

Assignment 5: Data Visualization

Hannah Smith

OVERVIEW

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

Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., “Salk_A05_DataVisualization.Rmd”) prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and cowplot packages. Upload the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (tidy and gathered) and the processed data file for the Niwot Ridge litter dataset.
2. Make sure R is reading dates as date format; if not change the format to date.

```
#1
getwd()

## [1] "/Users/hannahsmith/Documents/ENV872/Environmental_Data_Analytics_2020"

library(tidyverse)
#install.packages("cowplot")
library(cowplot)
library(viridis)
#install.packages("viridis")
library(RColorBrewer)
library(colormap)
PeterPaul.nutrients <- read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaul_Processed.csv")
PeterPaul.nutrients.gathered <-
  read.csv("./Data/Processed/NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv")
PeterPaul.chem.phys <- read.csv("./Data/Processed/NTL-LTER_Lake_ChemistryPhysics_PeterPaul_Processed.csv")
NIWOLitter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(NIWOLitter$collectDate)

## [1] "factor"
```

```

NIWOLitter$collectDate <- as.Date(NIWOLitter$collectDate, format = "%Y-%m-%d")

class(PeterPaul.chem.phys$sampledte)

## [1] "factor"
PeterPaul.chem.phys$sampledte <- as.Date(PeterPaul.chem.phys$sampledte, format = "%Y-%m-%d")

class(PeterPaul.nutrients$sampledte)

## [1] "factor"
PeterPaul.nutrients$sampledte <- as.Date(PeterPaul.nutrients$sampledte, format = "%Y-%m-%d")

class(PeterPaul.nutrients.gathered$sampledte)

## [1] "factor"
PeterPaul.nutrients.gathered$sampledte <- as.Date(PeterPaul.nutrients.gathered$sampledte,
                                                    format = "%Y-%m-%d")

```

Define your theme

3. Build a theme and set it as your default theme.

```

#3
mytheme <- theme_classic(base_size = 16) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
theme_set(mytheme)

```

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 by phosphate, 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.

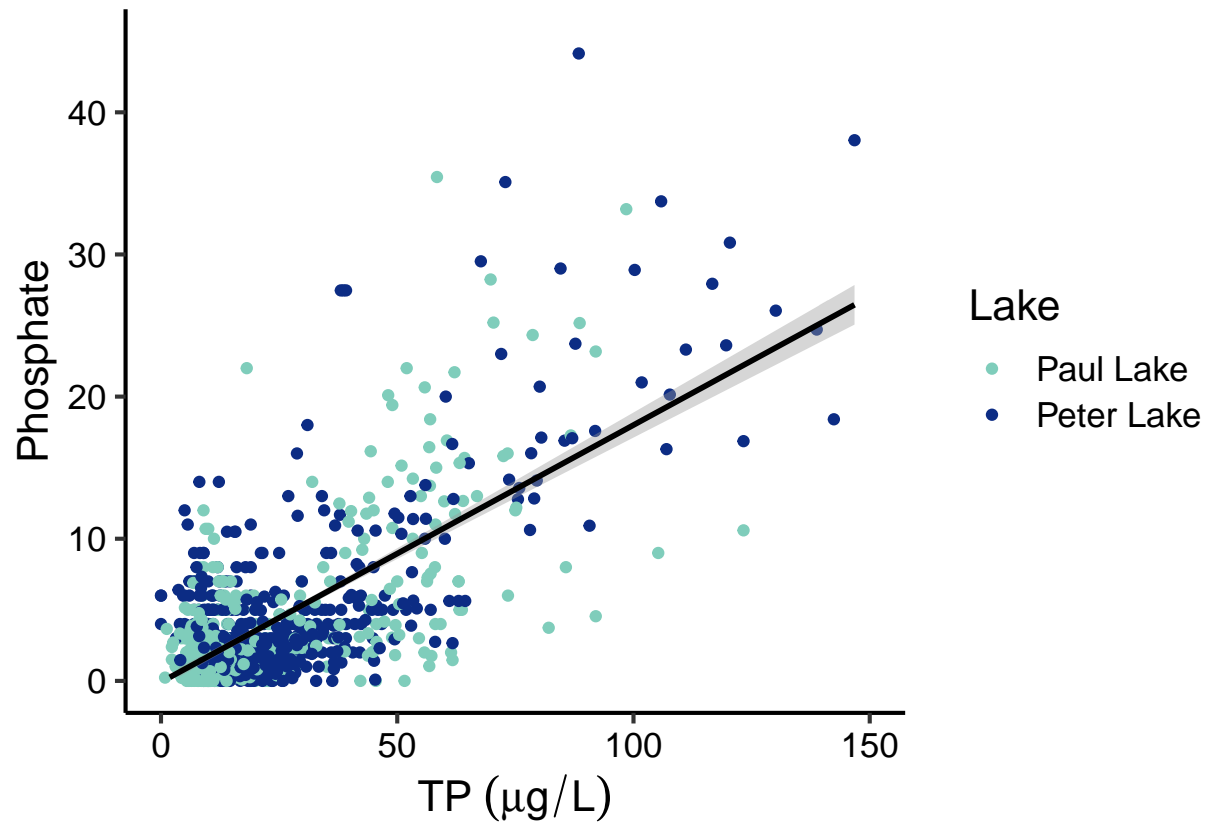
```

#4
PvsP04plot <-
  ggplot(PeterPaul.nutrients, aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() +
  labs(x = expression(TP ~ (mu*g / L)), y = "Phosphate",
       color = "Lake", shape = "") +
  scale_color_manual(values = c("#7fcdbb", "#0c2c84")) +
  xlim(0, 150) +
  ylim(0, 45) +
  geom_smooth(method = lm, color = "black")

print(PvsP04plot)

## Warning: Removed 1346 rows containing non-finite values (stat_smooth).
## Warning: Removed 1346 rows containing missing values (geom_point).
## Warning: Removed 1 rows containing missing values (geom_smooth).

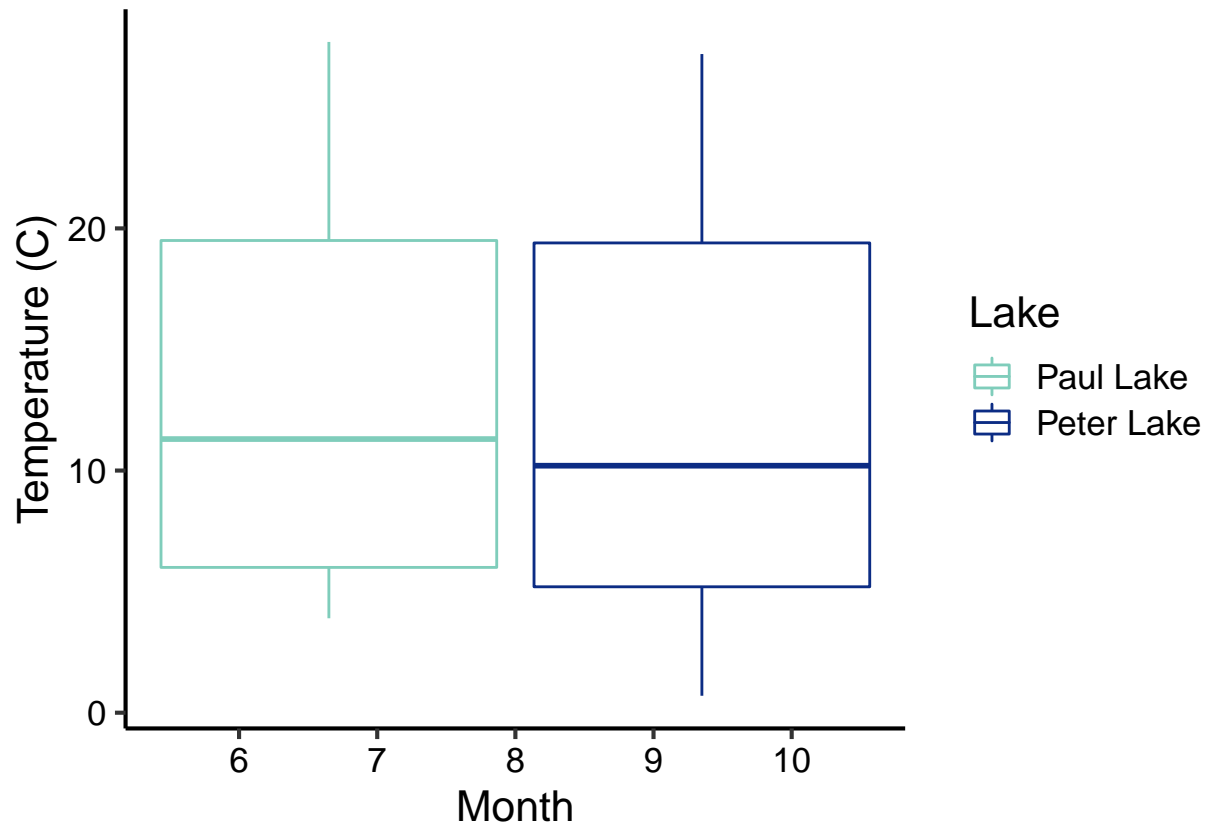
```



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.

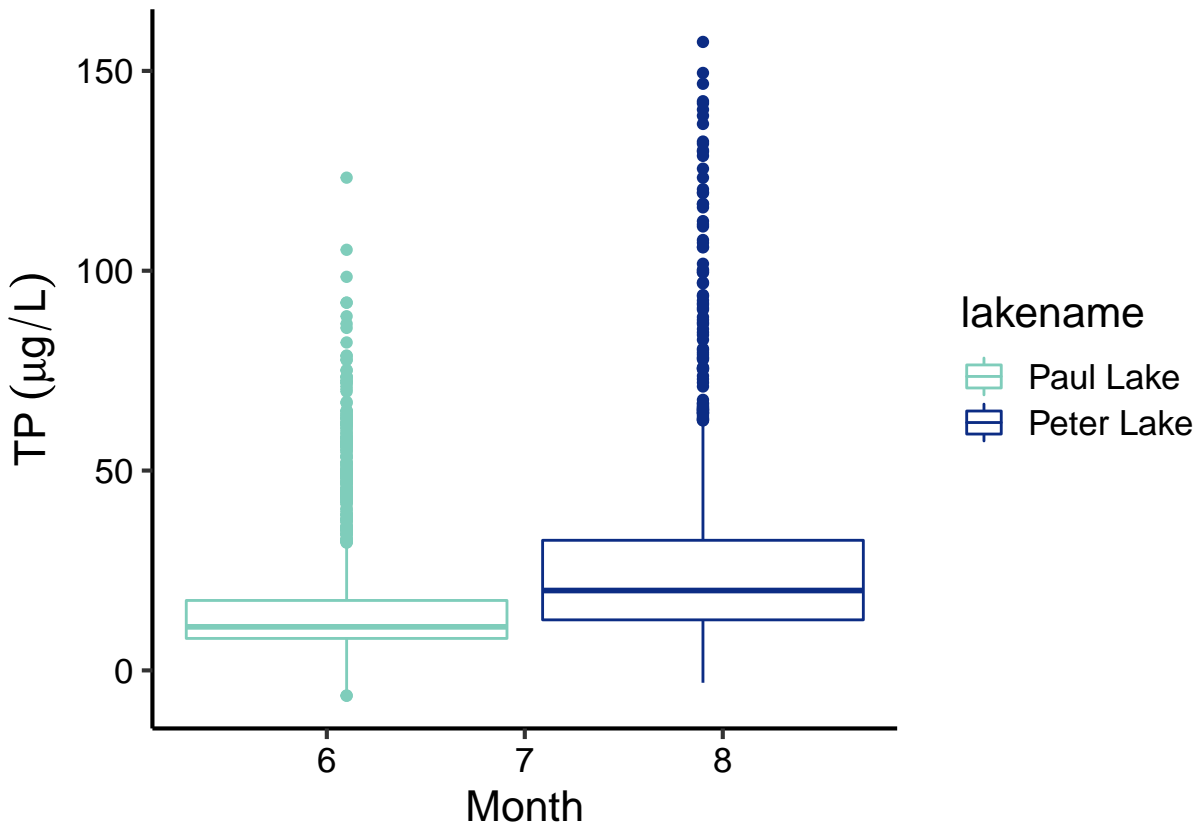
```
#5
TempPlot <- ggplot(PeterPaul.chem.phys) +
  geom_boxplot(aes(x = month, y = temperature_C, color = lakename)) +
  labs(x = "Month", y = "Temperature (C)",
       color = "Lake", shape = "") +
  scale_color_manual(values = c("#7fcdbb", "#0c2c84"))
print(TempPlot)
```

```
## Warning: Removed 2171 rows containing non-finite values (stat_boxplot).
```



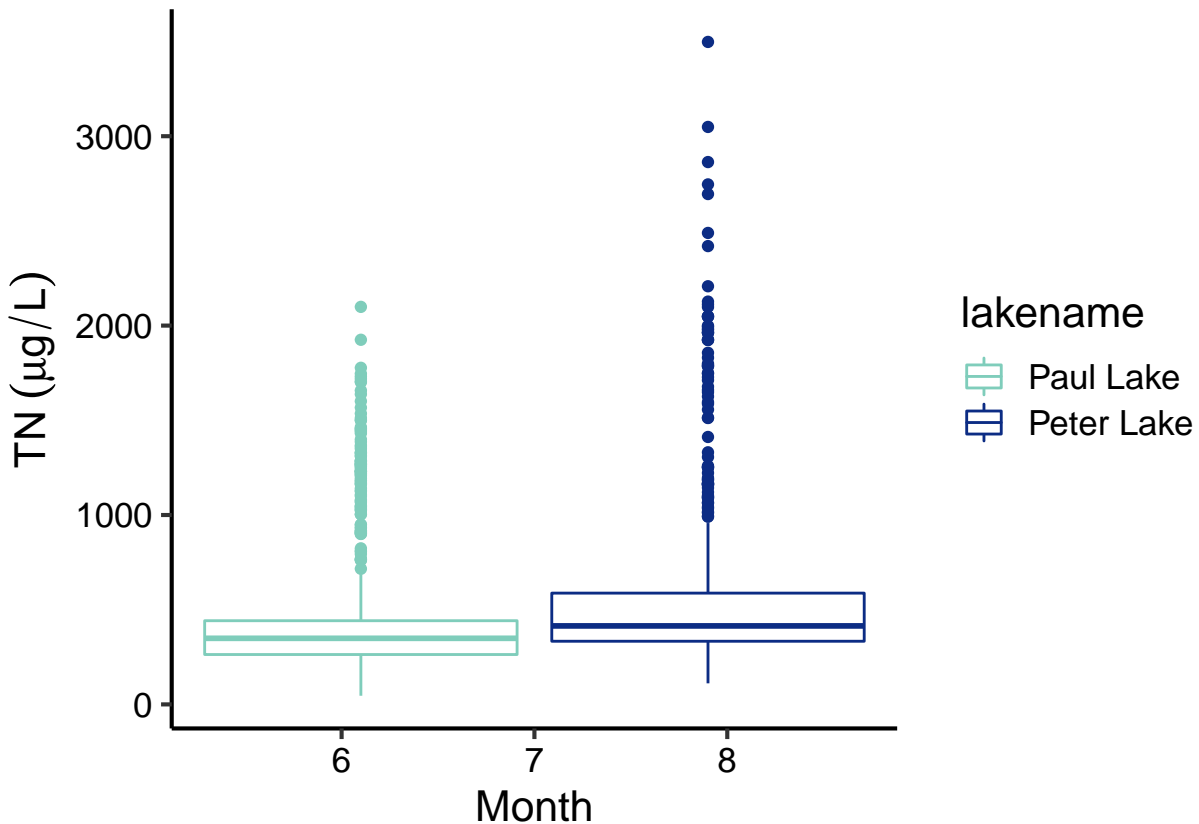
```
TPplot <- ggplot(PeterPaul.nutrients) +
  geom_boxplot(aes(x = month, y = tp_ug, color = lakename)) +
  labs(x = "Month", y = expression(TP ~ (mu*g / L),
    color = "Lake", shape = "")) +
  scale_color_manual(values = c("#7fcdbb", "#0c2c84"))
print(TPplot)
```

```
## Warning: Removed 127 rows containing non-finite values (stat_boxplot).
```



```
TNplot <- ggplot(PeterPaul.nutrients) +
  geom_boxplot(aes(x = month, y = tn_ug, color = lakename)) +
  labs(x = "Month", y = expression(TN ~ (mu*g / L),
    color = "Lake", shape = "")) +
  scale_color_manual(values = c("#7fcdbb", "#0c2c84"))
print(TNplot)
```

```
## Warning: Removed 981 rows containing non-finite values (stat_boxplot).
```



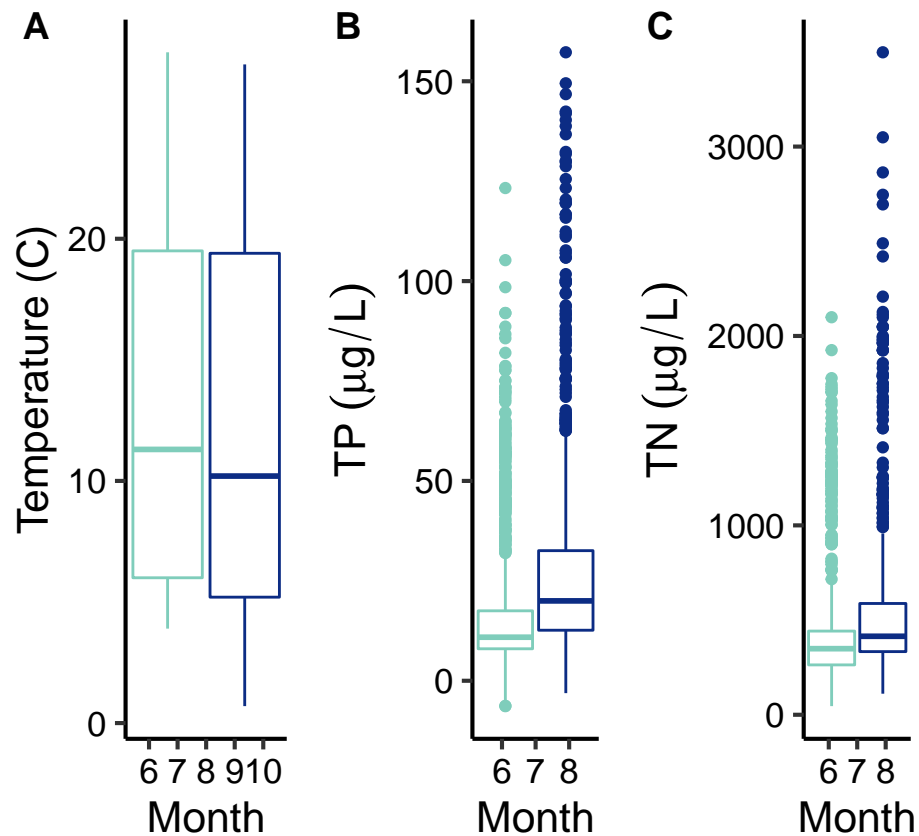
```
CombinedPlot <- plot_grid(
  TempPlot + theme(legend.position="none"),
  TPpplot + theme(legend.position="none"),
  TNpplot + theme(legend.position="none"),
  labels = c("A", "B", "C"),
  hjust = -1,
  nrow = 1,
  rel_heights = c(1.25, 1),
  axis = "bt",
  legend)

## Warning: Removed 2171 rows containing non-finite values (stat_boxplot).
## Warning: Removed 127 rows containing non-finite values (stat_boxplot).
## Warning: Removed 981 rows containing non-finite values (stat_boxplot).
## Warning: Package `gridGraphics` is required to handle base-R plots. Substituting
## empty plot.

legend <- get_legend(
  TempPlot + theme(legend.box.margin = margin(0, 0, 0, 12)))

## Warning: Removed 2171 rows containing non-finite values (stat_boxplot).
```

```
print(CombinedPlot)
```



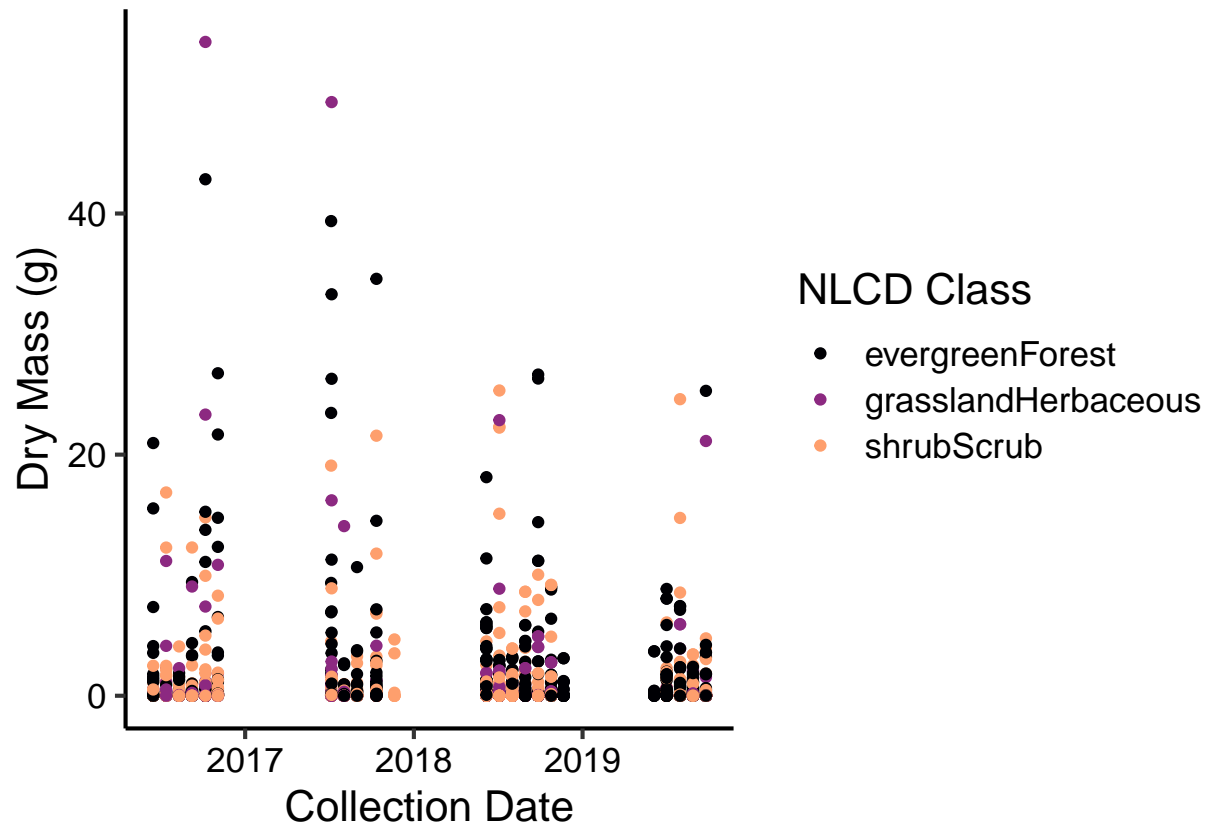
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: The temperature does not vary much between the two lake, but Peter Lake has a higher nutrient concentration compared to Paul Lake.

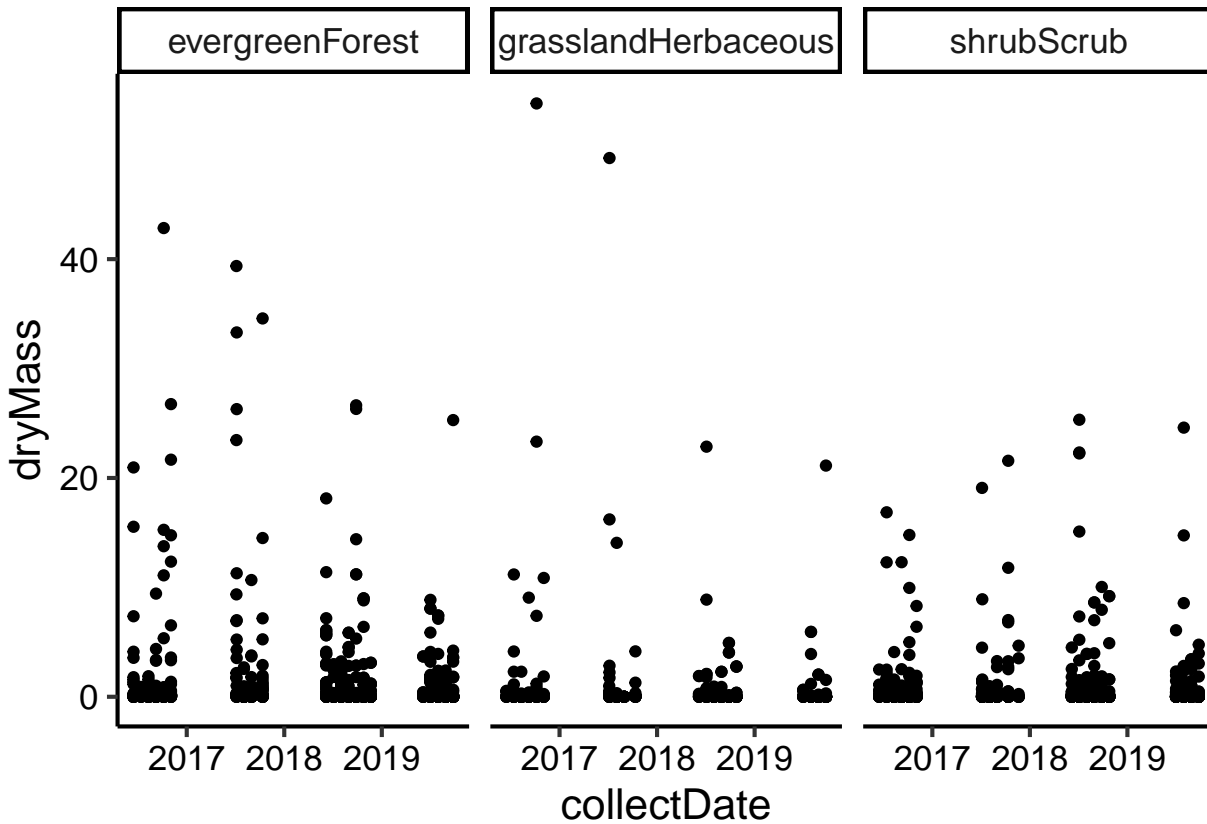
- [Niwot Ridge] Plot a subset of the litter dataset by displaying only the “Needles” functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
- [Niwot Ridge] Now, plot the same plot but with NLCD classes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

#6

```
NeedlesPlot1 <-
  ggplot(subset(NIWOLitter, functionalGroup = Needles),
    aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  labs(x = "Collection Date", y = "Dry Mass (g)",
    color = "NLCD Class", shape = "") +
  scale_color_viridis(discrete = TRUE, option = "magma", end = 0.8) +
  geom_point()
print(NeedlesPlot1)
```



```
#7
NeedlesPlot2 <-
  ggplot(subset(NIWOLitter, functionalGroup = Needles),
    aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass))
print(NeedlesPlot2)
```

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

Answer: Plot 7 (the facet-separated one) is more effective than plot 6 because the facets allow the collect date for each type of NLCD class to be seen more clearly than when they are on one plot like they are in 6. Although each NLCD class is color-coded in plot 6, it is still difficult to distinguish between collect dates for each NLCD class.