

IreneChang_A05_DataVisualization.Rmd

Irene Chang

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
library(tidyverse)

## -- 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()
```

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
##  
## The following objects are masked from 'package:base':  
##  
##    date, intersect, setdiff, union
```

```
library(cowplot)
```

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

```
library(here)
```

```
## here() starts at /home/guest/EDA/EDA-Spring2023
```

```
here()
```

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

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

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

```
#2
```

```
class(NIWOLitter.Processed$collectDate)
```

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

```
NIWOLitter.Processed$collectDate <- ymd(NIWOLitter.Processed$collectDate)
```

```
class(PeterPaulNutrients.Processed$sampledate)
```

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

```
PeterPaulNutrients.Processed$sampldate <- ymd(PeterPaulNutrients.Processed$sampldate)
```

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() +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
#mytheme is a classic looking plot with no gridlines.
#The axis text will be black and the legend will be at the top.

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 (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()).

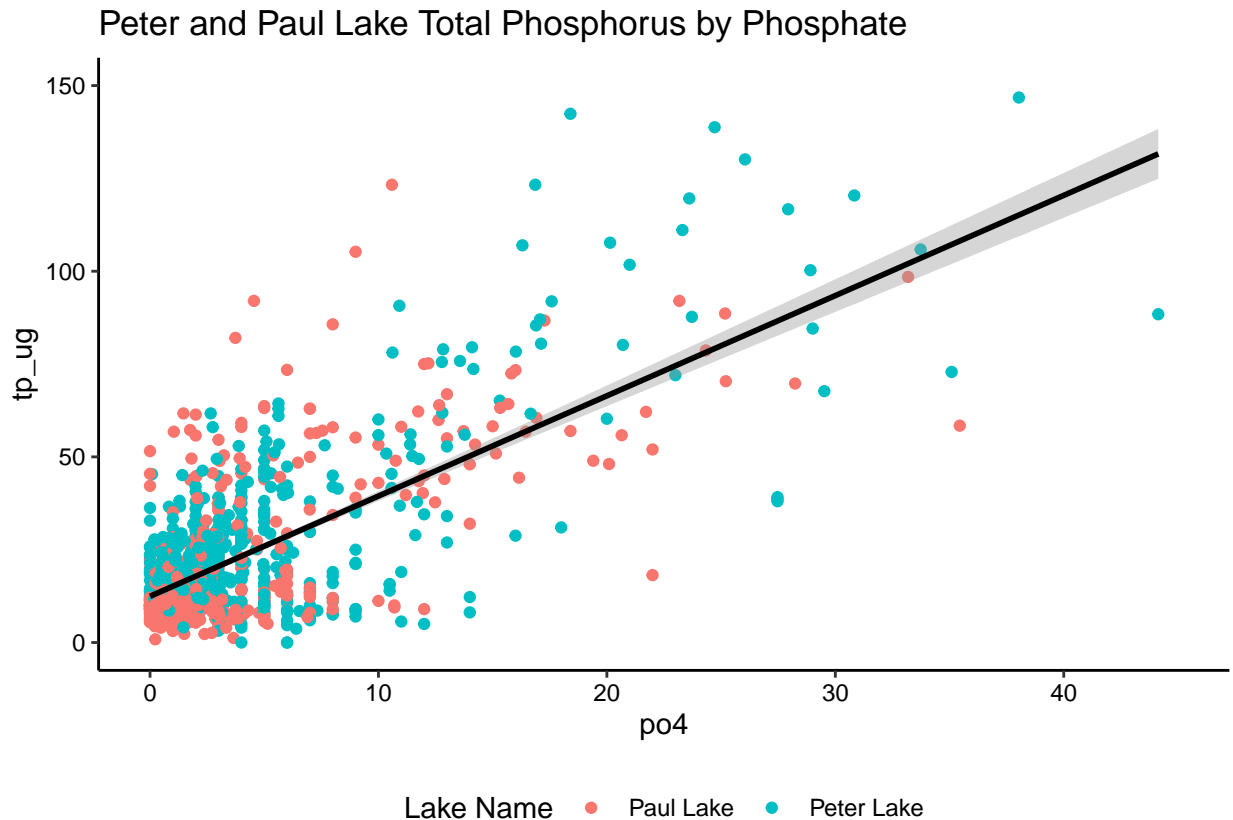
```
#4

totalPandP04 <-
  ggplot(PeterPaulNutrients.Processed, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  xlim(0, 45) +
  ylim(0, 150) +
  geom_smooth(method = lm, color = "black") +
  labs(
    title = "Peter and Paul Lake Total Phosphorus by Phosphate",
    color = "Lake Name")
print(totalPandP04)
```

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



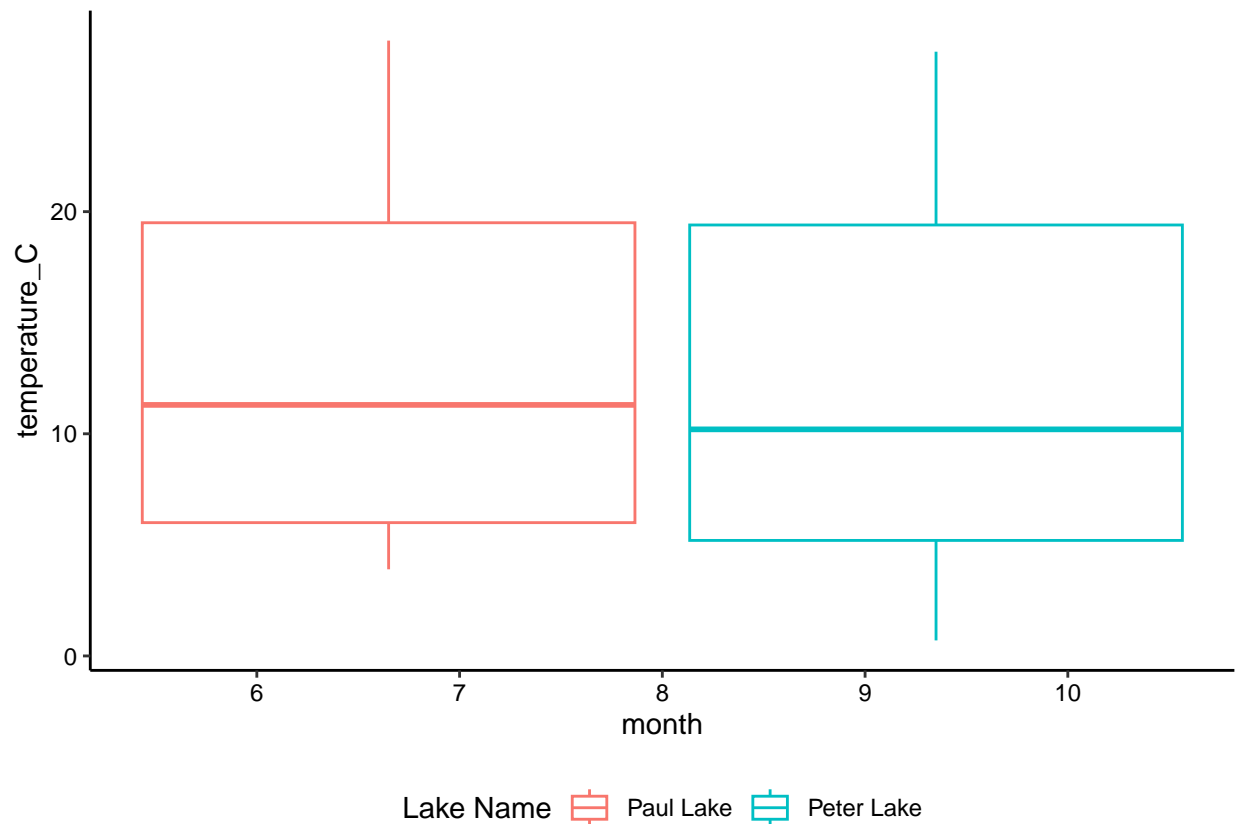
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

Nutrients.temp <-
  ggplot(PeterPaulNutrients.Processed,
    aes(x = month,
        y = temperature_C,
        color = lakename)) +
  labs(color = "Lake Name") +
  geom_boxplot()
print(Nutrients.temp)
```

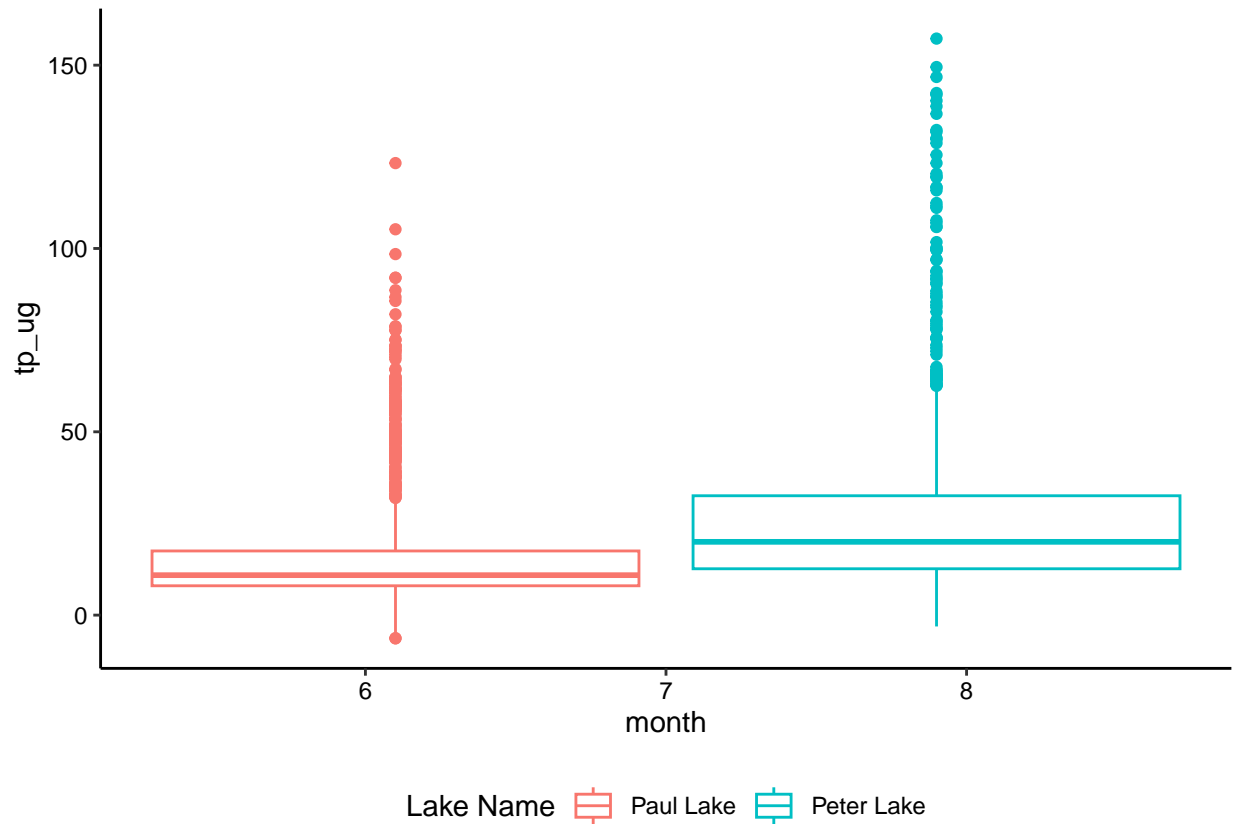
```
## Warning: Removed 3566 rows containing non-finite values ('stat_boxplot').
```



#alternative to see by month is to use aes(x = month.abb[month]) to see the breakdown of temp, tp, and

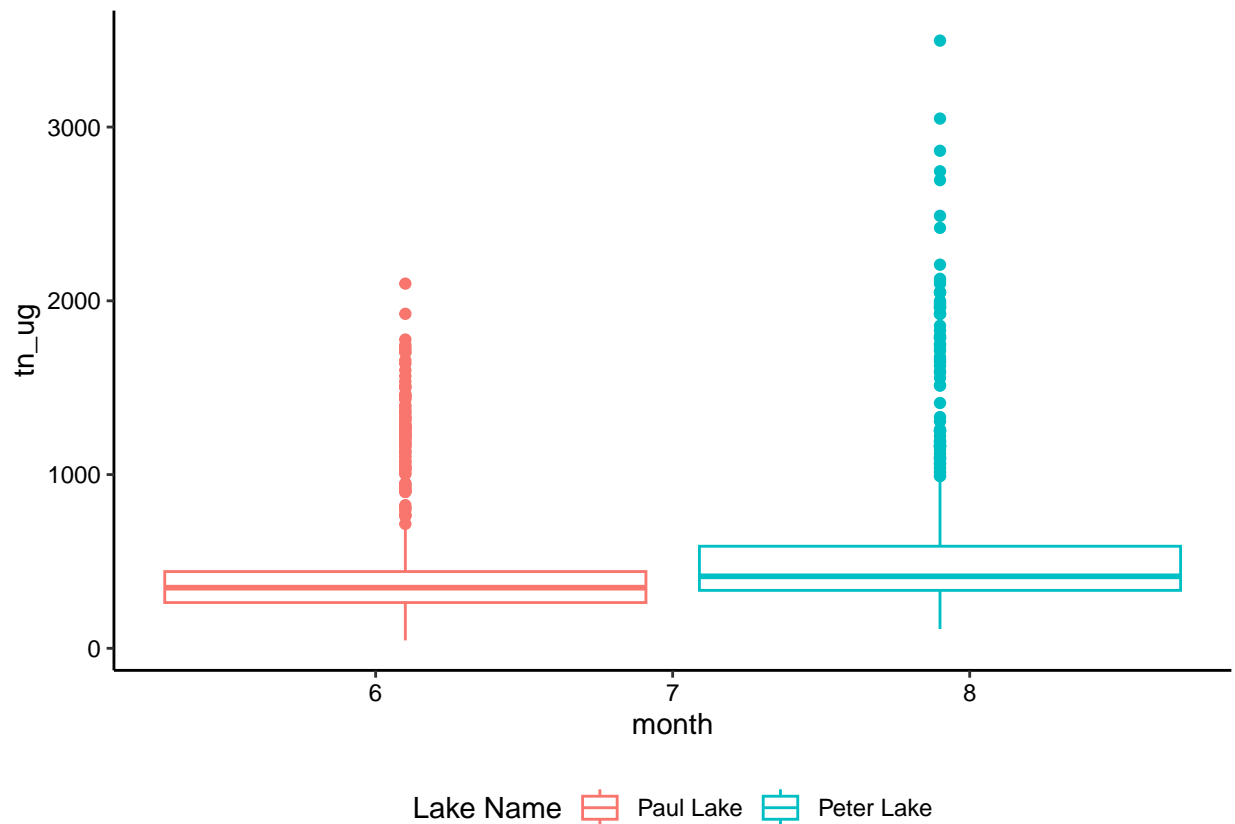
```
Nutrients.tp <-
  ggplot(PeterPaulNutrients.Processed,
    aes(x = month,
        y = tp_ug,
        color = lakename)) +
  labs(color = "Lake Name") +
  geom_boxplot()
print(Nutrients.tp)
```

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



```
Nutrients.tn <-
  ggplot(PeterPaulNutrients.Processed,
    aes(x = month,
        y = tn_ug,
        color = lakename)) +
  labs(color = "Lake Name") +
  geom_boxplot()
print(Nutrients.tn)
```

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



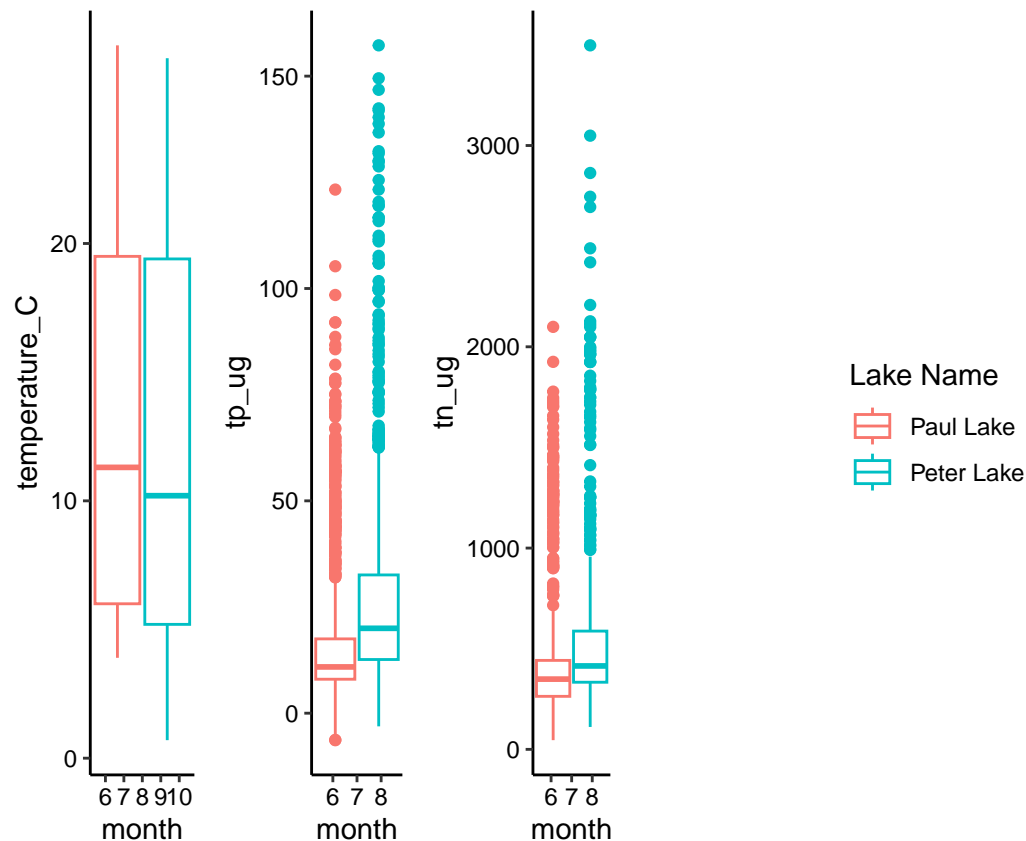
```
Nutrients.temp.tp.tn <-
  plot_grid(Nutrients.temp + theme(legend.position="none"),
    Nutrients.tp + theme(legend.position = "none"),
    Nutrients.tn + theme(legend.position = "none"),
    nrow= 1,
    align = "h")

## 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()').

legend <- get_legend(Nutrients.temp + theme(legend.position = "right"))

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

plot_grid(Nutrients.temp.tp.tn, legend)
```



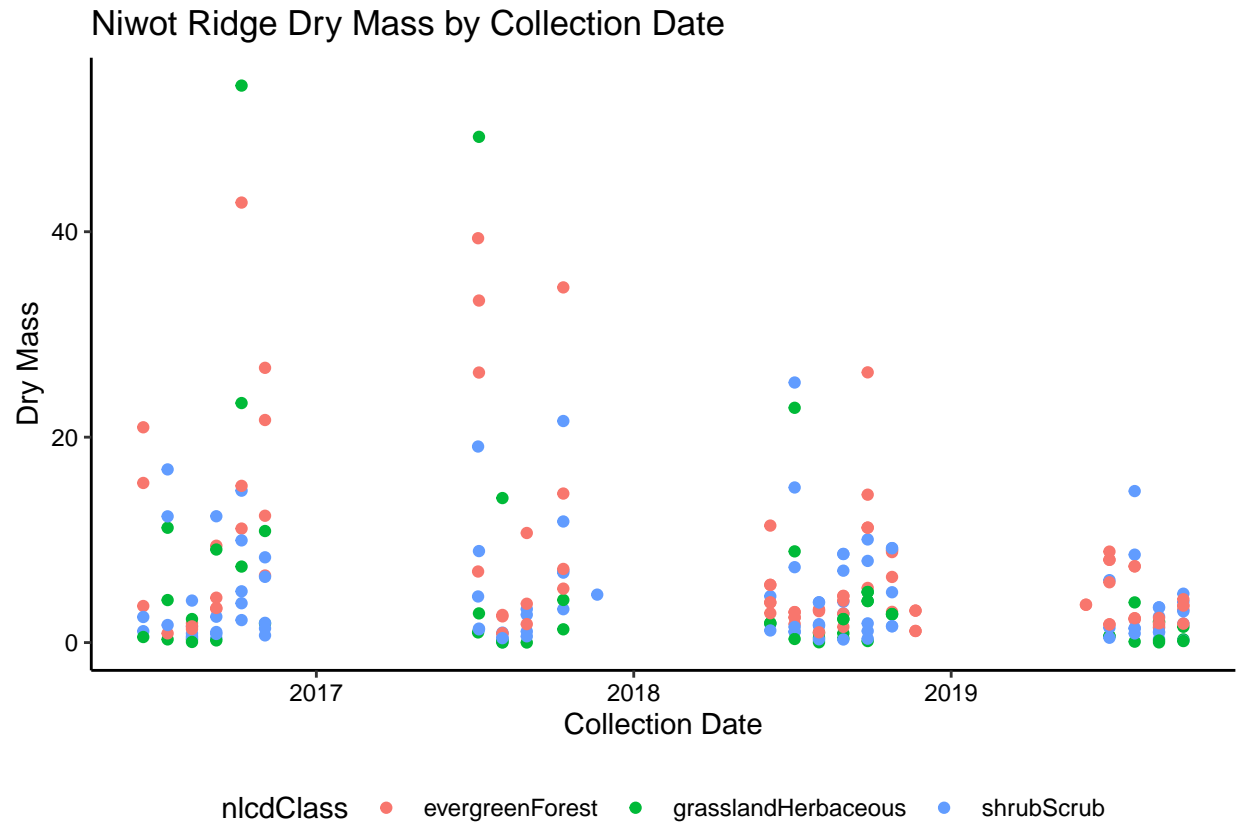
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: The median temperature is a bit higher overall at Paul Lake than Peter Lake. For both lakes, the median warmest month is September. The total phosphorus levels for Peter Lake is higher overall than Paul Lake. For both lakes, the highest median total phosphorus levels were observed in September. The total nitrogen levels are also higher overall for Peter lake than Paul Lake. For Peter Lake, the highest total nitrogen levels were observed in August. For both the highest median total nitrogen levels were observed in September.

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.

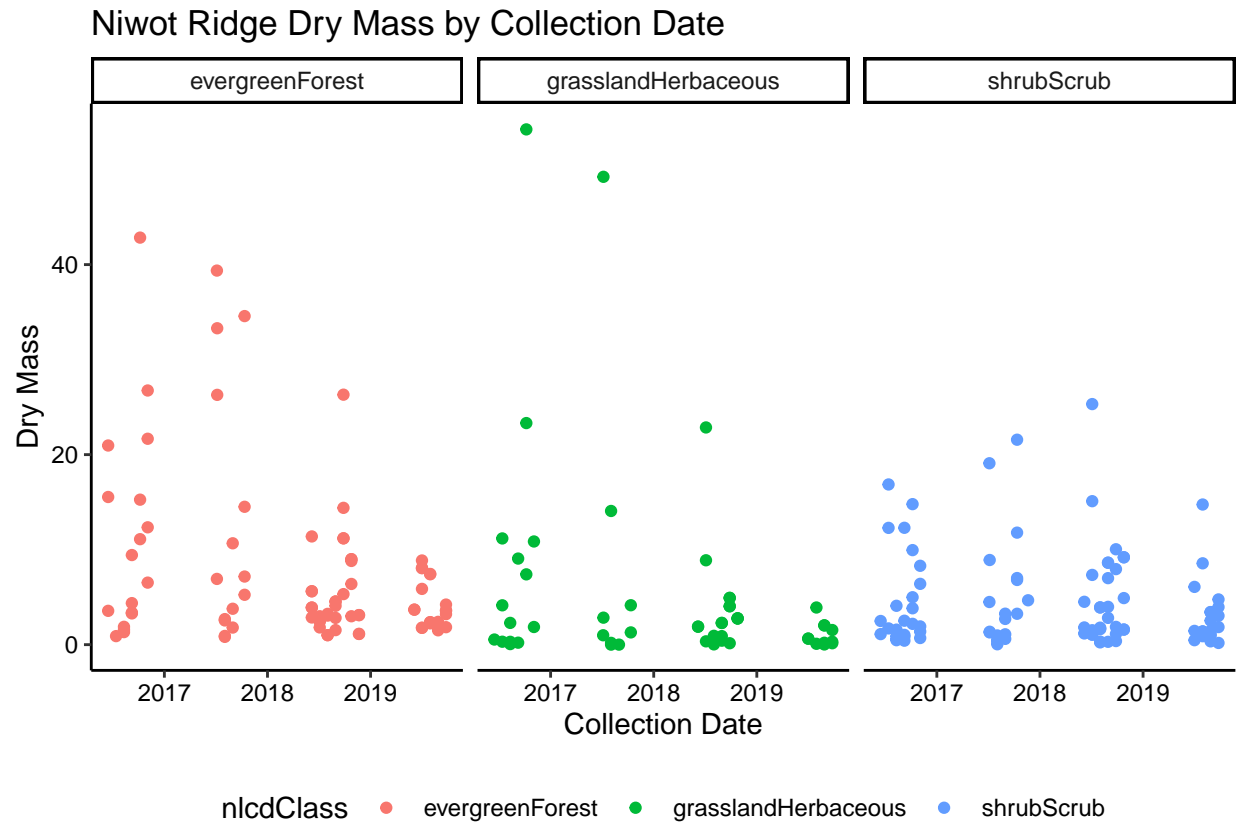
#6

```
Litter.Needles <-
  ggplot(subset(NIWOLitter.Processed, functionalGroup == "Needles"),
    aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(
    title = "Niwot Ridge Dry Mass by Collection Date",
    x = "Collection Date",
    y = "Dry Mass")
print(Litter.Needles)
```

```
#7

Litter.Needles.Facets <- NIWOLitter.Processed %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate,
             y = dryMass,
             color = nlcdClass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(
    title = "Niwot Ridge Dry Mass by Collection Date",
    x = "Collection Date",
    y = "Dry Mass")
print(Litter.Needles.Facets)
```



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

Answer: I think that the plot from #7 is more effective because the data kind of gets lost when you have it all on one graph, but when you create 3 facets, then the changes over time within nlcdClass become more clear and its a better visualization to compare them across nlcdClass.