Assignment 5: Data Visualization

Iman Byndloss

Fall 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: Used library() to load relevant packages.
# Used read.csv() to assign each set of raw data as a dataset in RStudio.
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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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 /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

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

```
Nutrients <- read.csv(
  here(
  "Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"
  )
  )

Litter <- read.csv(
  here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv")
  )

#2: # Used class() to determine class for date columns in the two dataframes.
# Used as.Date() to establish format for the date column of each date frame.
# Ensured it was saved to the appropriate place with "<-".
class(Nutrients$sampledate)
```

```
## [1] "character"
```

```
Nutrients$sampledate <- as.Date(Nutrients$sampledate, format = "%Y-%m-%d")

class(Litter$collectDate)
```

```
## [1] "character"
```

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

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: Looked up help(theme).
# Loaded the relevant package.
# Used theme_base() + theme() to establish a custom theme.
library(ggthemes)
```

```
##
## Attaching package: 'ggthemes'
```

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

```
custom_grid <- theme_base() +
  theme(
    panel.background = element_rect(fill="white"),
    panel.grid.major = element_line(color="gray", linewidth = 0.4),
    panel.grid.minor = element_line(color="gray", linewidth = 0.2),
    # Set a grey background with white grid lines
    plot.title = element_text(size=10, face="bold",hjust=0.5),
    # Set text size, emboldened, and centered for plot title
    axis.title = element_text(size=8),
    # Set text size for axis title
    axis.ticks = element_line(color="black"),
    axis.ticks.length=unit(0.15,"cm"),
    # Set length of axis ticks
    legend.box.background = element_rect(color="black", size=0.5),
    # Set a black border around legend
    strip.text = element_text(face="bold", size=8),
    # Set the facet labels to bold small text
    legend.position = "right"
    # Set legend position to right of graphs
    )
```

```
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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: Used ggplot() to graph "tp_ug" versus "po4" for Peter and Paul lakes
tp.po4.plot <- Nutrients %>%
  ggplot(aes(x=tp_ug, y=po4)) +
```

```
  geom_point() +
  facet_wrap(facets=vars(lakename), nrow = 2) +
  geom_smooth(method=lm, se=FALSE) +
  # se=FALSE removes shaded confidence interval
  labs(
    title="Total phsophorus versus phosphate for Peter and Paul lakes",
    x="Total Phosphorus",
    y="Phosphate") +
  xlim(-4, 150) +
  # Adjusted to remove extra black space from x axis
  ylim(0, 50) +
  # Adjusted to hide extreme value above 300 on y axis
  custom_grid
  # Added the customized theme
```

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.

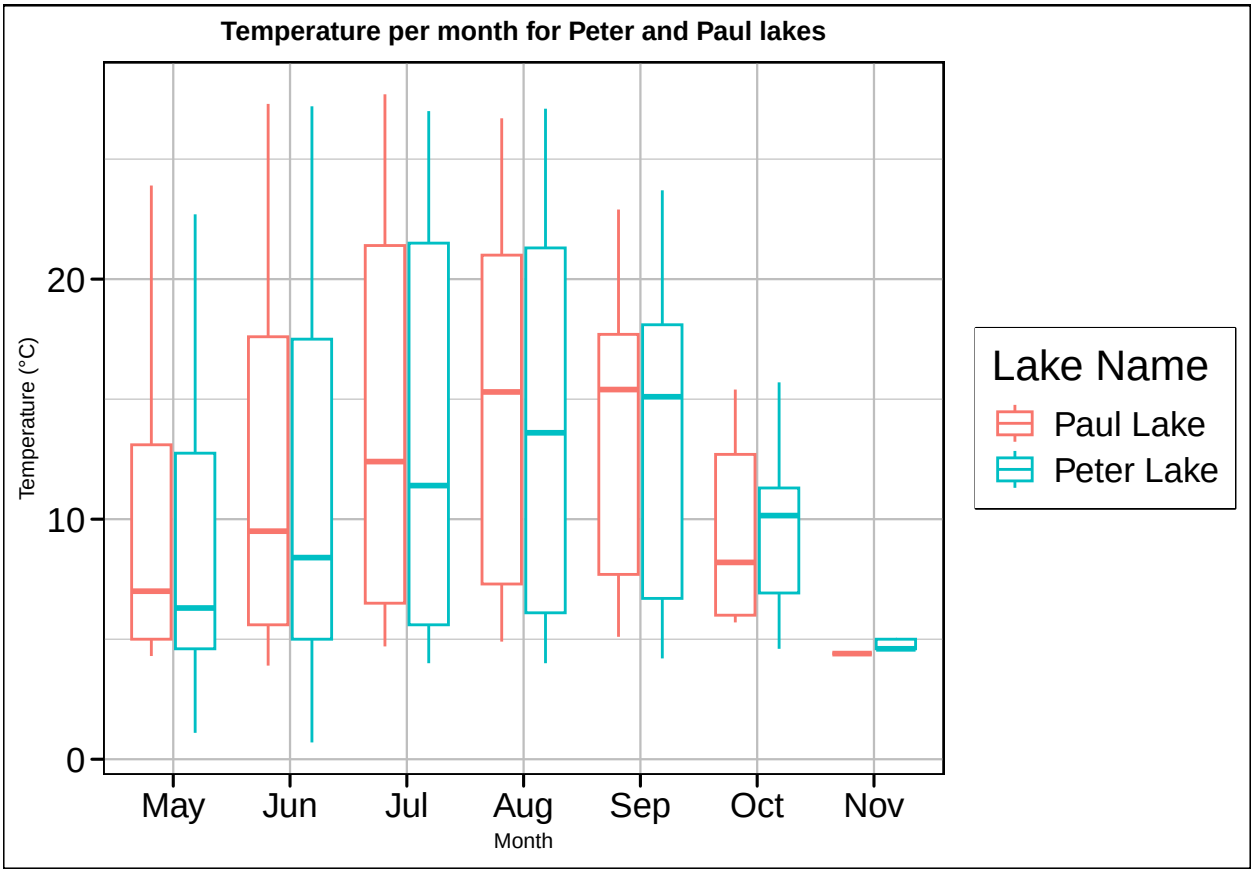
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5: Prior to graphing by month, converted months to abbreviations.
# Then, used ggplot() to creat each of the three plots independently.
# Finally, used cowplot's plot_grid() to combine the plots.
Nutrients$month <- factor(Nutrients$month, levels=1:12, labels=month.abb)

temp.plot <-
  ggplot(Nutrients, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  labs(
    title="Temperature per month for Peter and Paul lakes",
    x="Month",
    y="Temperature (°C)",
    color="Lake Name") +
  scale_x_discrete(limits = month.abb[5:11]) +
  # Set limit for x axis from May to Nov (the months with data for temp)
  custom_grid
print(temp.plot)
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (‘stat_boxplot()‘).
```

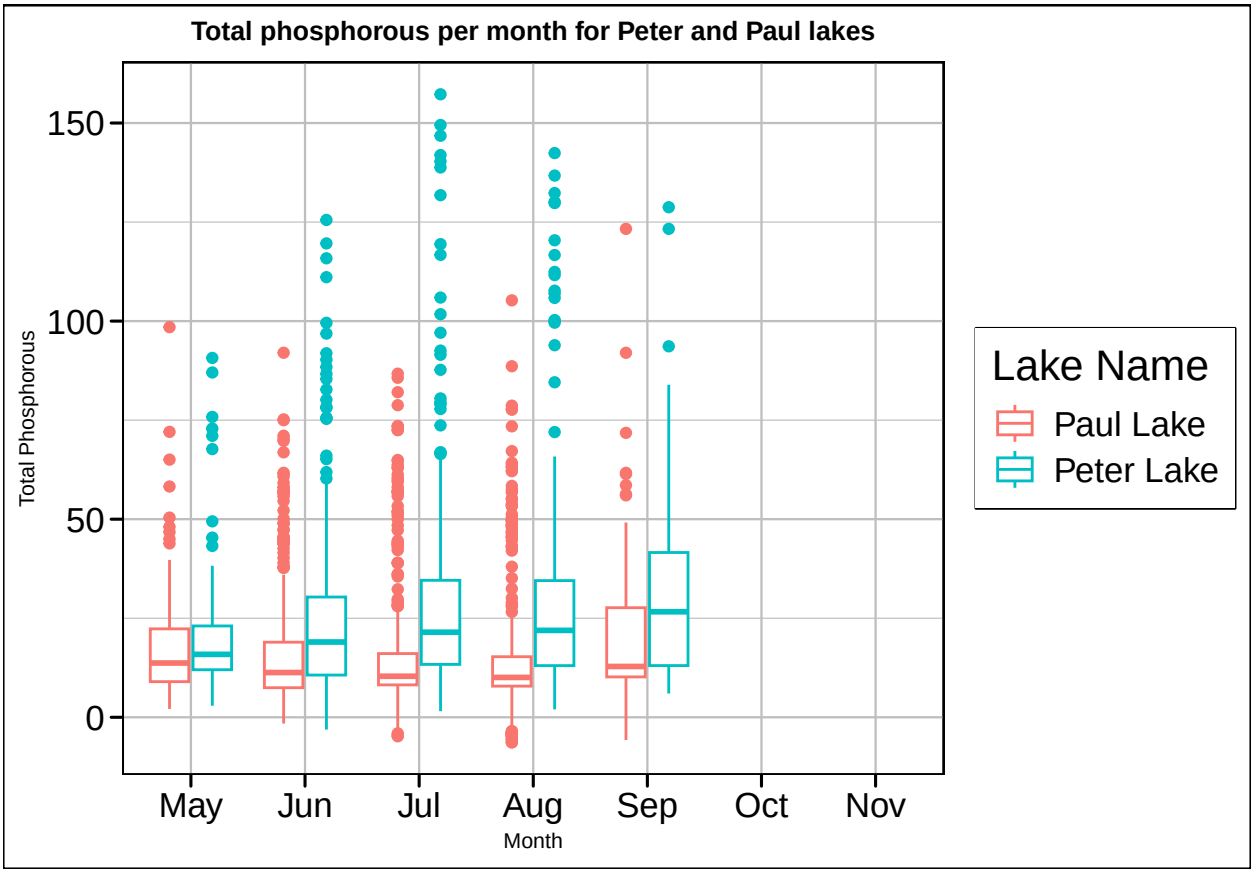
```
## Warning: Removed 3550 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
tp.plot <-
  ggplot(Nutrients, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(
    title="Total phosphorous per month for Peter and Paul lakes",
    x="Month",
    y="Total Phosphorous",
    color="Lake Name") +
  scale_x_discrete(limits = month.abb[5:11]) +
  # Set limit for x axis from May to Nov (for the sake of upcoming combining)
  custom_grid
print(tp.plot)
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (‘stat_boxplot()‘).
```

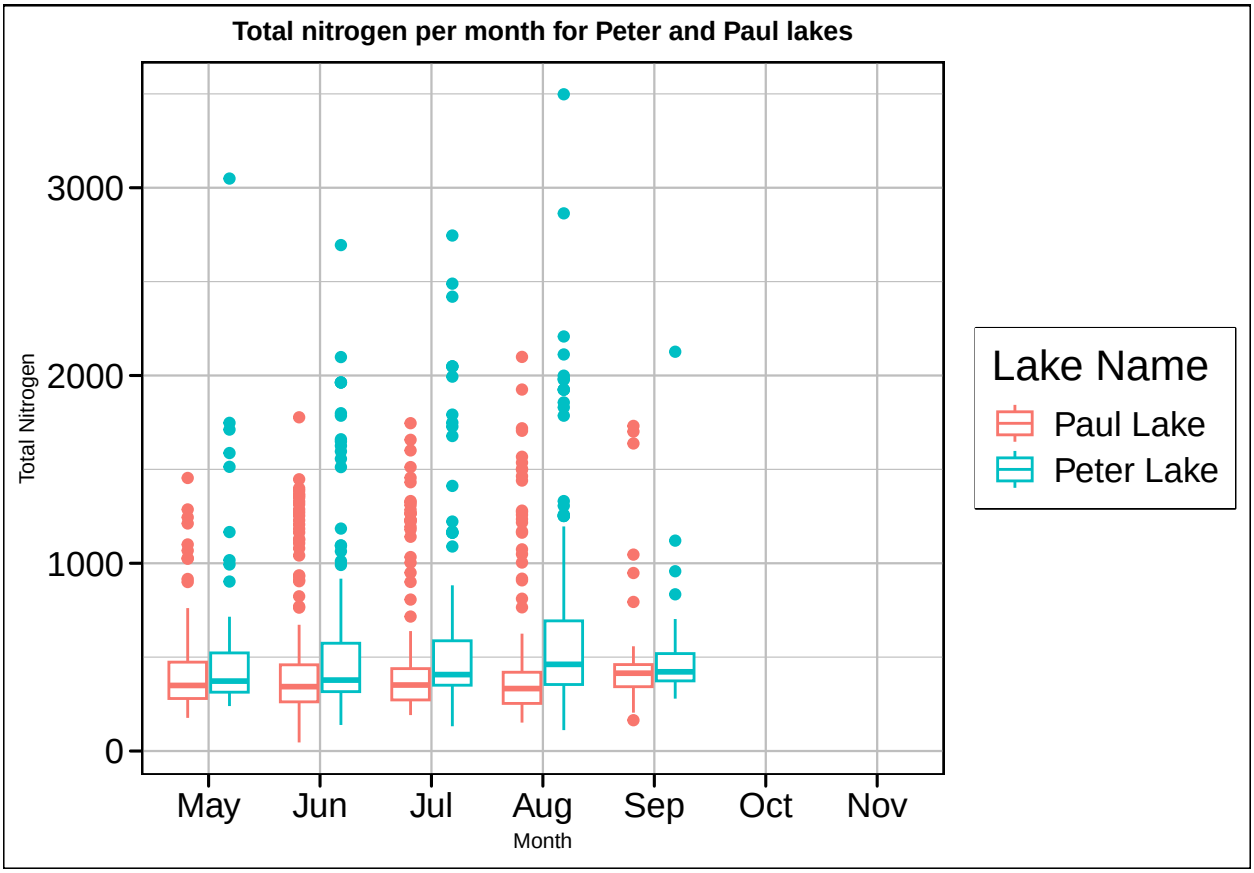
```
## Warning: Removed 20713 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
tn.plot <-
  ggplot(Nutrients, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(
    title="Total nitrogen per month for Peter and Paul lakes",
    x="Month",
    y="Total Nitrogen",
    color="Lake Name") +
  scale_x_discrete(limits = month.abb[5:11]) +
  # Set limit for x axis from May to Nov (for the sake of upcoming combining)
  custom_grid
print(tn.plot)
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 21567 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
library(cowplot)
plot_grid(
  temp.plot + theme(axis.title.x = element_blank(), legend.position = "none"),
  # Removed x axis title and legend
  tp.plot + theme(axis.title.x = element_blank(), legend.position="none"),
  # Removed x axis title and legend
  tn.plot,
  nrow = 3,
  # nrow specifies the number of rows
  align = 'v',
  # align ensures it is aligned vertically 'v' as opposed to horizontally 'h'
  rel_heights = c(0.95, 1.15, 1.5))
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`stat_boxplot()`).
```

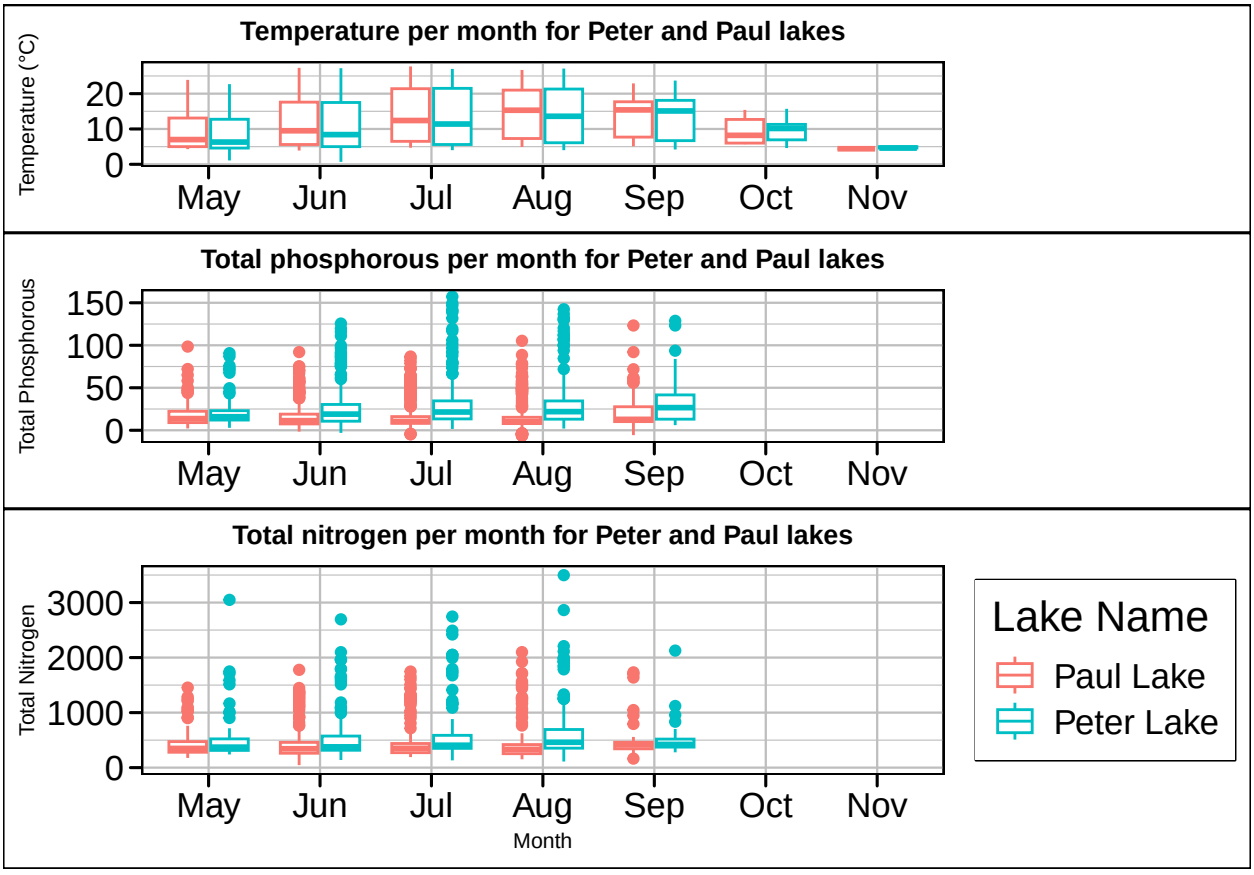
```
## Warning: Removed 3550 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 20713 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`stat_boxplot()`).
```

```
## Warning: Removed 21567 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
  # rel_heights establishes relative size for plots (based on y axis)
```

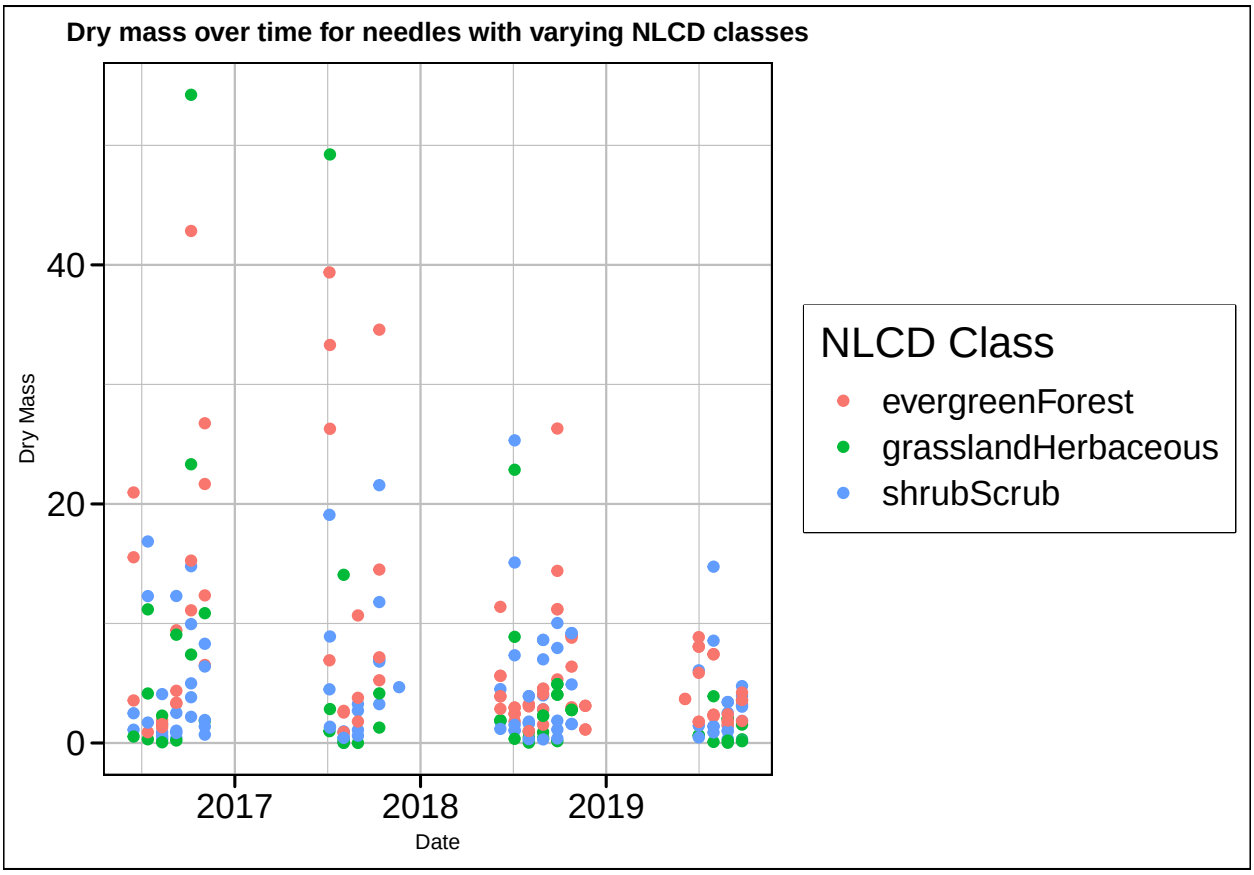
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: While temperature has readings for October and November, the other variables do not. With this information in mind, the three variables can only be compared from May to September. Still, it is important to note these three variables have varying scales (with total nitrogen having the largest and temperature having the smallest). To compare the data, temperature displays a lower median in May which notably rises from June through September before sharply declining in October (which seemingly continues into November but data appears limited) for each lake; total phosphorous displays increasing medians from May through September for Peter Lake but decreasing medians from May through August before increasing in September for Paul Lake; and total nitrogen displays relatively consistent medians (with only slight fluctuations) from May through September for each lake. Based on these observations, I would assume that these three variables are not very highly correlated, but there may be slight influence in temperature on the availability of phosphorous and nitrogen (direct or indirect relationships). More data is required to make any definitive conclusions.

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)

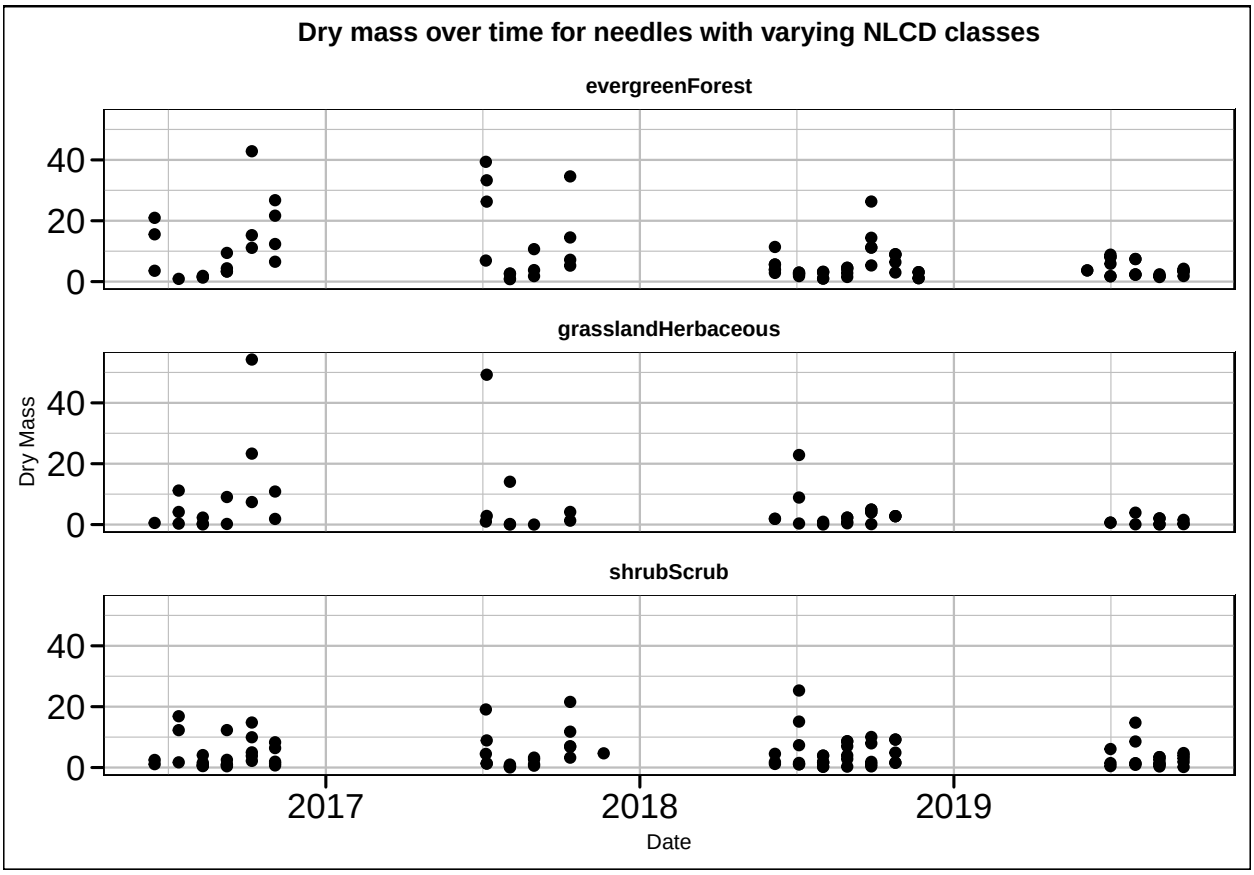
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6: Used ggplot() to plot "collectdate" vs "drymass" with "color=nlcdClass".
needles.massdate.colorNLCD.plot <- Litter %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point() +
  labs(
    title="Dry mass over time for needles with varying NLCD classes",
    x="Date",
    y="Dry Mass",
    color="NLCD Class") +
  custom_grid
print(needles.massdate.colorNLCD.plot)
```




```
#7: Used ggplot() to plot "collectDate" vs "dryMass".
# Adding a facet_wrap() for "nlcdClass" instead of using "color=".
# This change result in the NLCD classes being graphed separately.
needles.massdate.facetNLCD.plot <- Litter %>%
  filter(functionalGroup=="Needles") %>%
  ggplot(aes(x=collectDate, y=dryMass)) +
```

```
  geom_point() +
  facet_wrap(facets=vars(nlcdClass), nrow = 3) +
  labs(
    title="Dry mass over time for needles with varying NLCD classes",
    x="Date",
    y="Dry Mass") +
  custom_grid
print(needles.massdate.facetNLCD.plot)
```



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

Answer: In my opinion, plot 7 (with facets instead of colors) is more effective due to the high number of points plotted when all NLCD classes are on the same graph. With such a high number of points, plot 6 appears crowded (especially below 10 for dry mass where most of the points are concentrated). Such a depiction makes plot 6 more difficult to analyze in this case, but basing colors on a variable may be more useful in other cases; for example, if there are less categories or if the data can be depicted in a bar graph.