

# Assignment 5: Data Visualization

Sarah Kear

Spring 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    3.4.4      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 /Users/sarah/Documents/872_EDA/EDA_Spring2024
```

```
library(cowplot)
```

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

```
library(ggplot2)
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
#2
processed_data = "Data/Processed_KEY"
PeterPaul.chem.nutrients <- read.csv(
  here(
    processed_data,
    "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE
)

Niwot.Ridge.Litter <- read.csv(
  here(
    processed_data, "NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE
)
```

## 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
A05_theme <- theme_base() + theme(
  plot.title = element_text( #Updating Title
    color= 'darkslategray',
    size = 14
  ),
  axis.text = element_text( #Updating axis text
    color = "black",
    size = 10
  ),
  axis.title.x = element_text( #Updating x-axis
    color = "black",
    size = 12
  ),
  axis.title.y = element_text( #Updating y-axis
    color = "black",
    size = 12
  ),
  legend.position = "top", #Putting legend to top
  legend.title = element_text( #Updating legend title
    color='black',
    size = 12
  ),
  legend.text = element_text( #Updating legend text
    size = 12
  ),
  plot.background = element_blank() #removing plot edge/background
)

```

## 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
plot4 <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
    x=po4,
    y=tp_ug,
    color=lakename)
  ) +
  geom_point(alpha=0.70) +
  xlim(0, 10) + #moved max to 10 since the majority of po4 was in the 0 to 10 range
  ylim(0, 75) +
  ylab("Total Phosphorus") +
  xlab("Total Phosphate") +
  ggtitle("Phosphorus by Phosphate at Paul Lake and Peter Lake") +
  scale_color_manual(values = c("darkorange2", "cornflowerblue"),

```

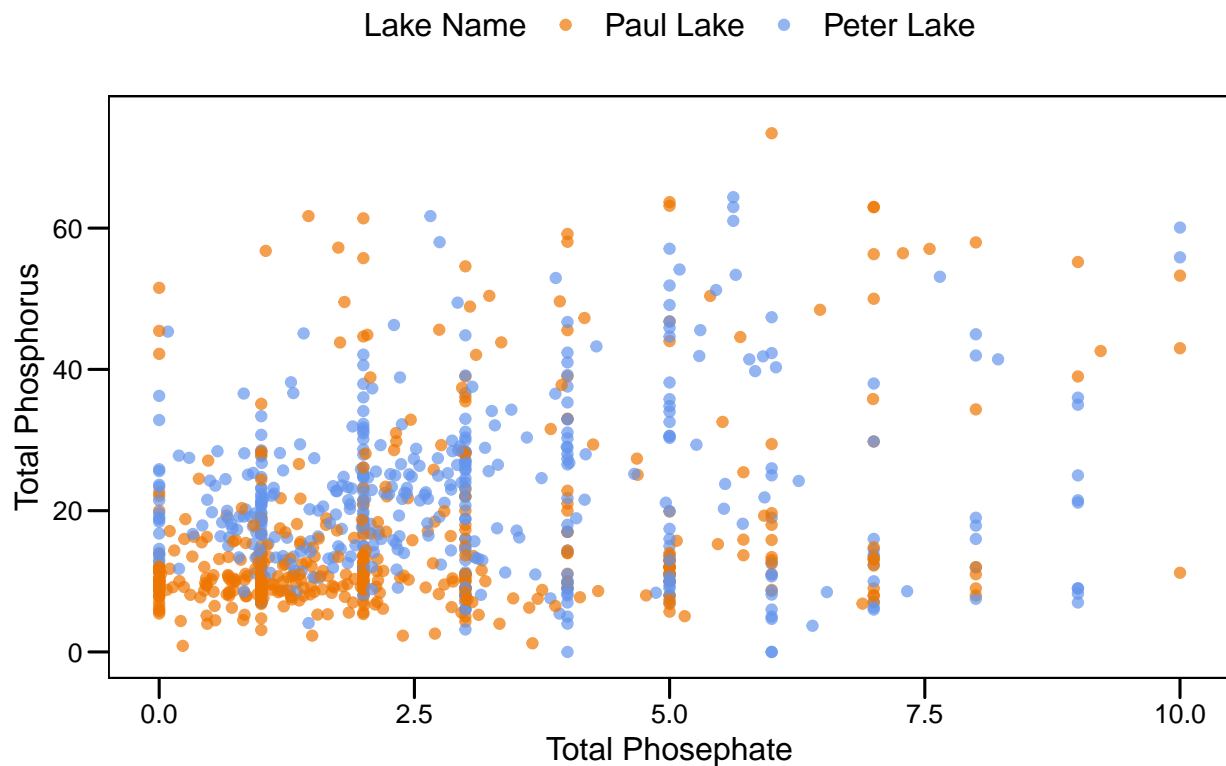
```

    name = "Lake Name") +
  A05_theme
print(plot4)

```

## Warning: Removed 22064 rows containing missing values (‘geom\_point()’).

## Phosphorus by Phosphate at Paul Lake and Peter Lake



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
# created new theme for question 5.
A05_theme1 <- theme_base() + theme(
  plot.title = element_text(
    color= 'darkslategray',
    size = 12 #decreased font size
  ),
  axis.text = element_text(
    color = "black",

```

```

    size = 8 #decreased font size
  ),
  axis.title.x = element_blank(), #leaving x-axis blank
  axis.title.y = element_text(
    color = "black",
    size = 10 #decreased font size
  ),
  legend.title = element_text(
    color='black',
    size = 8 #decreased font size
  ),
  legend.text = element_text(
    size = 8 #decreased font size
  ),
  axis.text.x = element_text(
    angle = 90, # turned x axis text sideways
    vjust = 0.5,
    hjust=1),
  legend.position = "top",
  plot.background = element_blank()
)

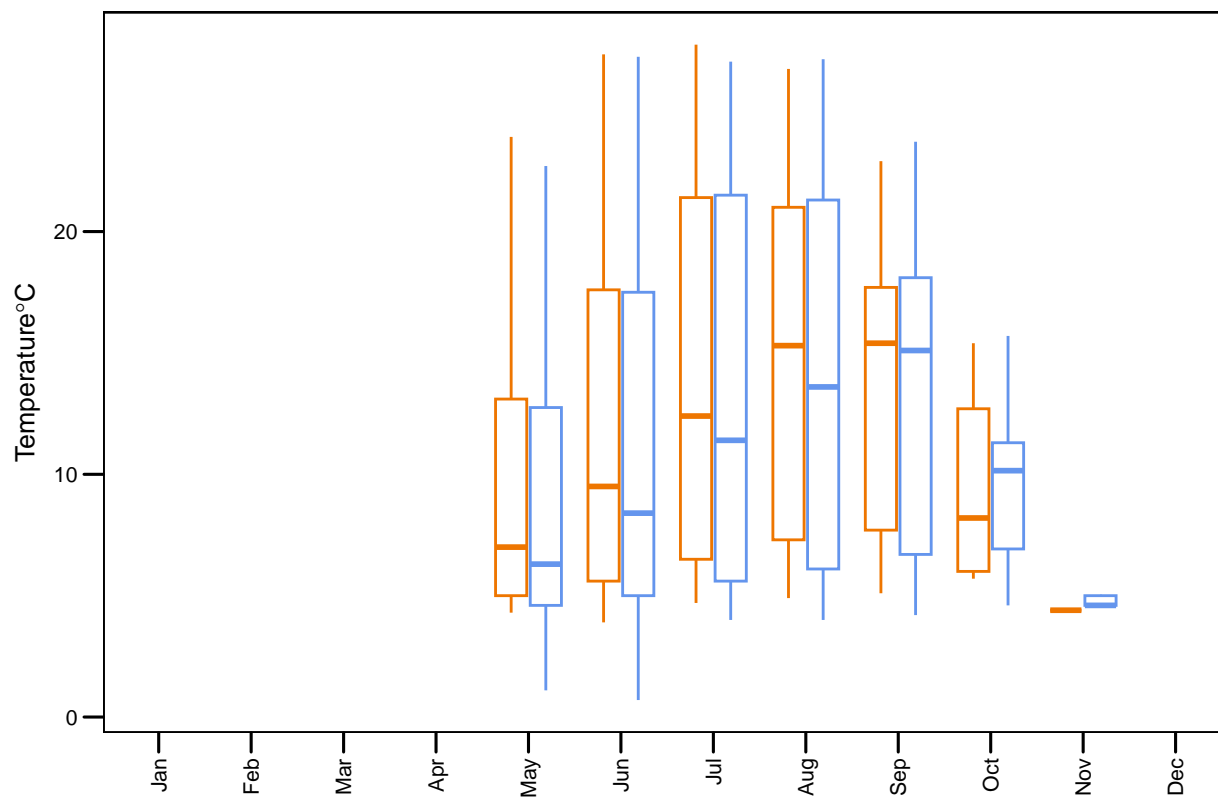
# changed month column to a factor
PeterPaul.chem.nutrients$month_f <- factor(
  PeterPaul.chem.nutrients$month,
  levels=1:12,
  labels = month.abb
)

#1st plot
temp <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
    x=month_f,
    y=temperature_C,
    color=lakename)
  ) +
  geom_boxplot() +
  ylab(expression(paste("Temperature", degree, "C"))) +
  ggtitle("Temperature, Phosphorus, & Nitrogen by Month") +
  scale_color_manual(values = c("darkorange2", "cornflowerblue"),
    name = "Lake Name") +
  A05_theme1 +
  theme(legend.position = "NONE") +
  scale_x_discrete(drop=FALSE) # so all months show
print(temp)

```

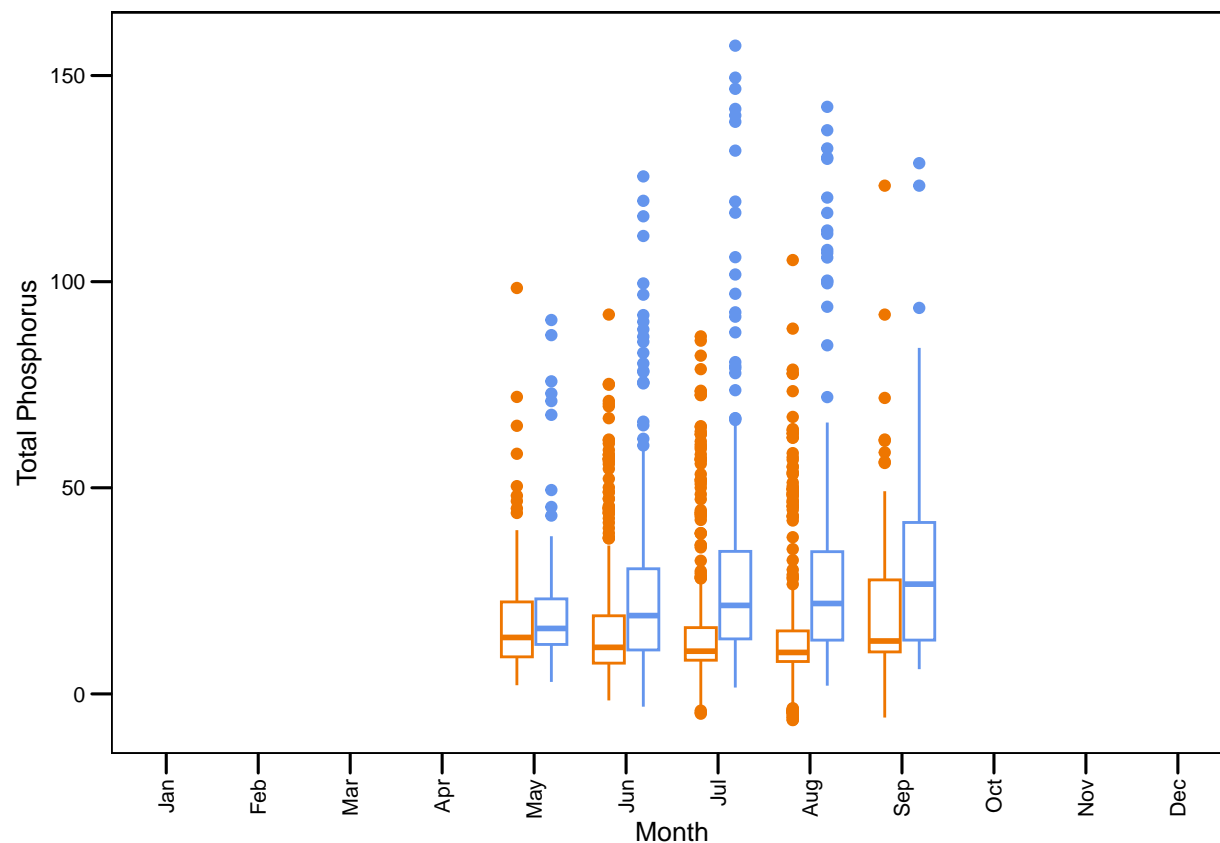
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```

## Temperature, Phosphorus, & Nitrogen by Month



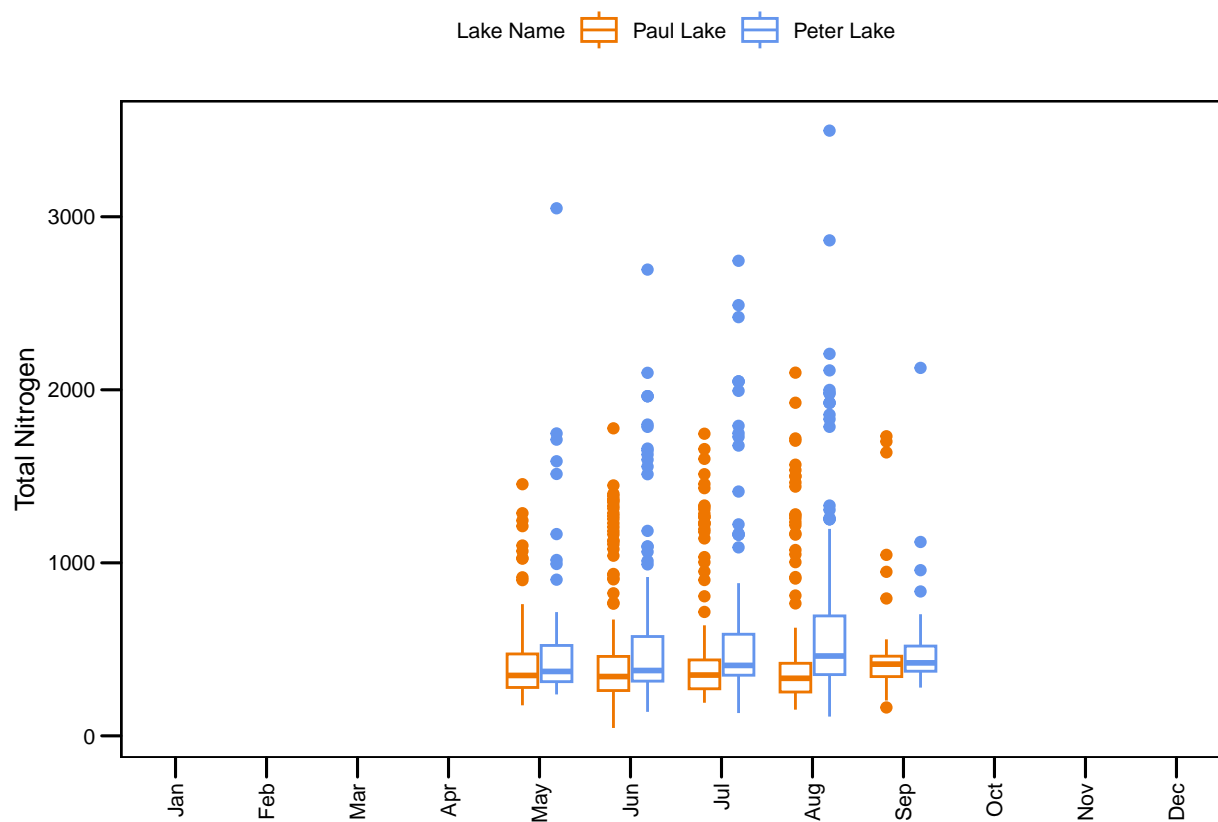
```
#2nd plot
tp <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
    x=month_f,
    y=tp_ug,
    color=lakename)
  ) +
  geom_boxplot() +
  ylab("Total Phosphorus") +
  xlab("Month") +
  scale_color_manual(values = c("darkorange2", "cornflowerblue"),
    name = "Lake Name") +
  A05_theme1 +
  theme(legend.position = "NONE",
    axis.title.x = element_text(
      color = "black",
      size = 10)) +
  scale_x_discrete(drop=FALSE)
print(tp)
```

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



```
# 3rd plot
tn <- PeterPaul.chem.nutrients %>%
  ggplot(aes(
    x=month_f,
    y=tn_ug,
    color=lakename)
  ) +
  geom_boxplot() +
  ylab("Total Nitrogen") +
  scale_color_manual(values = c("darkorange2", "cornflowerblue"),
    name = "Lake Name") +
  A05_theme1 +
  scale_x_discrete(drop=FALSE)
print(tn)
```

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



```
#combining the 3 plots together
plot5 <- plot_grid(temp,
  tp,
  tn,
  nrow = 1,
  align = "h",
  axis = "tblr", # so axis for all is the same
  rel_widths = c(1.5, 1.5, 1.5)
)
```

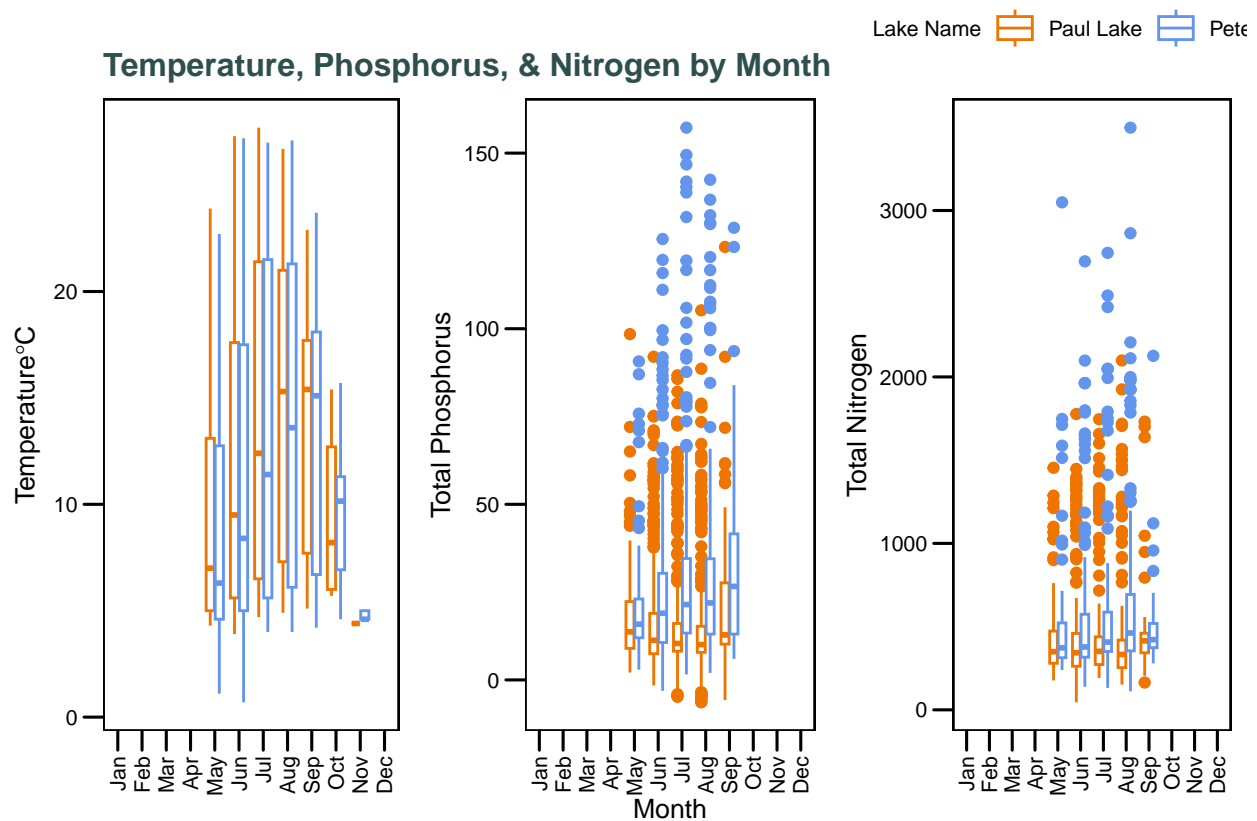
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot()').
```

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

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



```
print(plot5)
```



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

Answer: The temperature between both lakes are fairly similar throughout the months with collected data. Peter Lake seems to have more phosphorus and nitrogen than Paul Lake. However, during June through September, the median of phosphorus in Peter Lake is increasing, but median phosphorus slowly decreases in Paul Lake. During that same time period, Peter Lake has greater range (min, max) of nitrogen than Paul Lake while Paul Lake's range continues to shrink. Yet, Paul Lake seems to have more outliers than Peter Lake when it comes to nitrogen amounts. This may skew its box and whisker plots.

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.

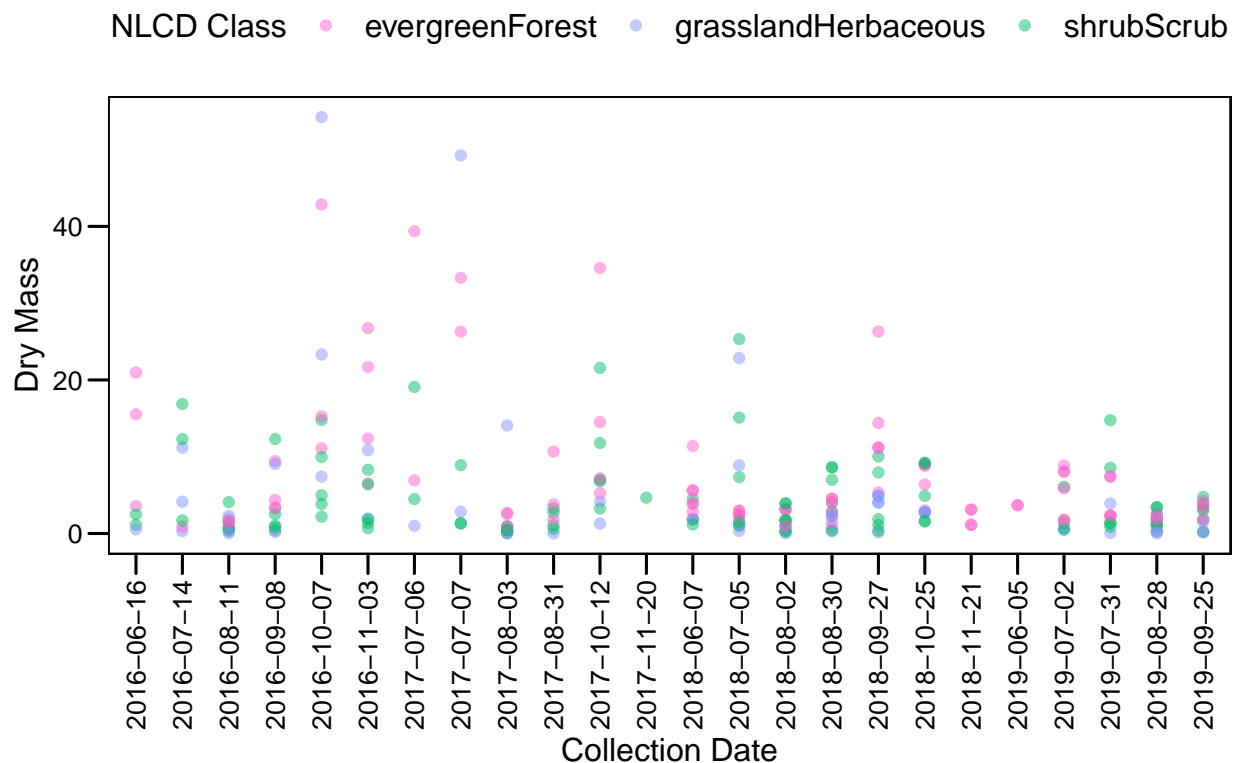
```
#6
plot6 <- Niwot.Ridge.Litter %>%
  filter(functionalGroup == "Needles") %>% #filter for just needles
  ggplot(aes(
    x=collectDate,
    y=dryMass,
```

```

    color=nlcdClass)
  ) +
  # so that data bunching can be better visualized
  geom_point(alpha=0.5) +
  ylab("Dry Mass") +
  xlab("Collection Date") +
  ggtitle("Needle Dry Mass by Collection Date") +
  scale_color_manual(values = c("#ff61cc", "#8494ff", "#00be67"),
    name = "NLCD Class") +
  A05_theme +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
print(plot6)

```

## Needle Dry Mass by Collection Date

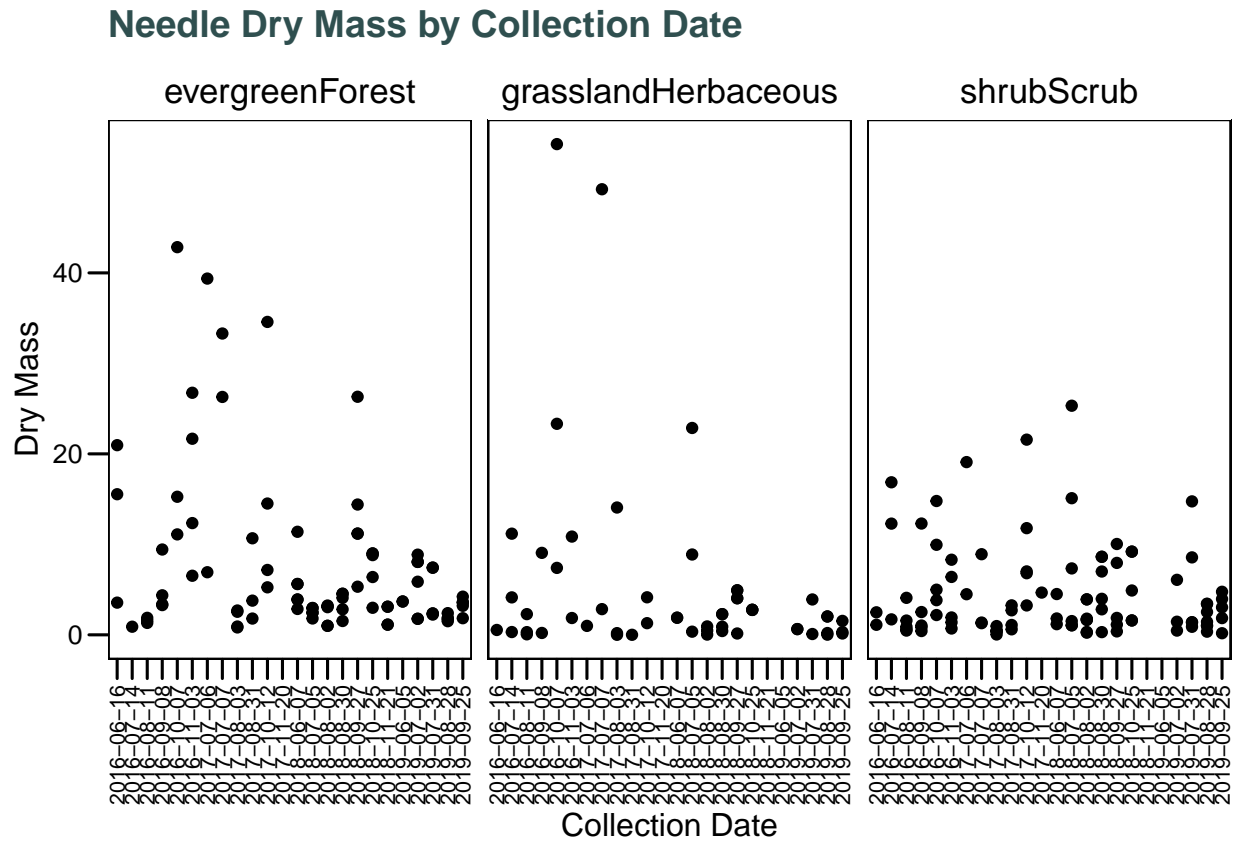


```

#7
plot7 <- Niwot.Ridge.Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(
    x=collectDate,
    y=dryMass)) +
  geom_point() +
  ylab("Dry Mass") +
  xlab("Collection Date") +
  ggtitle("Needle Dry Mass by Collection Date") +
  facet_wrap(vars(nlcdClass)) +
  A05_theme +

```

```
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1, size = 8))
print(plot7)
```



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

Answer: I think plot 6 is more effective. Although there is bunching in plot 6 that can make it hard to read, i found that trying to compare needle dry mass in plot 7 was actually harder. This was due to having so many collection dates for each NLCD class. Plot 7 is more effective in comparing needle dry mass within one NLCD class, whereas plot 6 is better at comparing the classes together. With Plot 6, i found it easier to find trends within needle dry massa cross the NLCD classes, as well as outliers throughout the collection dates.