Assignment 5: Data Visualization

Eric Newton

Fall 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. 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); library(lubridate); library(here); library(cowplot)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     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
```

```
## here() starts at C:/Users/enewt/OneDrive/Documents/Duke/ENV872/EDE_Fall2023
##
##
## 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'),
  stringsAsFactors = TRUE)

litter <- read.csv(
  here('Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv'),
  stringsAsFactors = TRUE)
#2
litter$collectDate <- ymd(litter$collectDate)
nutrients$sampledate <- ymd(nutrients$sampledate)

#used lubridate to convert to 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

```
#3
mytheme <- theme_classic(base_size = 14)+
  theme(legend.background = element_rect(
    color = "grey",
    fill = "white"),
    plot.title = element_text(hjust = 0.5)
  )

#created a theme, adjusting legend background, font size, and title alignment

theme_set(mytheme)

#set theme for all plots on this Rmarkdown
```

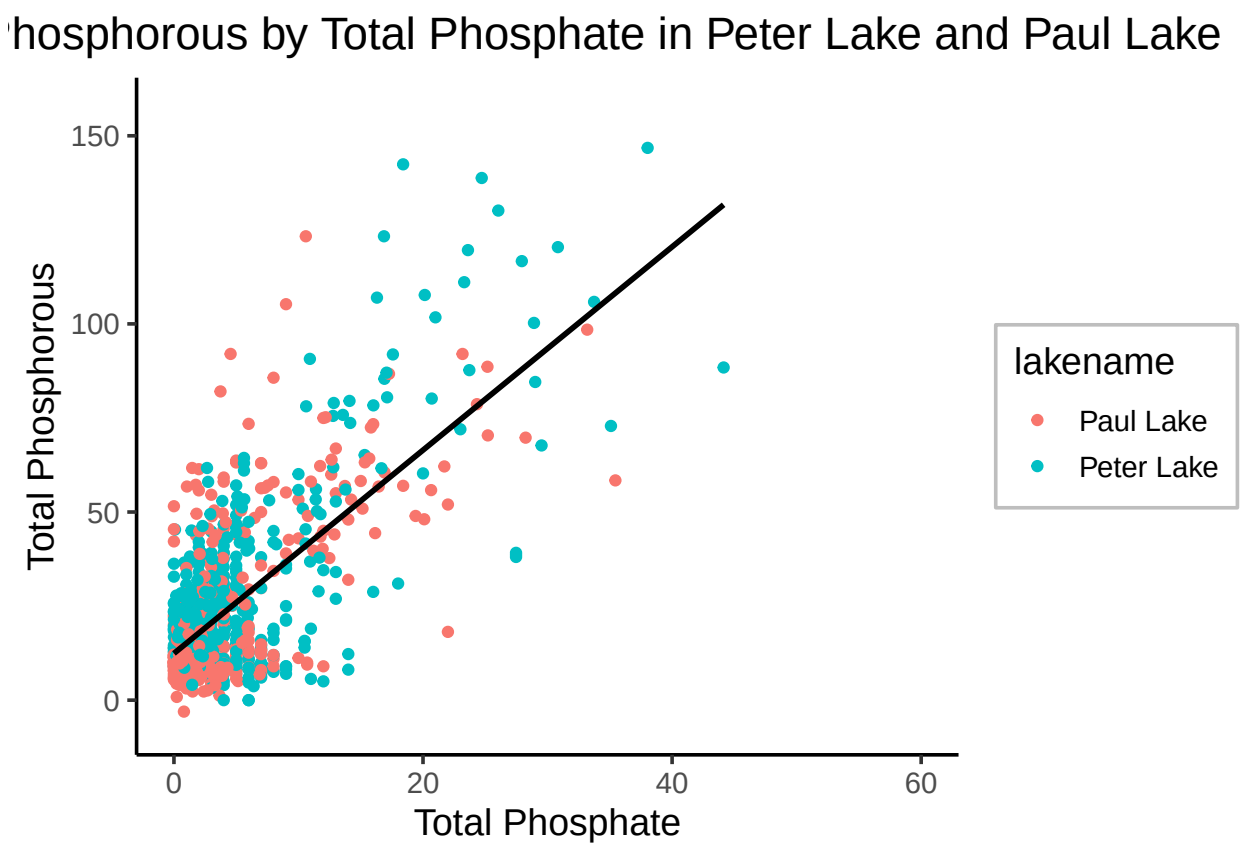
Create graphs

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
phosphorous_by_phosphate <- ggplot(nutrients, aes(
  x=po4,
  y=tp_ug,
  color=lakename))+
geom_point()+
  xlim(0,60)+
geom_smooth(
  method="lm",
  se=FALSE,
  color="black") +
labs(
  title = "Total Phosphorous by Total Phosphate in Peter Lake and Paul Lake",
  x = "Total Phosphate",
  y = "Total Phosphorous")
print(phosphorous_by_phosphate)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 21947 rows containing missing values (`geom_point()`).
```



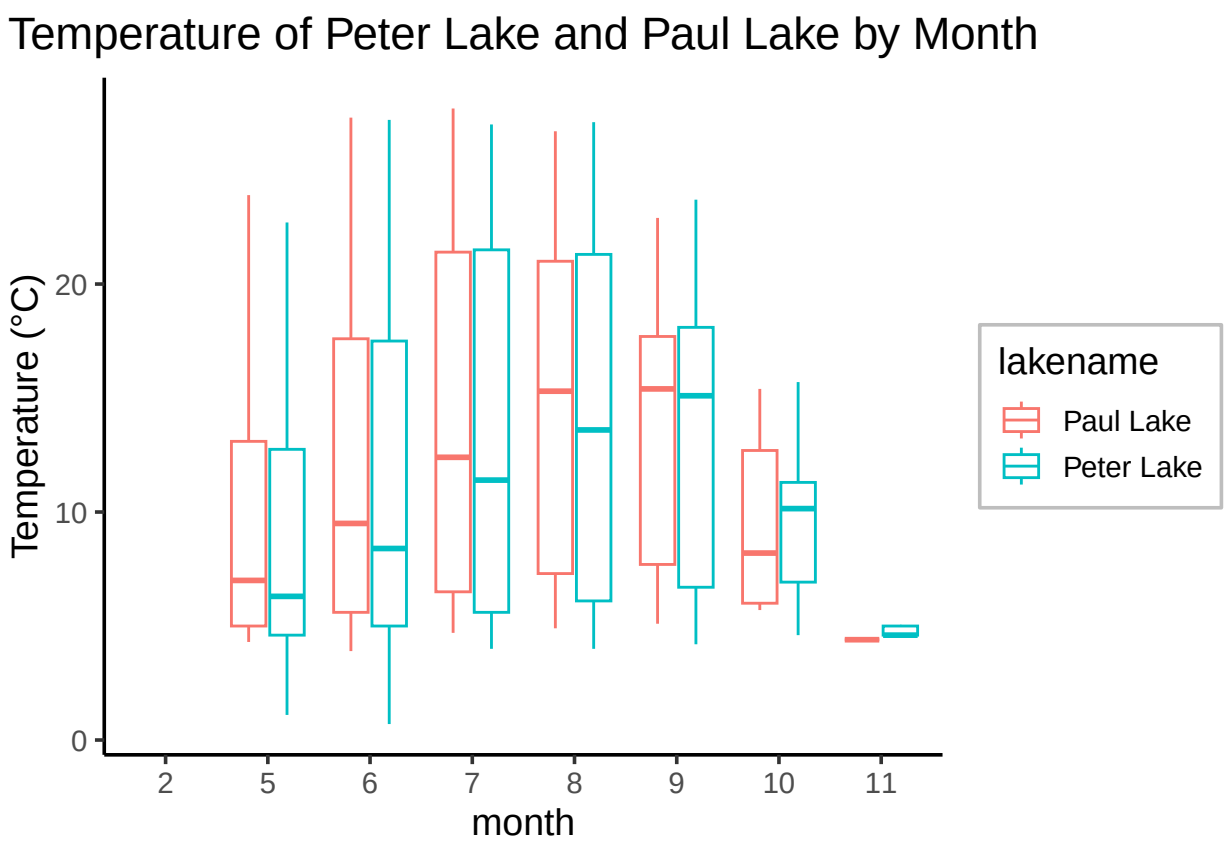
```
#plotted phosphorous by phosphate with lakes separated by color
#added a linear regression line using geom_smooth and set color to black
```

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: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

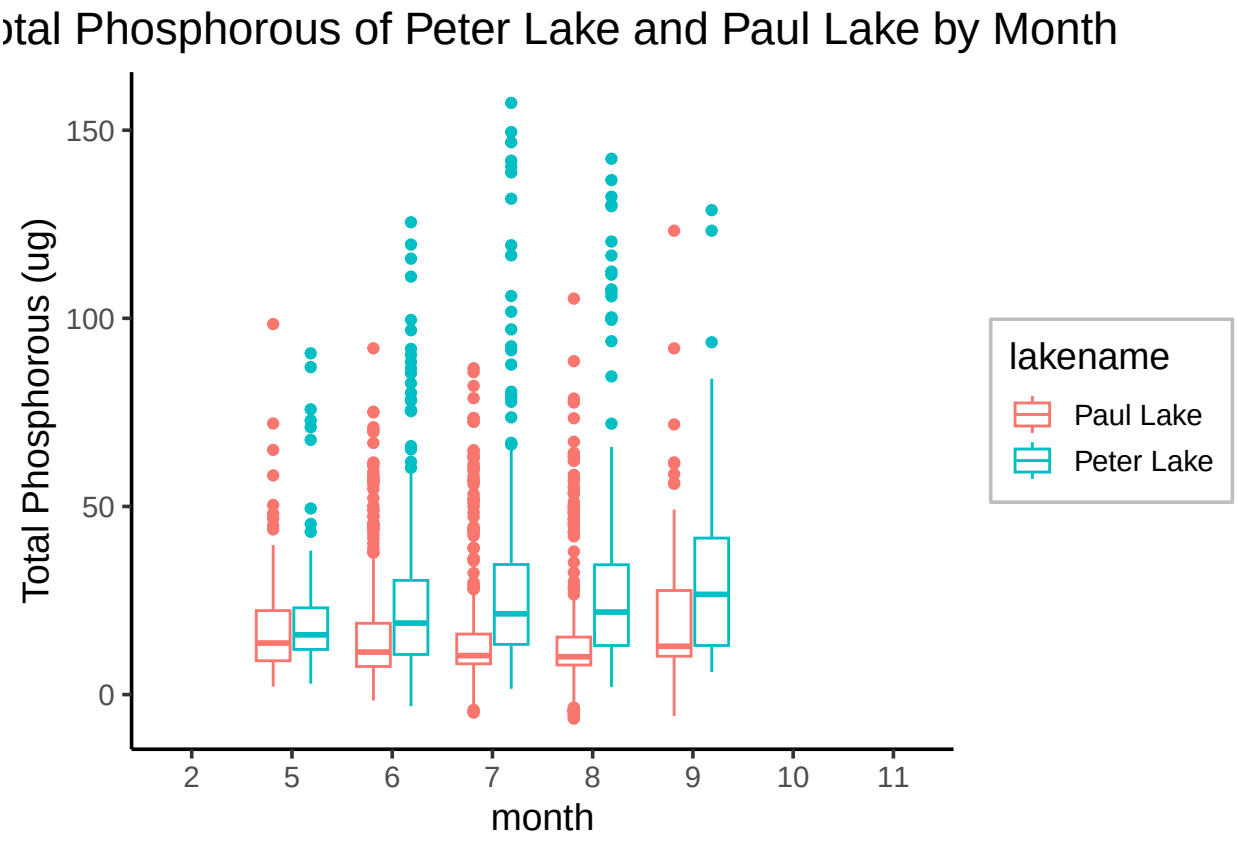
```
#5
nutrients$month <- factor(nutrients$month)

#set month as factor

tempbymonth <- ggplot(nutrients, aes(
  x=month,
  y=temperature_C,
  color=lakename)) +
geom_boxplot() +
  labs(title='Temperature of Peter Lake and Paul Lake by Month',
       y='Temperature (°C)')
print(tempbymonth)
```

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



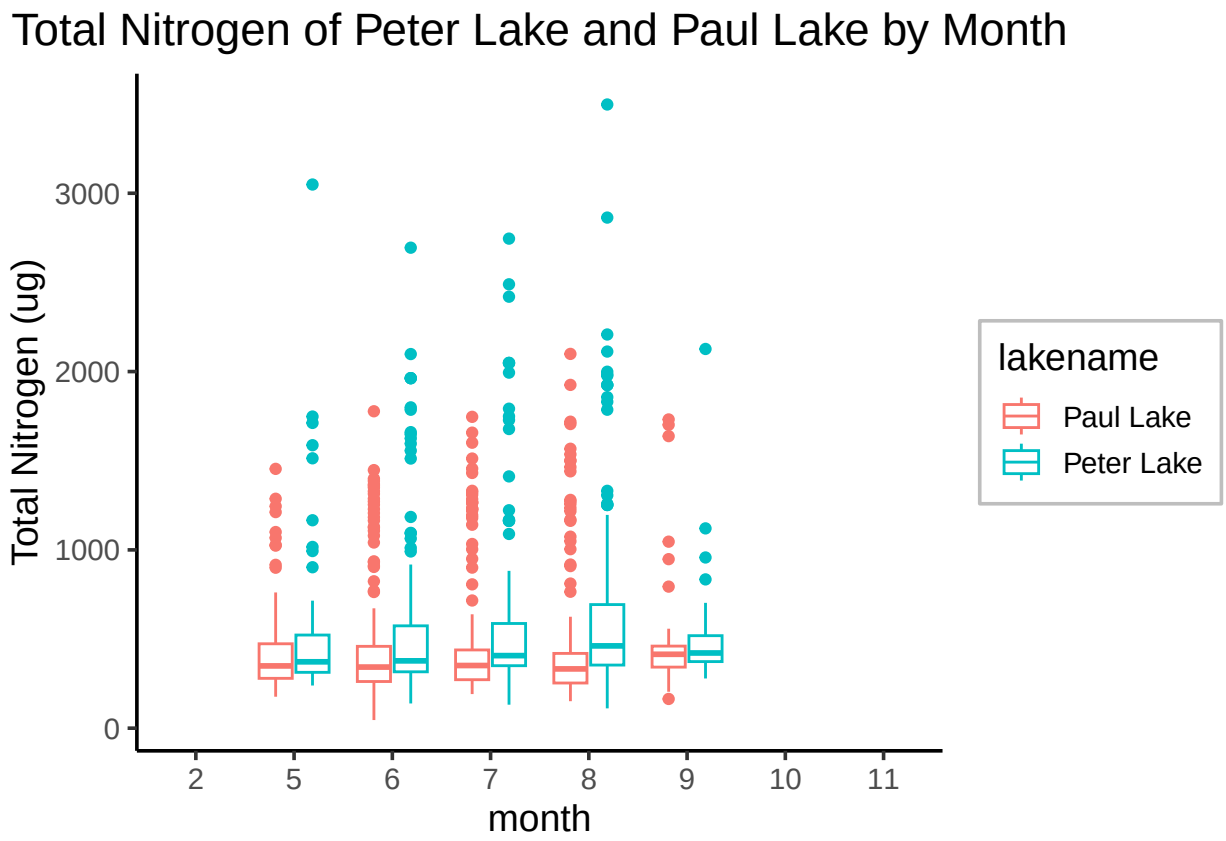
```
TPbymonth <- ggplot(nutrients, aes(
  x=month,
  y=tp_ug,
  color=lakename)) +
geom_boxplot() +
labs(
  title='Total Phosphorous of Peter Lake and Paul Lake by Month',
  y='Total Phosphorous (ug)')
print(TPbymonth)
```

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




```
TNbymonth <- ggplot(nutrients, aes(
  x=month,
  y=tn_ug,
  color=lakename)) +
geom_boxplot() +
labs(
  title='Total Nitrogen of Peter Lake and Paul Lake by Month',
  y='Total Nitrogen (ug)')
print(TNbymonth)
```

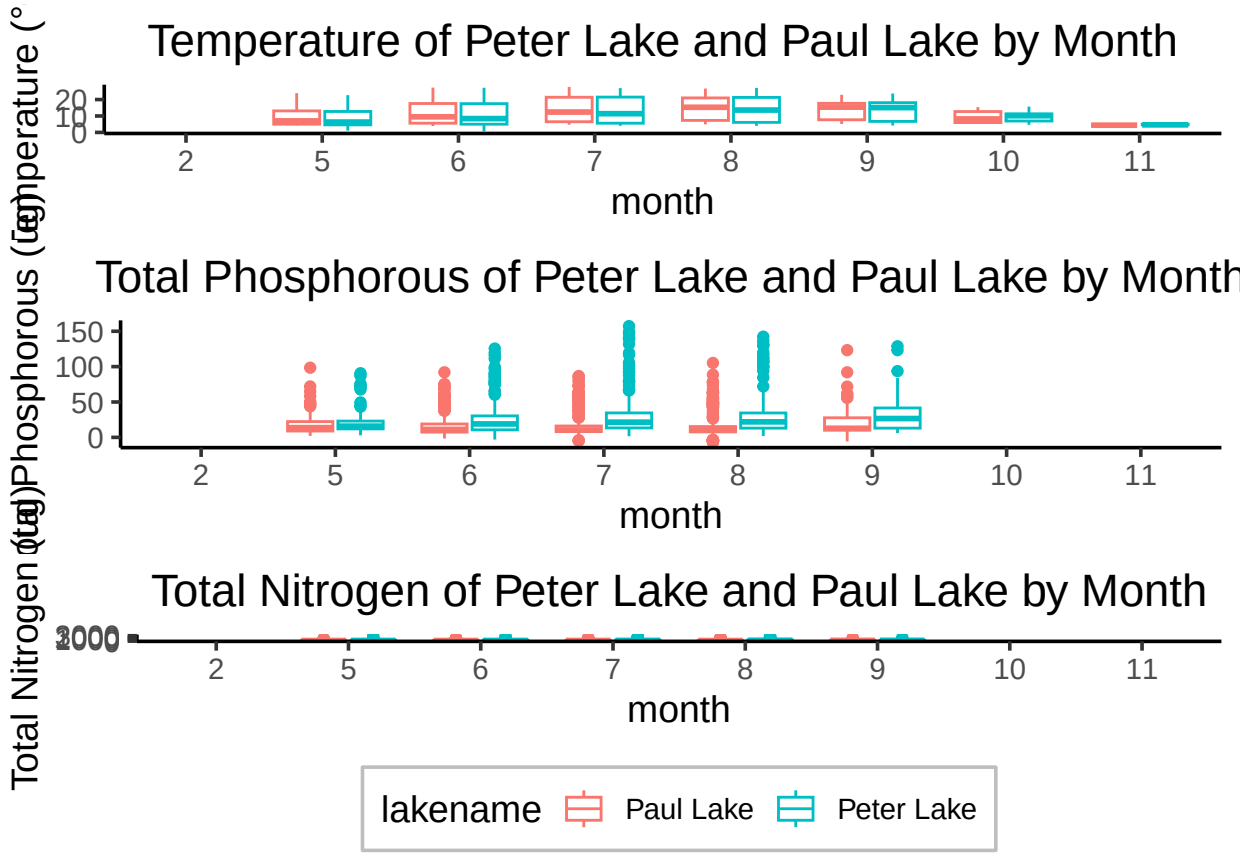
```
## Warning: Removed 21583 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#plotted total nitrogen, phosphorous, and temperature by month
#color set to lakename in aesthetics to differentiate by lake

combined_nutrients <- plot_grid(
  tempbymonth + theme(legend.position="none"),
  TPbymonth + theme(legend.position="none"),
  TNbymonth + theme(legend.position="bottom"),
  nrow = 3,
  rel_heights = c(.75, 1, 1))
```

```
## 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(combined_nutrients)
```




```
#I know that this plot is compressed, but I couldn't figure out how to fix it
#used plot_grid to combine boxplots into one plot
#adjusted relative height to display spread of nutrient plots
#removed legends for first two plots and set legend position to bottom for last
```

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

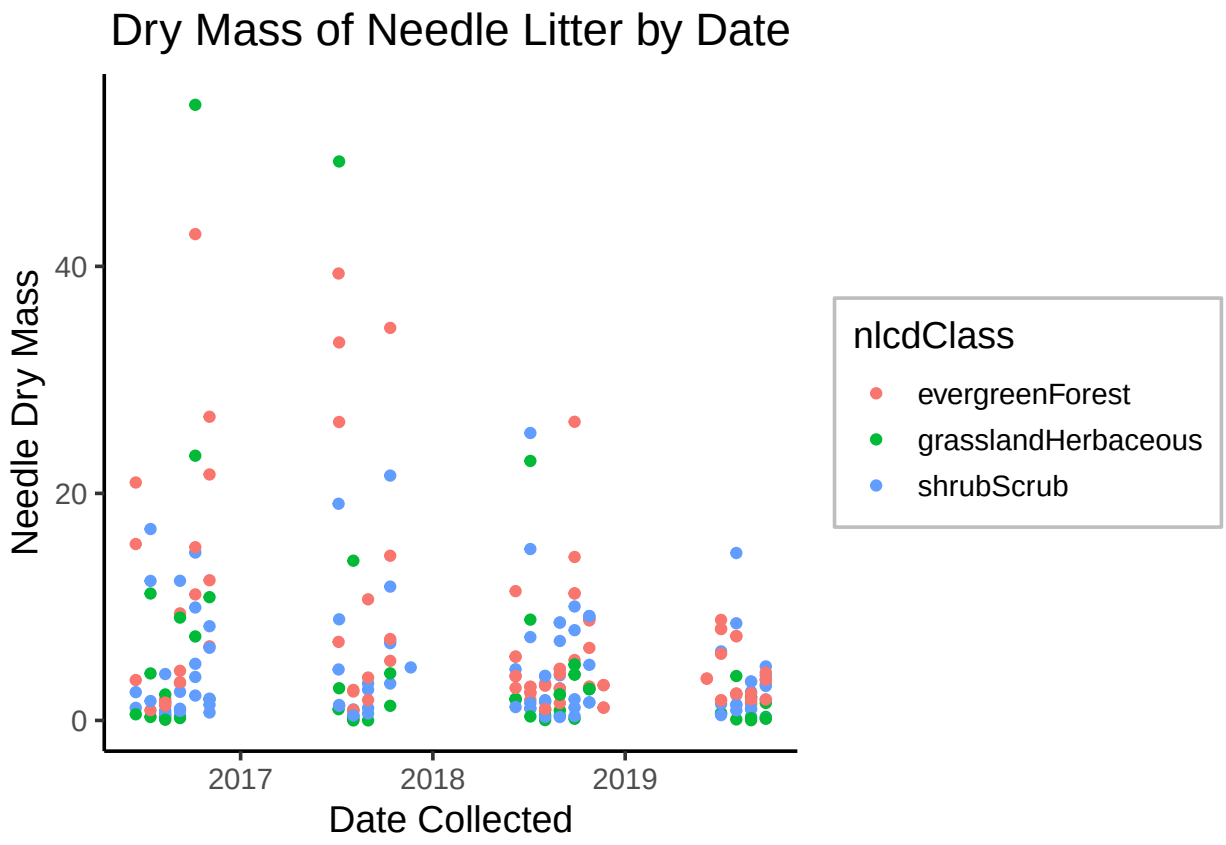
> Answer: Peter Lake has a greater quanitiy of phsophorous and nitrogen compared to Paul Lake, and there is a greater quantity of nitrogen than phosphorous in both lakes. Phosphorous levels increase in Peter Lake over the summer, from May to September, while levels are more variable for nitrogen in Peter Lake.

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.

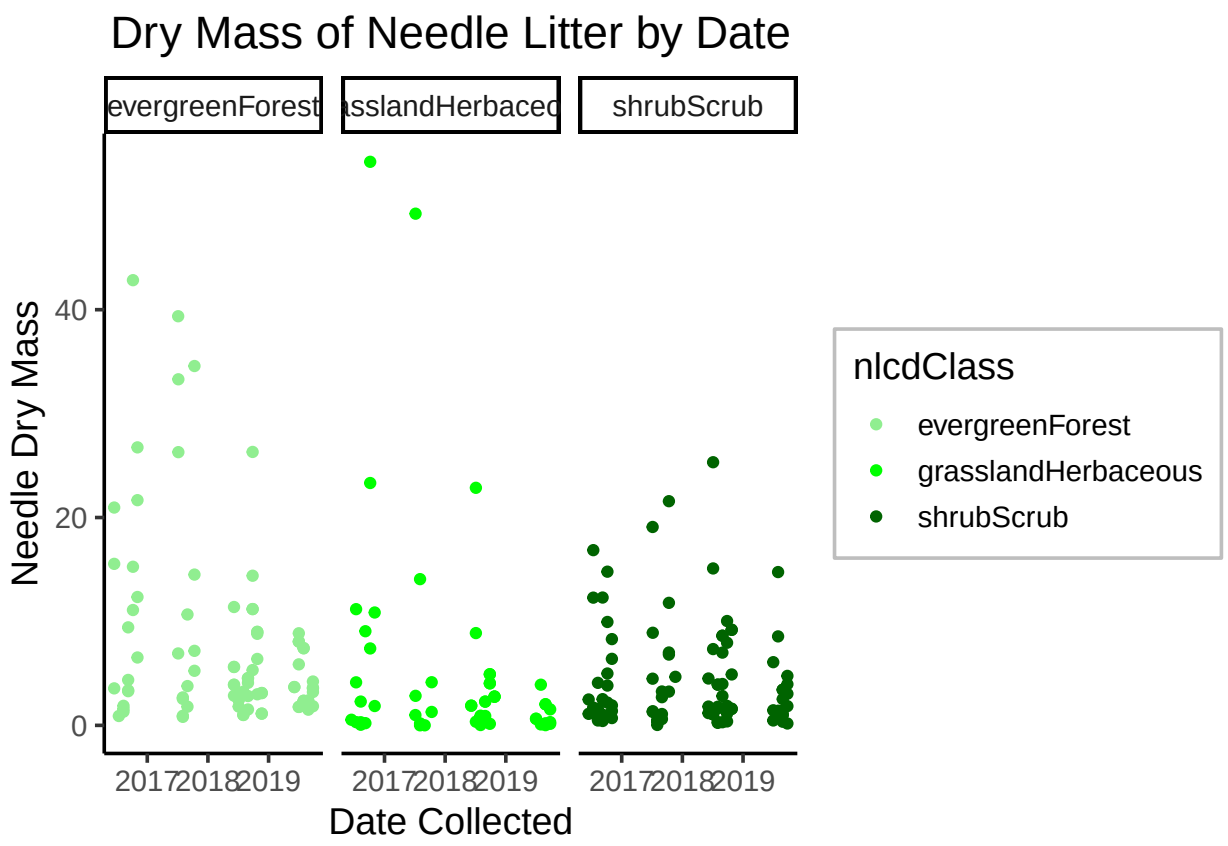
```
#6
needlesbydate <- ggplot(
  litter %>%
  filter(functionalGroup == "Needles")) +
geom_point(aes(
  x=collectDate,
```

```
  y=dryMass,
  color=nlcdClass)) +
labs(title="Dry Mass of Needle Litter by Date",
  x="Date Collected",
  y="Needle Dry Mass")
print(needlesbydate)
```




```
#plotted subset of functional group by filtering column in litter database
#plotted dry mass by date and separated by class of dry mass

#7

needlesbydate_faceted <- ggplot(
  litter %>%
  filter(functionalGroup == "Needles")) +
geom_point(aes(
  x=collectDate,
  y=dryMass,
  color=nlcdClass)) +
labs(title="Dry Mass of Needle Litter by Date",
  x="Date Collected",
  y="Needle Dry Mass") +
facet_wrap(~nlcdClass, ncol = 3) +
  scale_color_manual(
  breaks = c("evergreenForest", "grasslandHerbaceous", "shrubScrub"),
```

```
  values = c("light green", "green", "dark green"))
print(needlesbydate_faceted)
```




```
#separated classes into three facets using facet_Wrap
#set colors to shades of green because they are classes of vegetation
```

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

> Answer: The faceted plot (7) is more effective because you can more clearly see how each class contriubtes to needle litter by given year. In plot 6, the data is not as informative becuase it is difficult to identify points on the plot.