Assignment 5: Data Visualization

Jordan Mullens

Spring 2023

OVERVIEW

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

Directions

1. Rename this file `Jordan_Mullens_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 packages
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## 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
## v purrr   1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(here)
```

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

```
here()
```

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

```
PeterPaul.chem.nutrients <-
  read.csv(here("Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
PeterPaul.Niwot.Litter <-
  read.csv(here("Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)

#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
PeterPaul.Niwot.Litter$collectDate <- ymd(PeterPaul.Niwot.Litter$collectDate)
class(PeterPaul.chem.nutrients$sampledate)
```

```
## [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

```r
#3
library(ggthemes)
my_theme <- theme_base() +
  theme(
    line = element_line(
      color='black',
      linewidth =2
    ),
    legend.background = element_rect(
      color='grey',
      fill = 'green'
    ),
    legend.title = element_text(
      color='blue'
    ))
```

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()`).

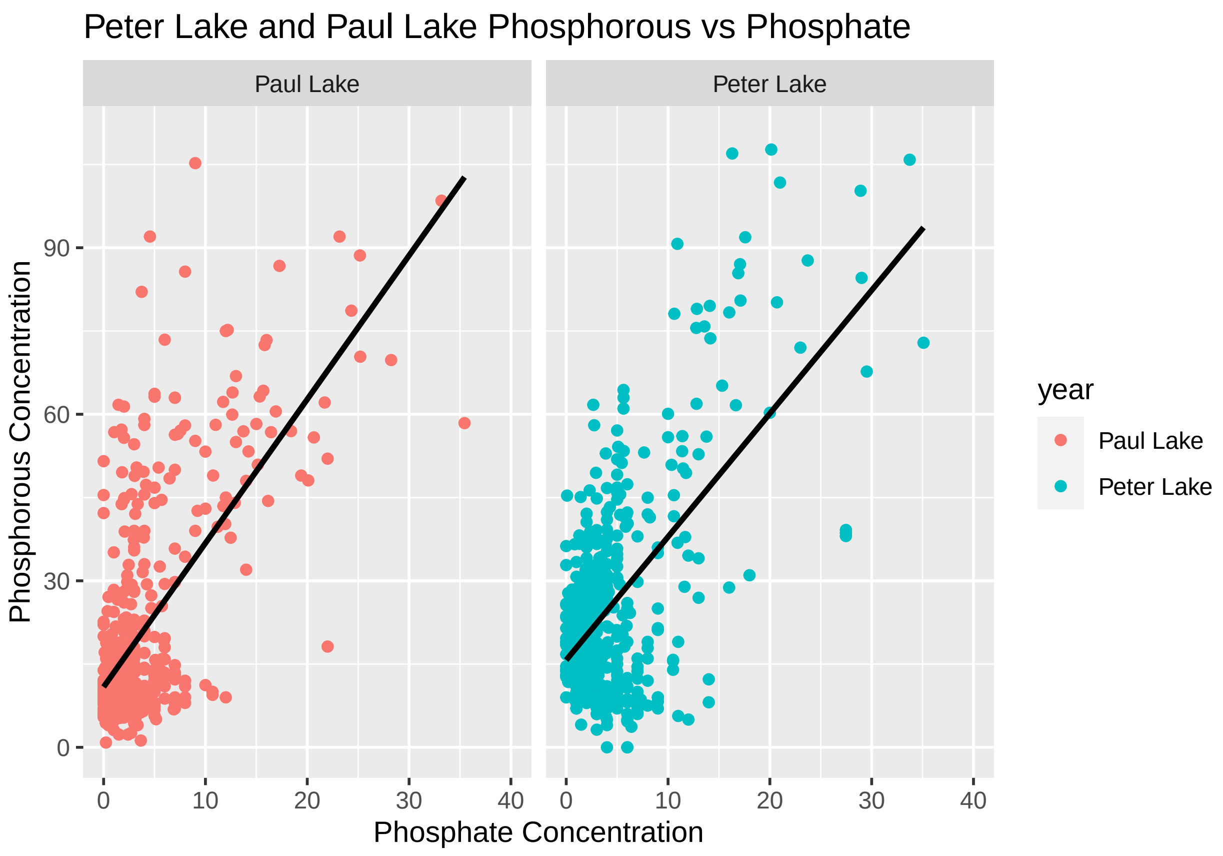
```r
#4
peterpaul.Phos.plot1 <-
  ggplot(PeterPaul.chem.nutrients, aes(
         x = po4,
         y = tp_ug,
         color = lakename)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE, color= "black") +
   xlim(0, 40) +
  ylim(0, 110) +
  facet_wrap(vars(lakename)) +
  labs(
    title = "Peter Lake and Paul Lake Phosphorous vs Phosphate",
    y= "Phosphorous Concentration",
    x= "Phosphate Concentration",
    color= "year")

print(peterpaul.Phos.plot1)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 21959 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 21959 rows containing missing values (‘geom_point()‘).
```



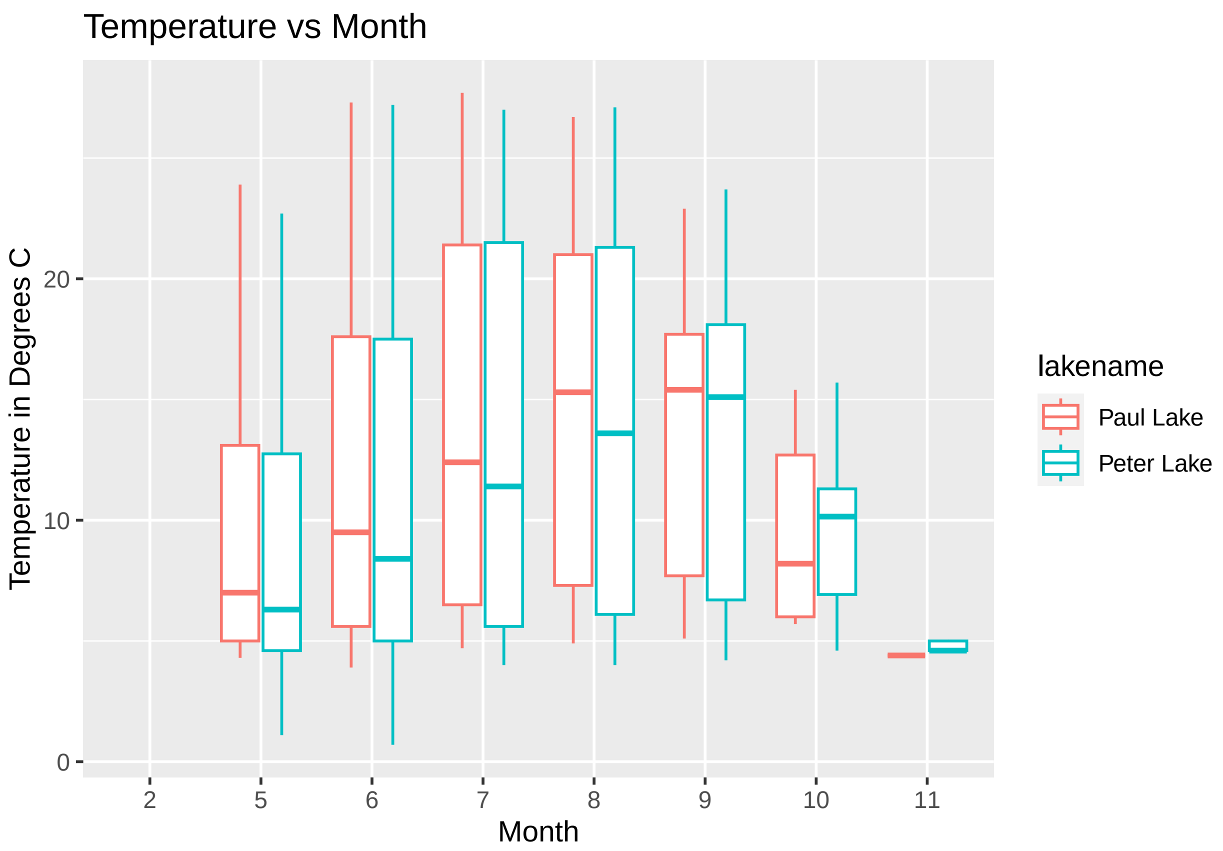
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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
#Temperature Box Plot
peterpaul.Temp.box.plot <-
  ggplot(PeterPaul.chem.nutrients, aes(
         x = factor(month),
         y = temperature_C,
         color = lakename)) +
  geom_boxplot() +
  labs(
    title = "Temperature vs Month",
    y= "Temperature in Degrees C",
    x= "Month")

print(peterpaul.Temp.box.plot)
```

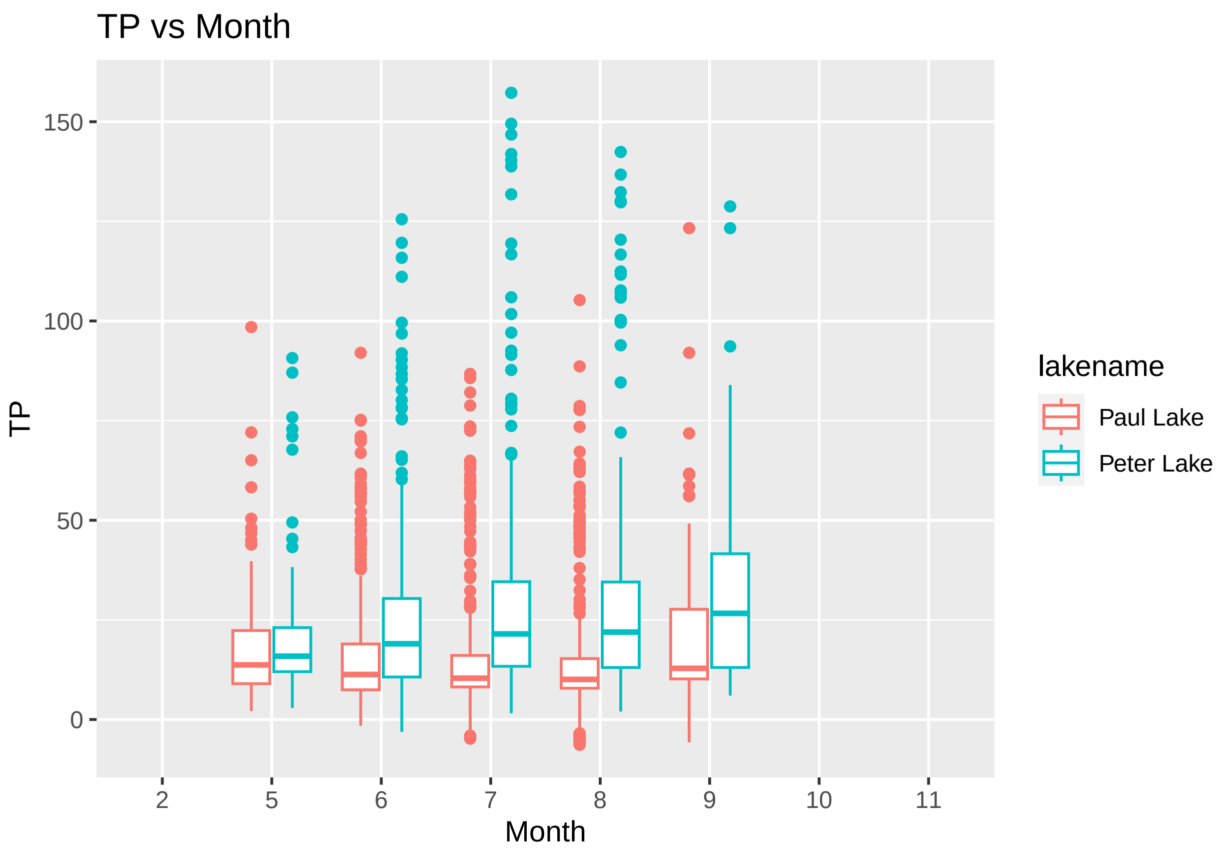
```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```



```r
#TP Box Plot
peterpaul.TP.box.plot <-
  ggplot(PeterPaul.chem.nutrients, aes(
         x = factor(month),
         y = tp_ug,
         color = lakename)) +
  geom_boxplot() +
  labs(
    title = "TP vs Month",
    y= "TP",
    x= "Month")

print(peterpaul.TP.box.plot)
```

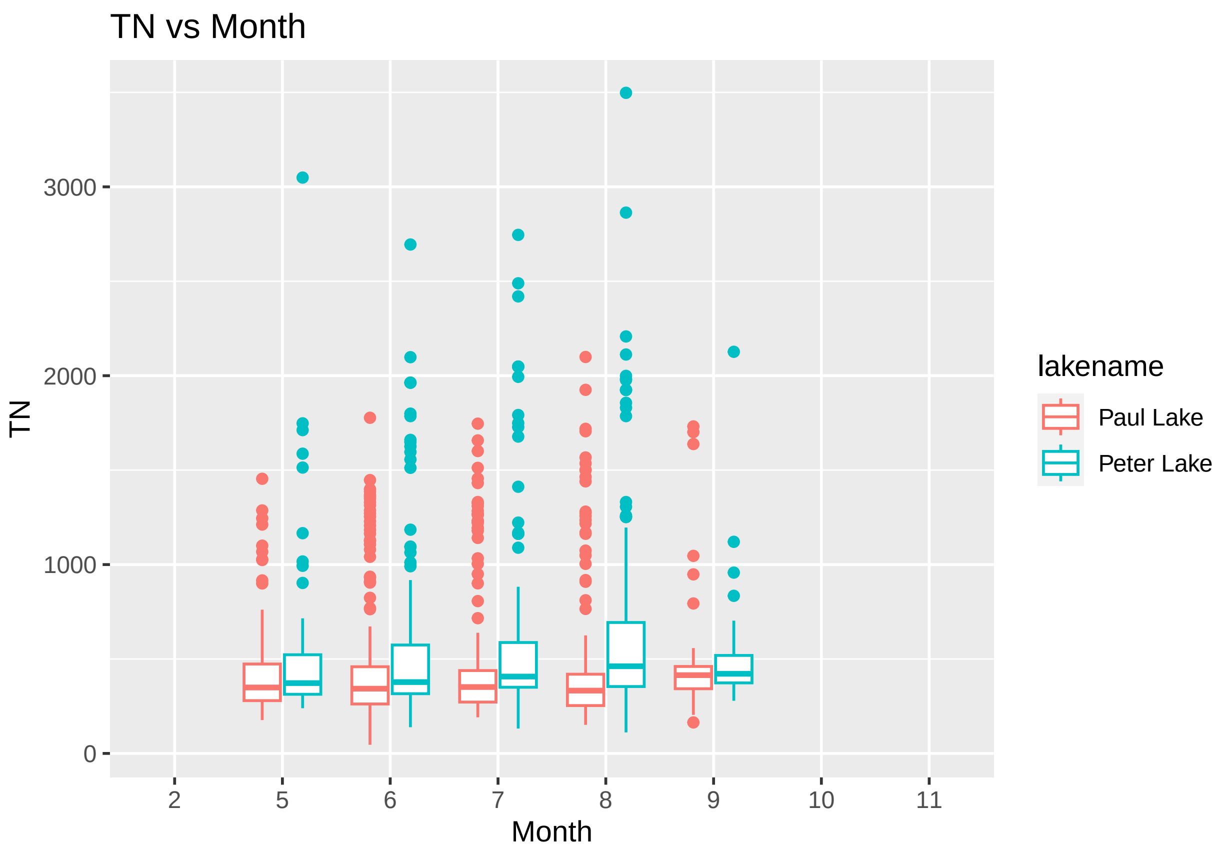
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```



```
#TN Box Plot
peterpaul.TN.box.plot <-
  ggplot(PeterPaul.chem.nutrients, aes(
         x = factor(month),
         y = tn_ug,
         color = lakename)) +
  geom_boxplot() +
  labs(
    title = "TN vs Month",
    y= "TN",
    x= "Month")

print(peterpaul.TN.box.plot)
```

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



```
#Creating a cowplot
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

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

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

```
legend_cow <- get_legend(
  peterpaul.TN.box.plot +
  guides(color = guide_legend(nrow = 1)) +
  theme(legend.position = "right"))
```

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

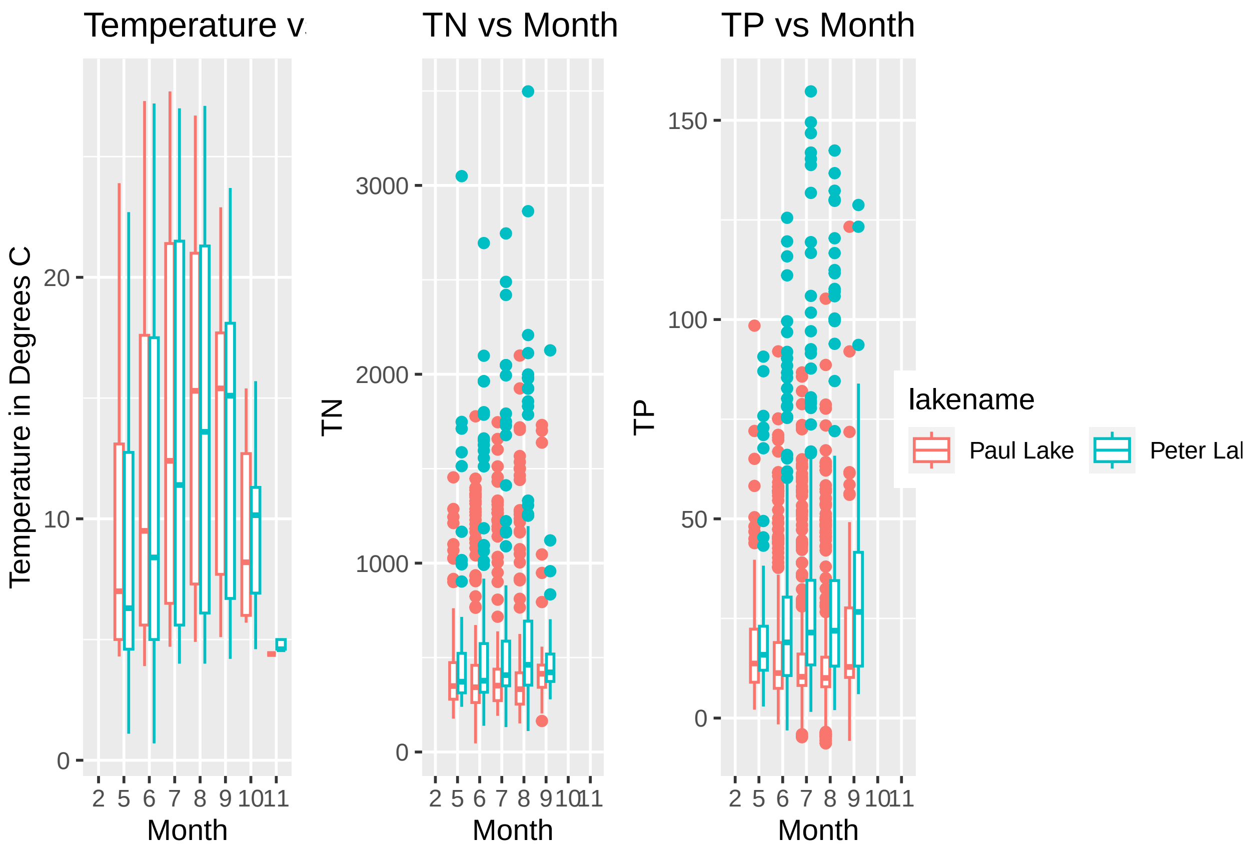
```
peter.paul.cowplot <- plot_grid(
  peterpaul.Temp.box.plot + theme(legend.position="none"),
  peterpaul.TN.box.plot + theme(legend.position="none"),
  peterpaul.TP.box.plot + theme(legend.position="none"),
  hjust = -1,
  nrow = 1,
  legend_cow
  )
```

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

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

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

```
print(peter.paul.cowplot)
```



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

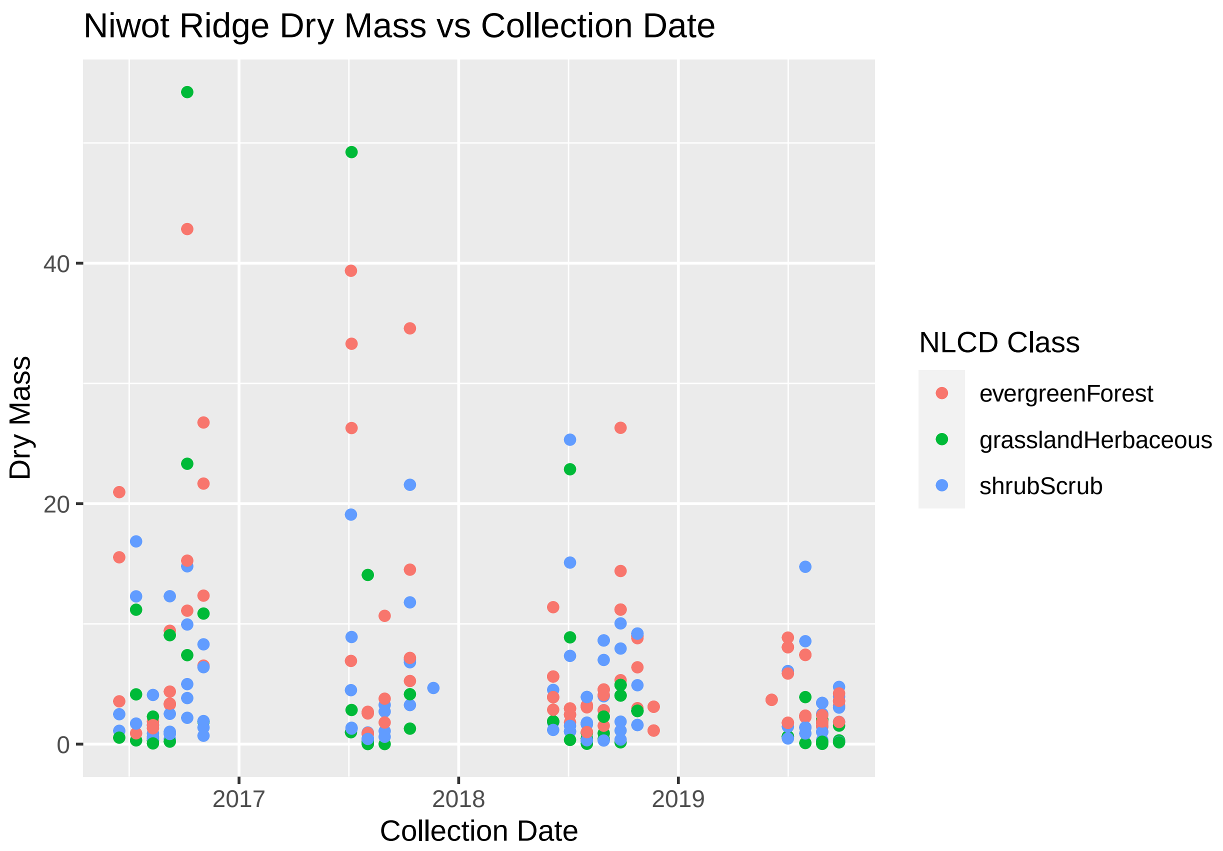
> Answer: Temperature peaks in the summer months for both lakes. TP peaks in the summer for Peter Lake and is at a minimum for Paul lake. TN appears to be relatively consistent throughout the year.

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.

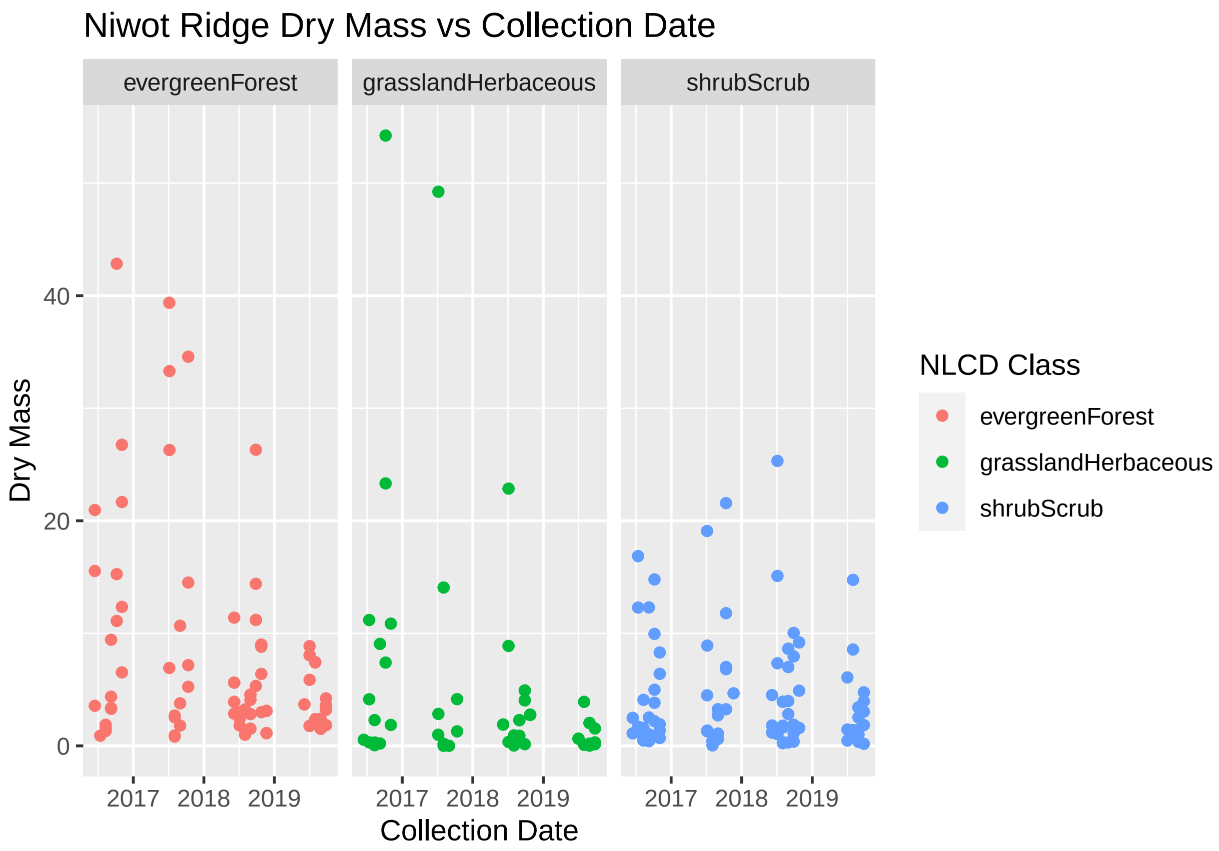
```
#6
Niwot.Ridge.plot1.litter <- PeterPaul.Niwot.Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    aes(
         x = collectDate,
         y = dryMass,
         color = nlcdClass)) +
  geom_point() +
  labs(
    title = "Niwot Ridge Dry Mass vs Collection Date",
    y= "Dry Mass",
    x= "Collection Date",
    color= "NLCD Class")

print(Niwot.Ridge.plot1.litter)
```



```
#7
Niwot.Ridge.plot1.litter <- PeterPaul.Niwot.Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    aes(
          x = collectDate,
          y = dryMass,
          color = nlcdClass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass)) +
  labs(
    title = "Niwot Ridge Dry Mass vs Collection Date",
    y= "Dry Mass",
    x= "Collection Date",
    color= "NLCD Class")

print(Niwot.Ridge.plot1.litter)
```



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

> Answer: The second one is more effective because it's easier to see variations within each NLCD class.