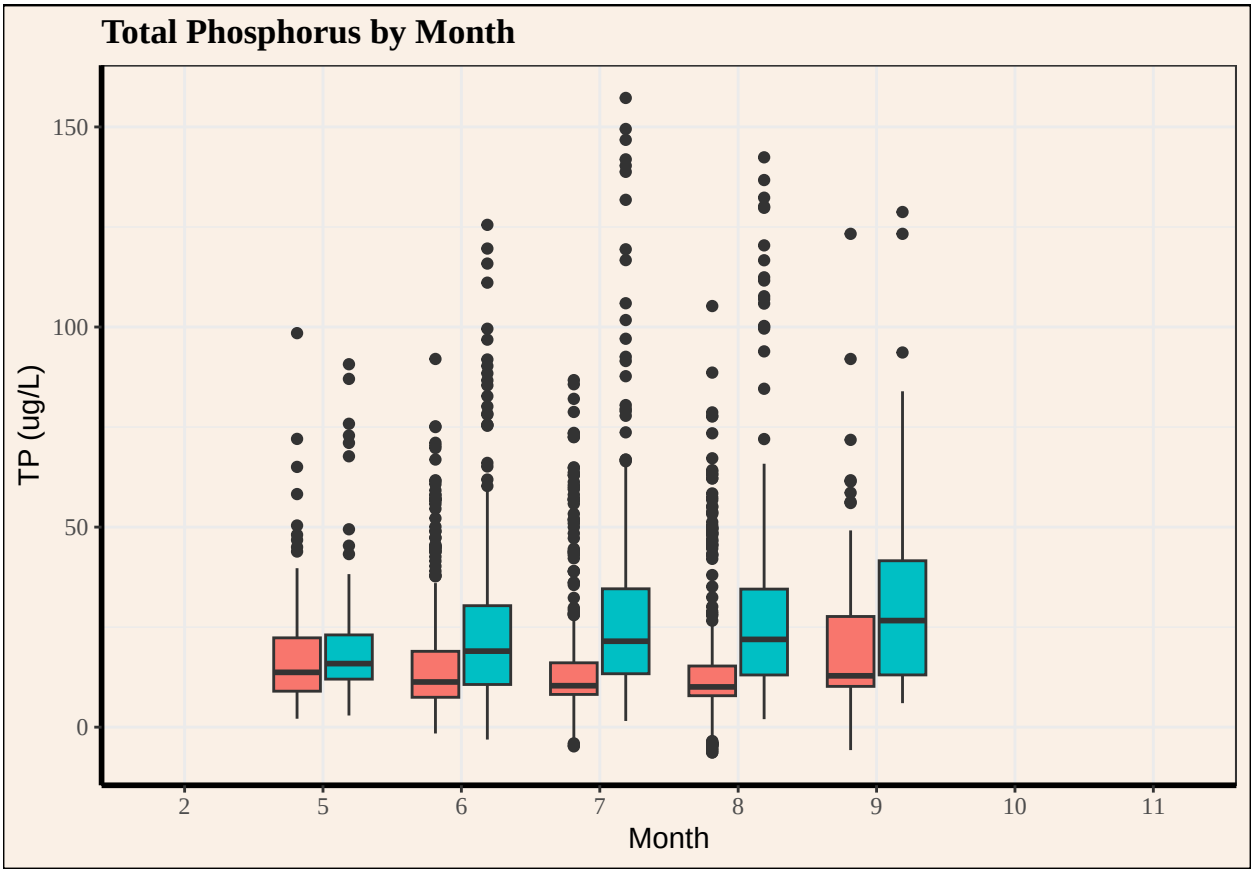
```
#5

#Below generates temperature boxplot
p1 <- ggplot(ntl_lter, aes(x = factor(month), y = temperature_C, fill = lakename)) +
  geom_boxplot() + labs(title = "Temperature by Month", x = "Month", y = "Temperature (°C)") +
  theme(legend.position = "none")

print(p1)
```

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



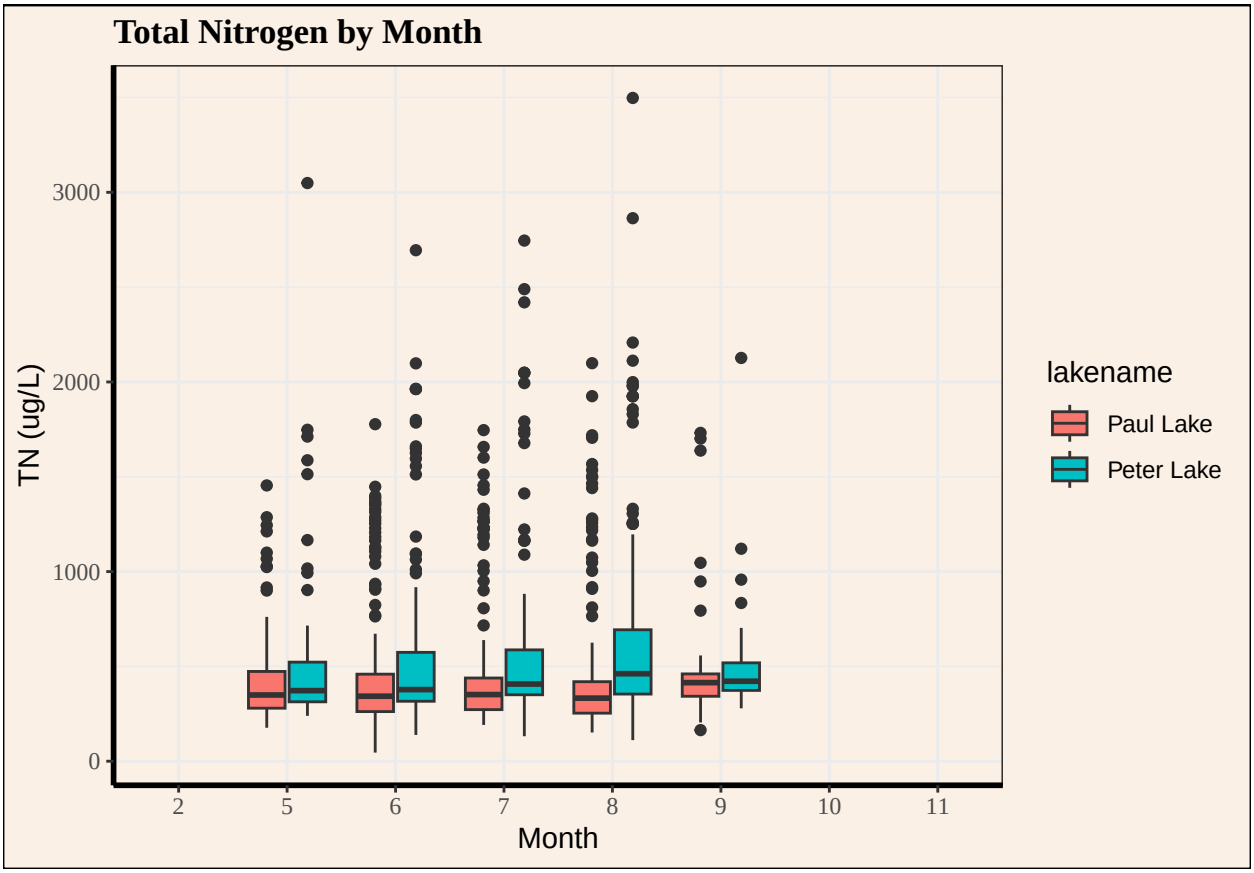

```
#Below generates Total Phosphorus boxplot
p2 <- ggplot(ntl_lter, aes(x = factor(month), y = tp_ug, fill = lakename)) +
  geom_boxplot() + labs(title = "Total Phosphorus by Month", x = "Month", y = "TP (ug/L)") +

  theme(legend.position = "none")

print(p2)
```

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




```
#Below generates Total Nitrogen boxplot

p3 <- ggplot(ntl_lter, aes(x = factor(month), y = tn_ug, fill = lakename)) +
  geom_boxplot() + labs(title = "Total Nitrogen by Month", x = "Month", y = "TN (ug/L)")

print(p3)
```

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

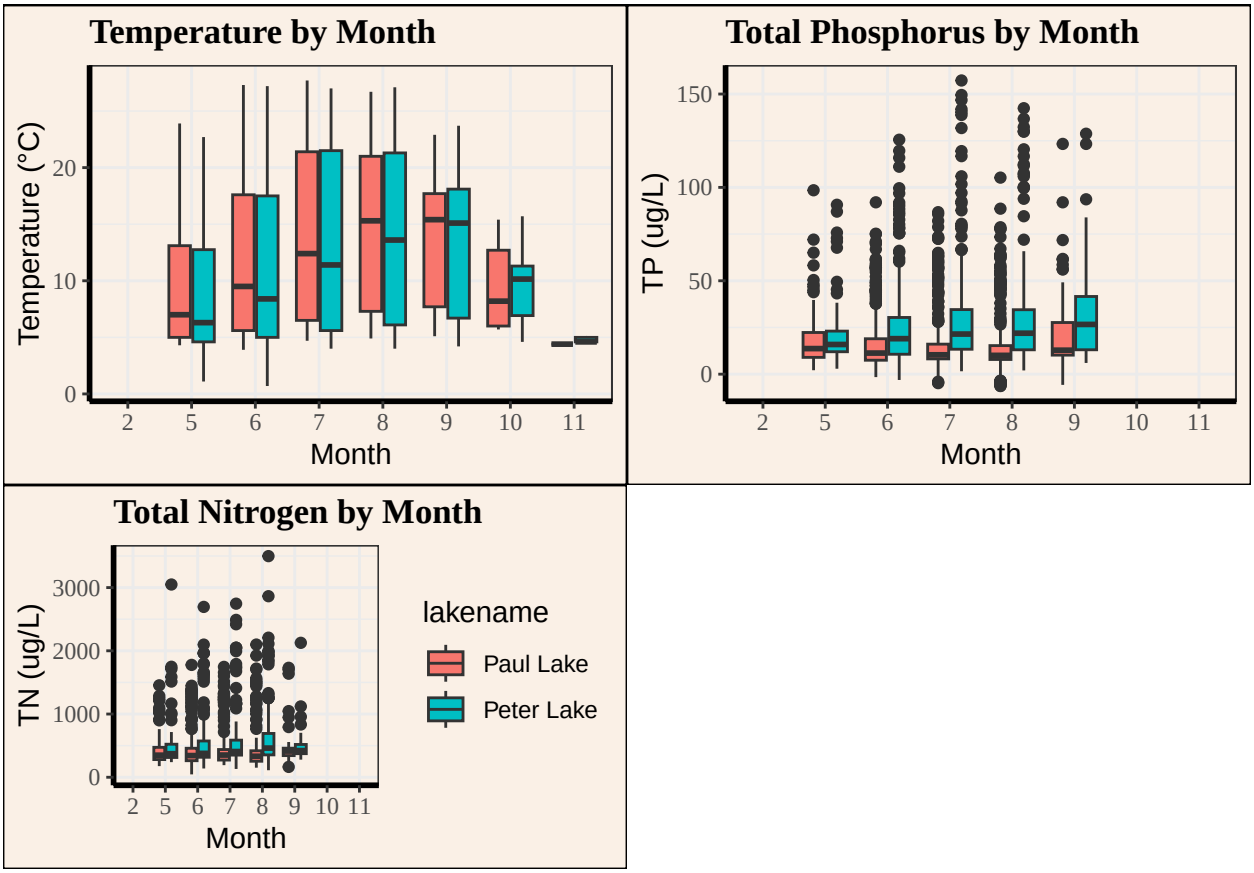



```
#Below arranges 3 plots into 1 figure using cowplot
plot_grid(p1, p2, p3, nrow = 2, align = 'h', rel_heights = c(1.25, 1))
```

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

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



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

Answer:

(a) For temperature trends over the moths - the seasonality aspect is clear. It rises from May to July with peaks in July and August, before gradually decreasing from September onwards. The temperature range is largest in summer (June-August). Both lakes show similar temperature distributions with no drastic differences. Any difference could be attributed to differences in the lakes' physical features and environment!

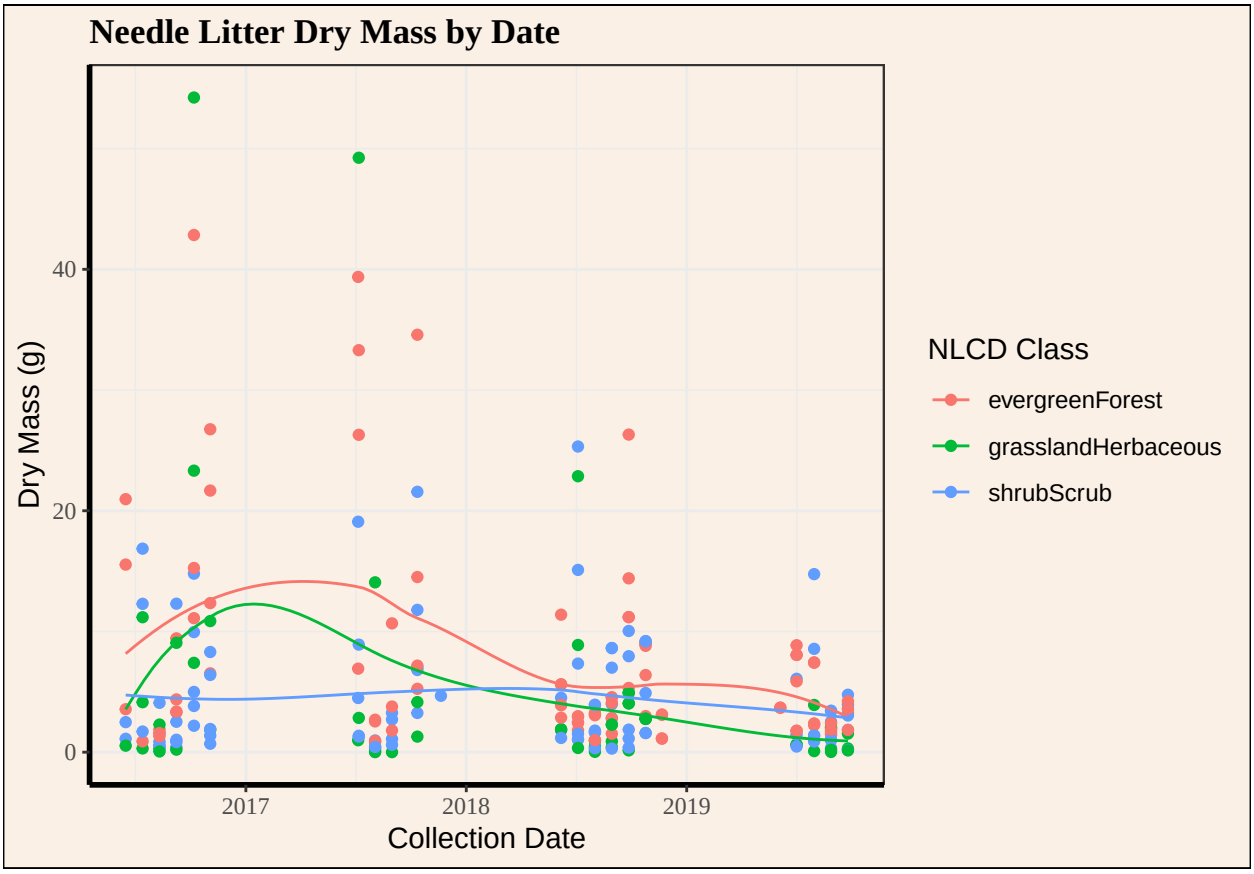
(b) Total Phosphorus levels show high variability, especially in the summer months, with a noticeable peak in July. There are more outliers in July and August (maybe there are phosophorus level hikes periodically?). Peter Lake (blue) generally has higher TP values than Paul Lake (red), particularly in summer months. Paul Lake shows lower variability in TP, while Peter Lake has a wider range with frequent high-value outliers.

(c) Total Nitgrogen levels also show high variability with highest values in summer (June-August). August shows the widest range and the most extreme outliers (some natural event leading to higher nutrient contents?). Between lakes Peter Lake (blue) seems to have slightly higher TN values than Paul Lake (red). Poth have similar median TN values, but Peter Lake has a higher spread and more outliers. Extreme outliers however, are present in both lakes, particularly in summer.

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)

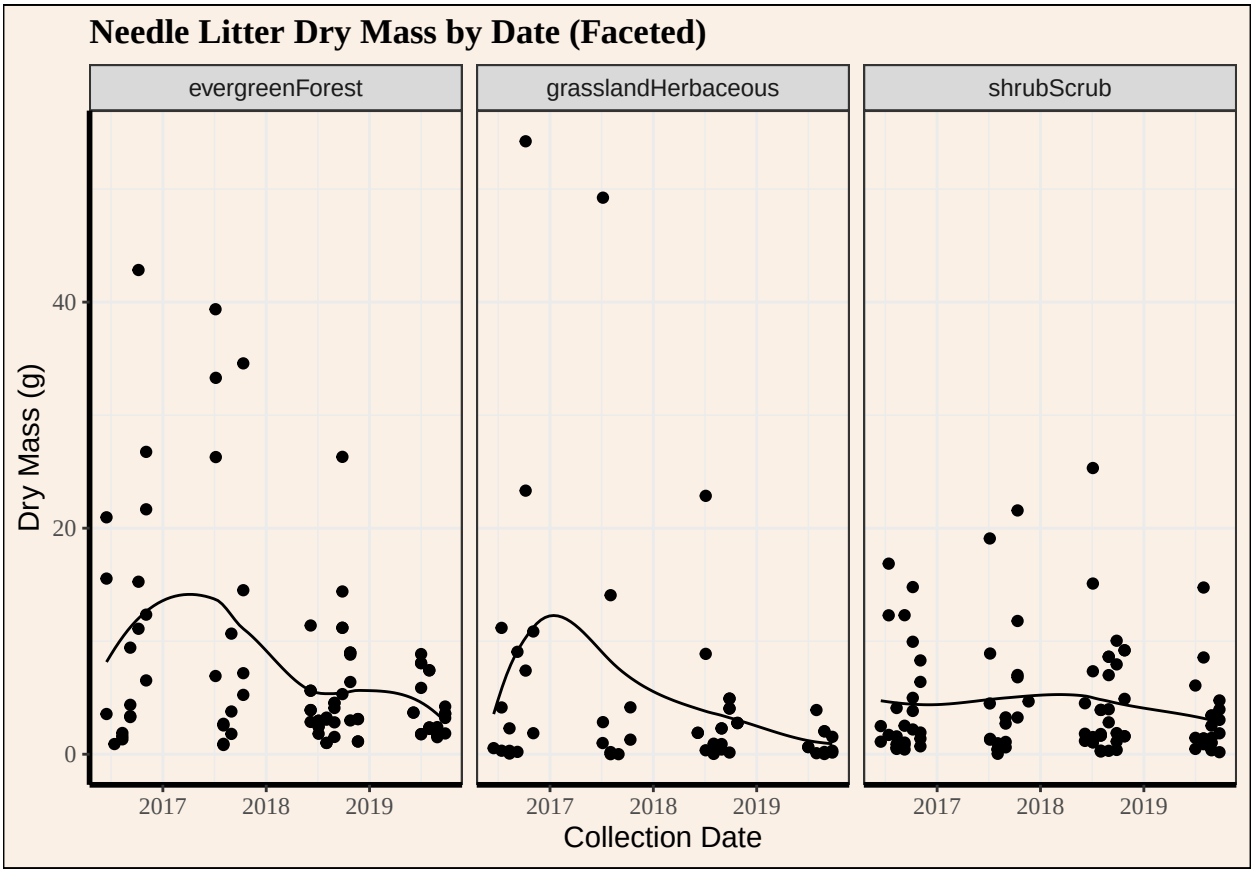
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

#Below filters only the "Needles" functional group from the litter dataset

needles_data <- niwot_ridge %>%
  filter(functionalGroup == "Needles")

#Below creates a plot where dry mass of a needle litter is plotted over time, categorized by NLCD class

ggplot(needles_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() + #adds scatter plot points
  geom_line(stat="smooth", se=FALSE) + #adds smooth trend lines
  labs(title = "Needle Litter Dry Mass by Date",
       x = "Collection Date", y = "Dry Mass (g)",
       color = "NLCD Class")
```




```
#7

#Below creates a similar plot but splits data into separate panels for each NLCD class

ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  geom_line(stat="smooth", se=FALSE) +
```

```
  facet_wrap(~ nlcdClass) + #separates plots by NLCD class
  labs(title = "Needle Litter Dry Mass by Date (Faceted)",
       x = "Collection Date", y = "Dry Mass (g)")
```



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

Answer:Effectiveness depends on purpose. If the goal is to compare trends across NCLD classes, given the color coded features, Plot 6 is better. If the goal is to analyze trends within each class separately, Plot 7 is better. In terms of size, if the data set is small - Plot 6 could be more effective; as with larger and overlapping data, Plot 7 can provide a clearer view.