Assignment 5: Data Visualization

Emma Brentjens

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.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 to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

The completed exercise is due on Friday, Oct 14th @ 5:00pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse, lubridate, & cowplot packages. 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_PeterPau` version) and the processed data file for the Niwot Ridge litter dataset (use the [`NEON_NIWO_Litter_mass_trap_Processed` version).

2. Make sure R is reading dates as date format; if not change the format to date.

```r
# 1 finding working directory
getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```r
## loading packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
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(ggplot2)

# uploading data
PeterPaul <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
    stringsAsFactors = T)

Litter <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = T)


# 2 checking date format
class(PeterPaul$sampledate)
```

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

```
class(Litter$collectDate)
```

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

```
## changing date format
PeterPaul$sampledate <- as.Date(PeterPaul$sampledate, format = "%Y-%m-%d")
Litter$collectDate <- as.Date(Litter$collectDate, format = "%Y-%m-%d")

class(PeterPaul$sampledate)
```

```
## [1] "Date"
```

```
class(Litter$collectDate)
```

```
## [1] "Date"
```

Define your theme

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

```
# 3 building default theme
Emma_theme <- theme_linedraw() + theme(axis.text = element_text(color = "black",
    size = 10), legend.position = "right")
```

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()`).

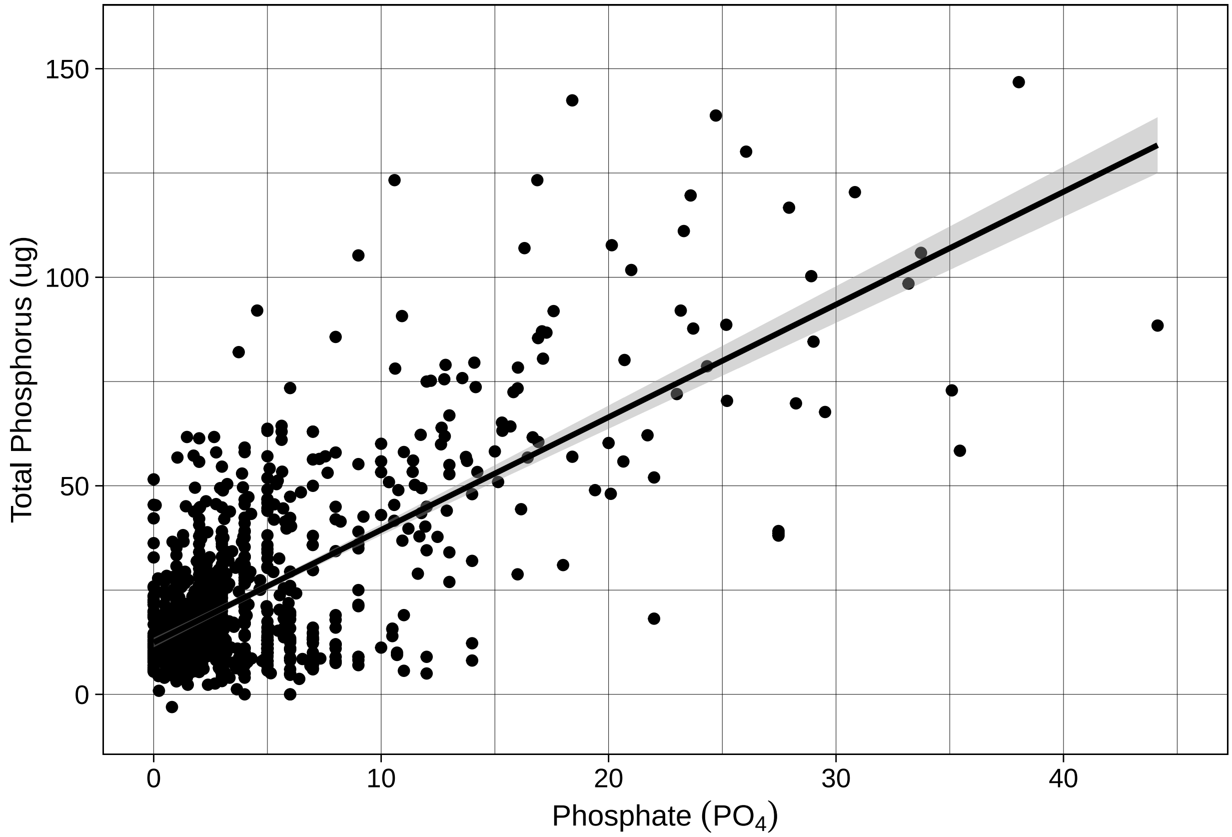
```r
# 4 total phosphorus and phosphate graph
TP_PO4_graph <- ggplot(data = PeterPaul, aes(x = po4, y = tp_ug)) + geom_point() +
    geom_smooth(method = lm, color = "black") + xlim(0, 45) + xlab(expression(`Phosphate `(PO[4]))) +
    ylab("Total Phosphorus (ug)") + Emma_theme

TP_PO4_graph
```

```
## `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).
```



5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and

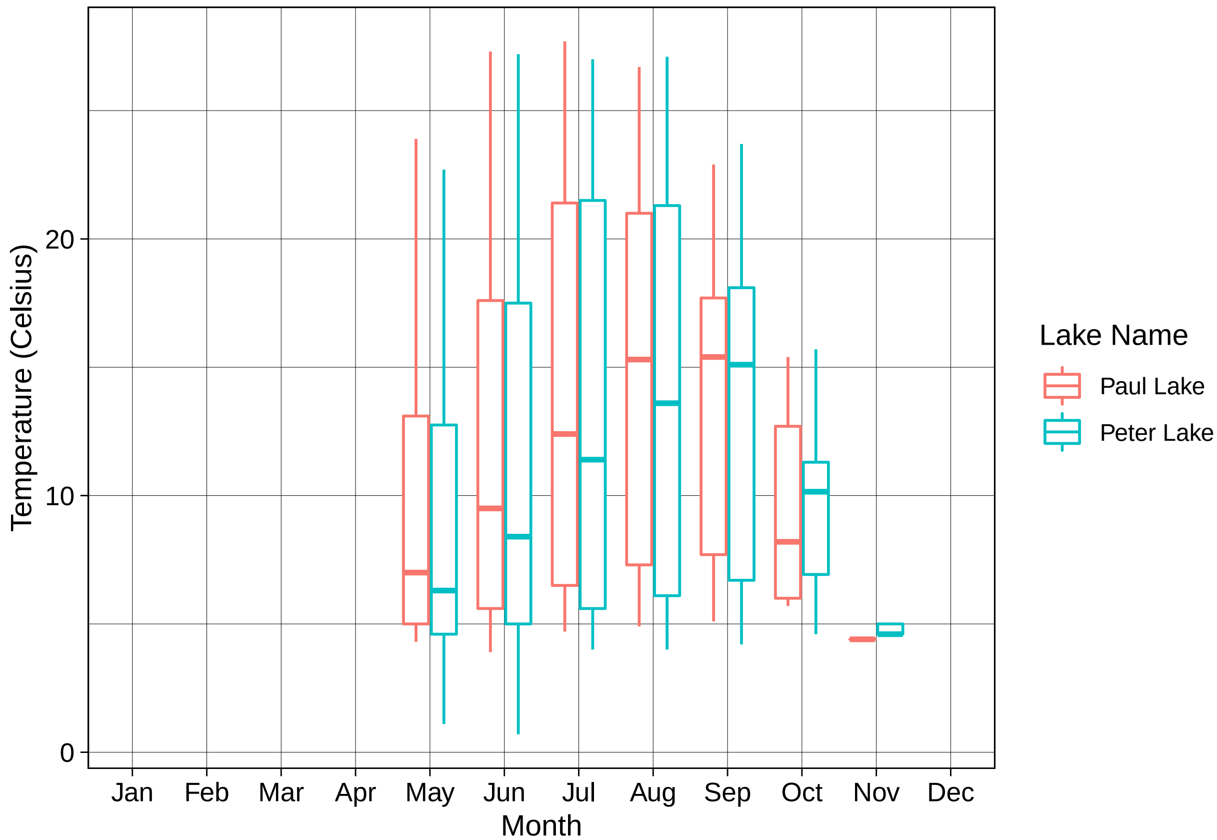
(c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

```r
# 5a temperature boxplot
temp_boxplot <- ggplot(data = PeterPaul, aes(x = factor(month, levels = c(1:12)),
    y = temperature_C)) + geom_boxplot(aes(color = lakename)) + xlab("Month") + ylab("Temperature (Cels
    guides(color = guide_legend(title = "Lake Name")) + scale_x_discrete(labels = month.abb[],
    drop = F) + Emma_theme

temp_boxplot
```

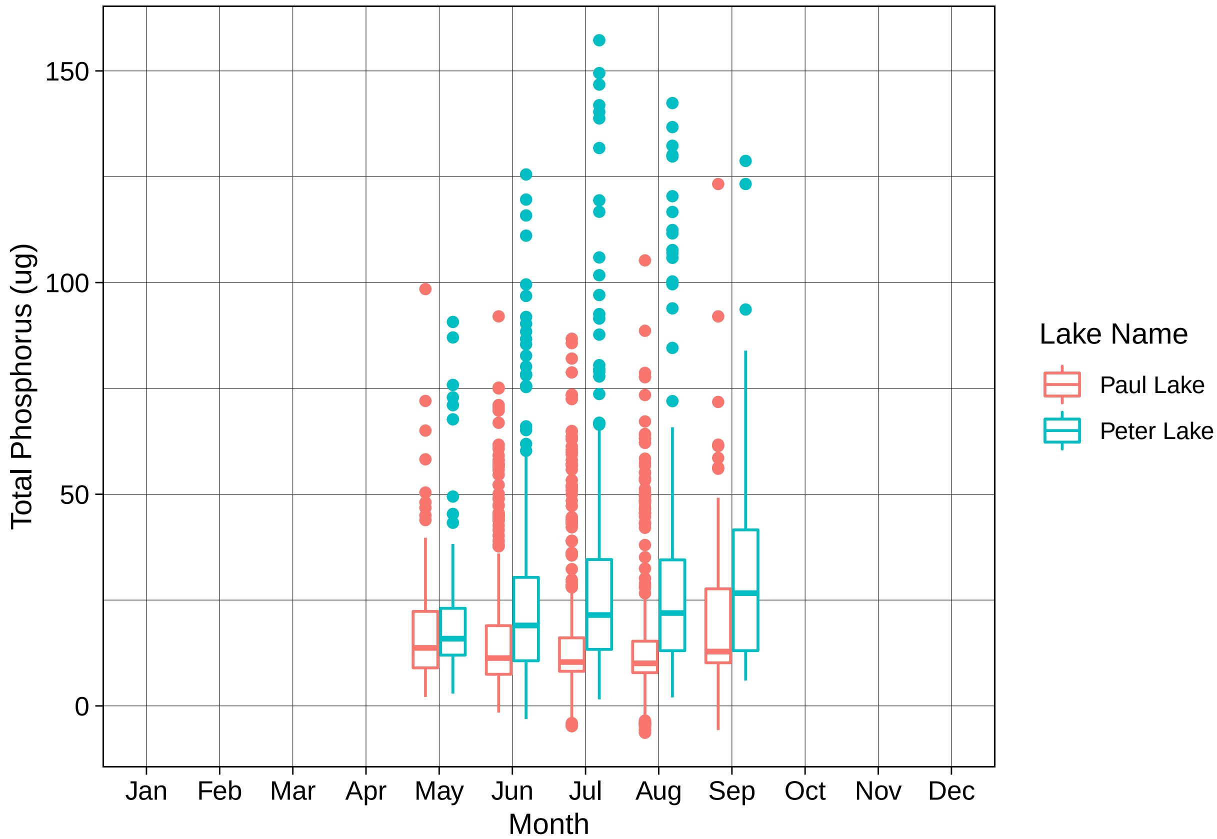
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```




```
# 5b TP boxplot
TP_boxplot <- ggplot(data = PeterPaul, aes(x = factor(month, levels = c(1:12)), y = tp_ug)) +
    geom_boxplot(aes(color = lakename)) + xlab("Month") + ylab("Total Phosphorus (ug)") +
    guides(color = guide_legend(title = "Lake Name")) + scale_x_discrete(labels = month.abb[],
    drop = F) + Emma_theme

TP_boxplot
```

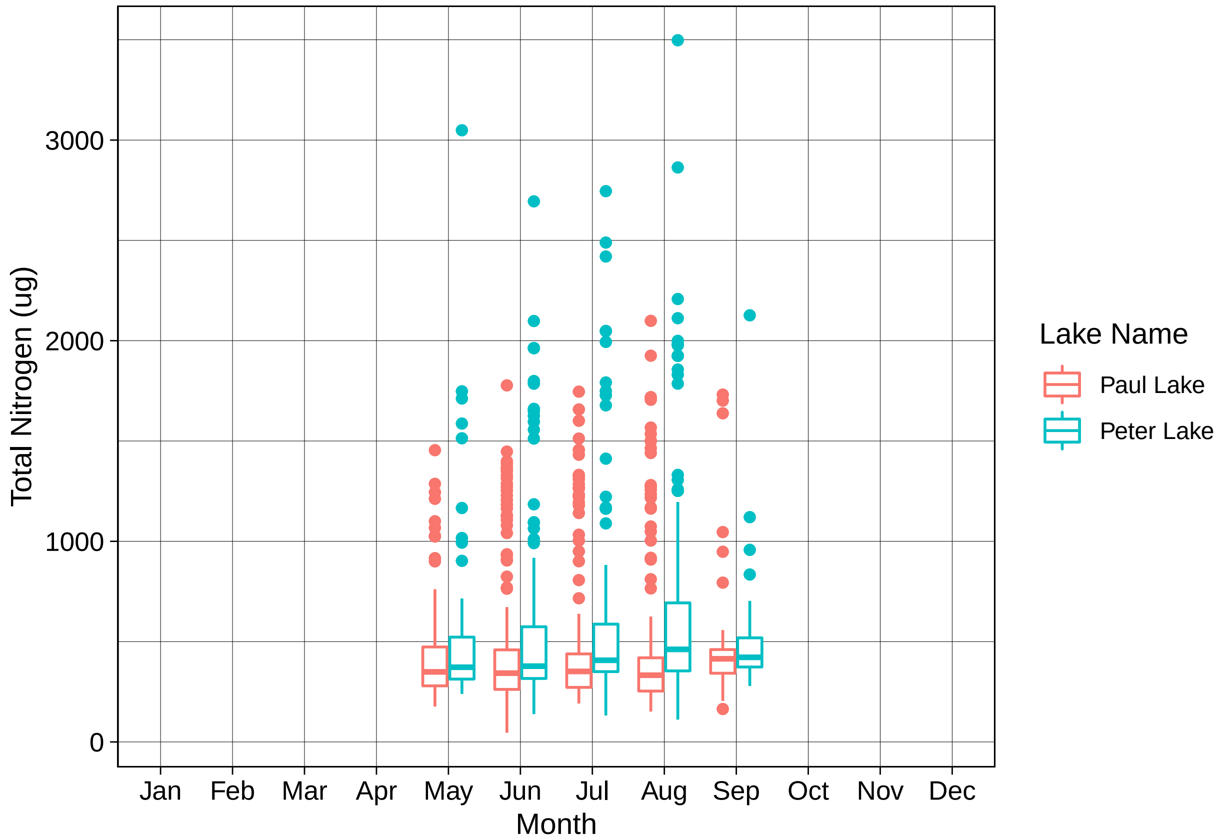
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```




```r
# 5c TN boxplot
TN_boxplot <- ggplot(data = PeterPaul, aes(x = factor(month, levels = c(1:12)), y = tn_ug)) +
    geom_boxplot(aes(color = lakename)) + xlab("Month") + ylab("Total Nitrogen (ug)") +
    guides(color = guide_legend(title = "Lake Name")) + scale_x_discrete(labels = month.abb[],
    drop = F) + Emma_theme

TN_boxplot
```

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




```
# 5d cowplot
PeterPaul_cowplot <- plot_grid(temp_boxplot + theme(legend.position = "none", axis.text = element_text(s
    axis.title.x = element_blank(), axis.title.y = element_text(size = 8)), TP_boxplot +
    theme(legend.position = "none", axis.text = element_text(size = 8), axis.title.x = element_blank(),
        axis.title.y = element_text(size = 8)), TN_boxplot + theme(legend.position = "bottom",
    axis.text = element_text(size = 8), axis.title.x = element_text(size = 10), axis.title.y = element_t
    nrow = 3, align = "vh", rel_heights = c(1.6, 1.7, 2.3))
```

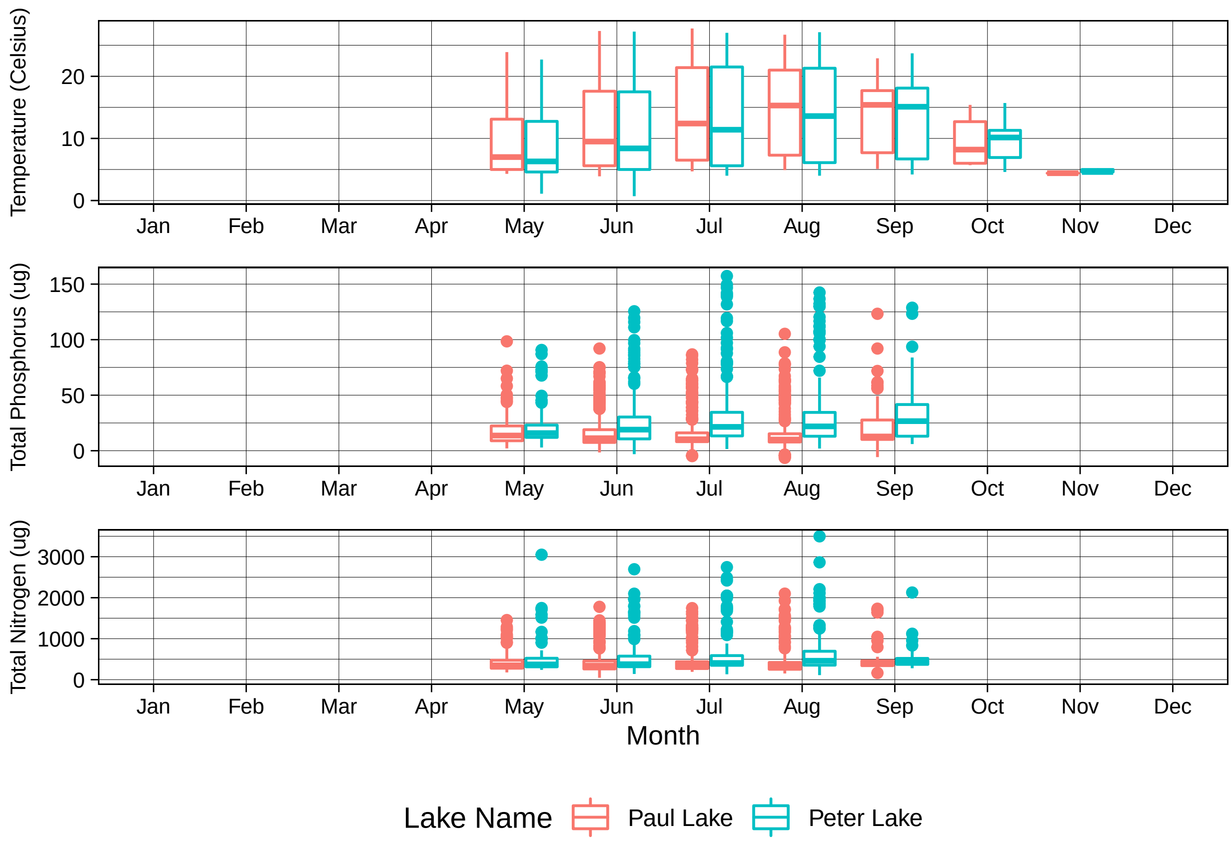
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).

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

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

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.
## Placing graphs unaligned.
```

```
PeterPaul_cowplot
```



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

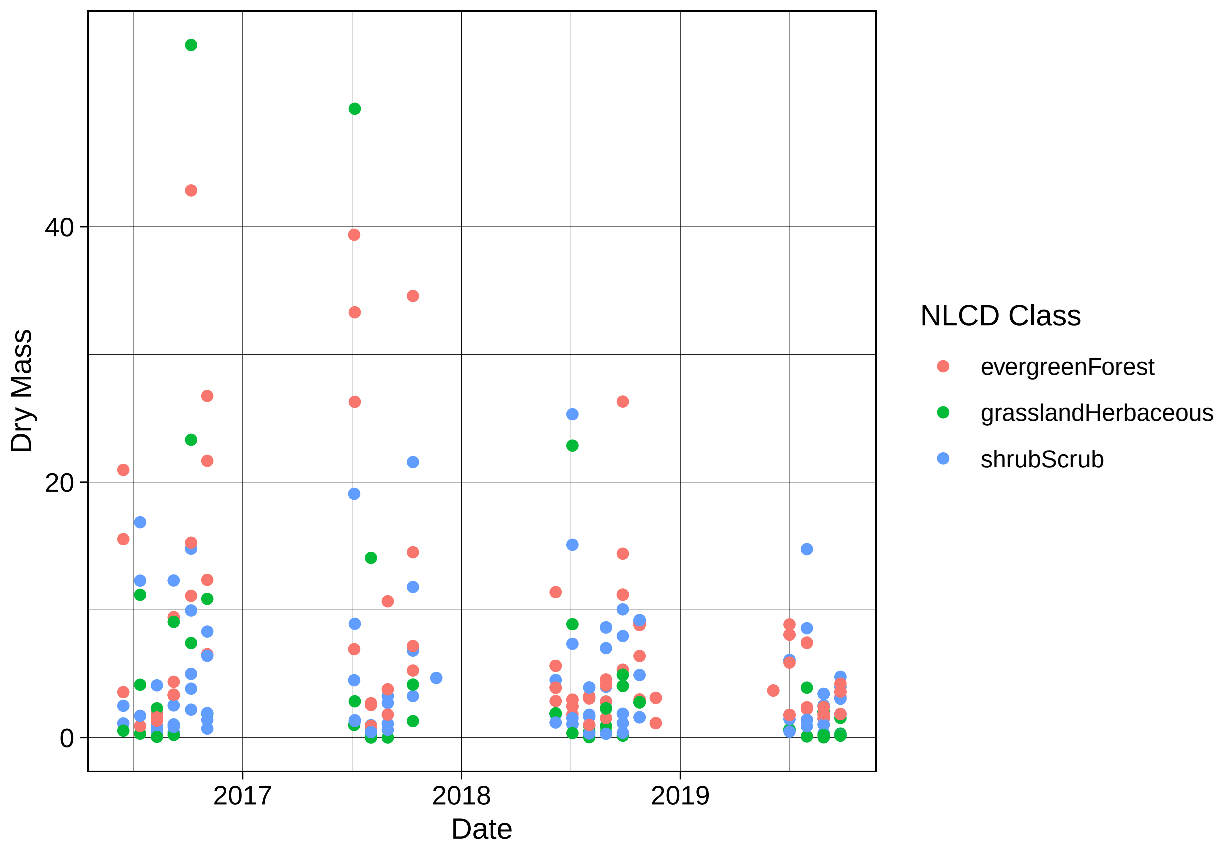
> Answer: Temperature generally seems to be highest in August. TP is highest in September and generally higher at Peter Lake than Paul Lake. TN does not seem to vary as much seasonally as temperature and TP, but is also generally a bit higher in August and at Peter Lake.

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.

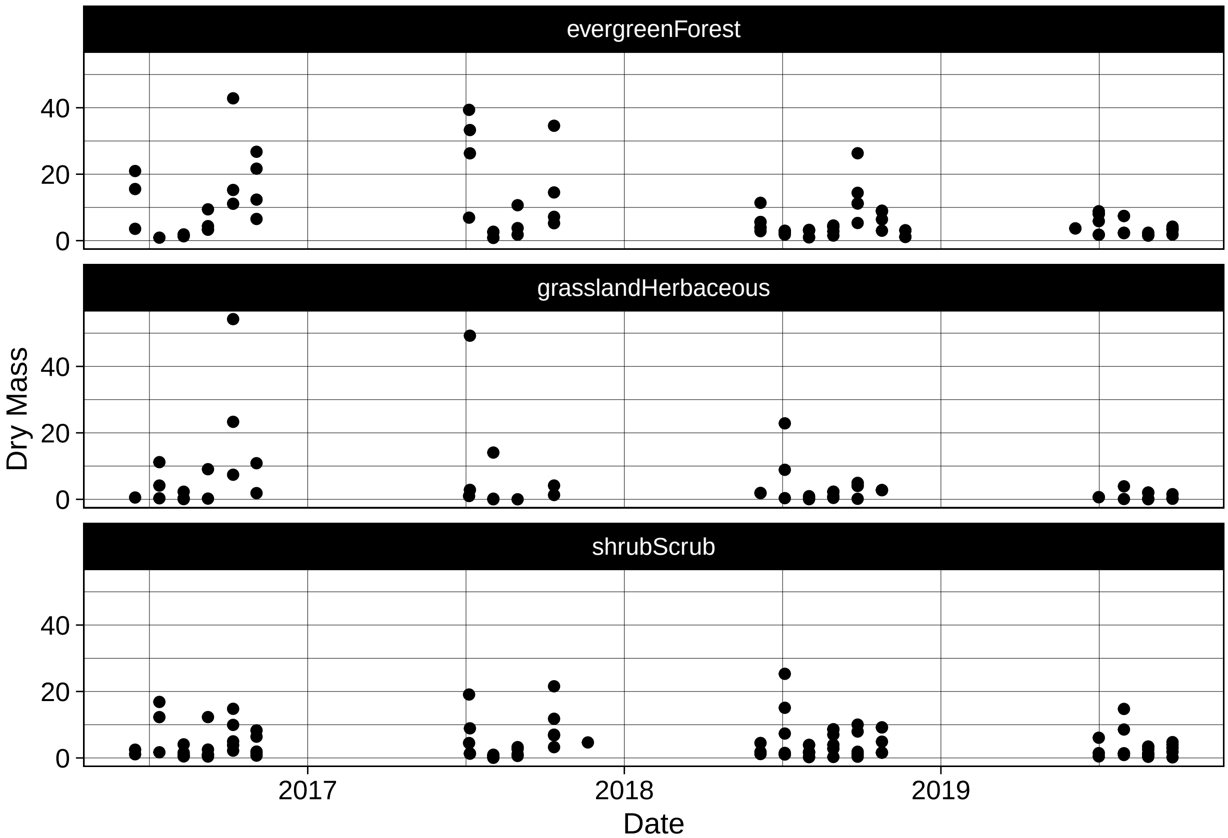
```
# 6
needles_plot <- ggplot(data = subset(Litter, functionalGroup == "Needles"), aes(x = collectDate,
    y = dryMass)) + geom_point(aes(color = nlcdClass)) + scale_x_date(date_breaks = "1 year",
    date_labels = "%Y", limits = as.Date(c("2016-06-16", "2019-09-25"))) + xlab("Date") +
    ylab("Dry Mass") + guides(color = guide_legend(title = "NLCD Class")) + Emma_theme

needles_plot
```




```r
# 7
needles_plot_facet <- ggplot(data = subset(Litter, functionalGroup == "Needles"),
    aes(x = collectDate, y = dryMass)) + geom_point() + scale_x_date(date_breaks = "1 year",
    date_labels = "%Y", limits = as.Date(c("2016-06-16", "2019-09-25"))) + xlab("Date") +
    ylab("Dry Mass") + guides(color = guide_legend(title = "NLCD Class")) + Emma_theme +
    facet_wrap(vars(nlcdClass), nrow = 3)

needles_plot_facet
```



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

Answer: I think the plot from question 7 is more effective because it allows you to observe the trends of dry mass in each three land cover classes while some points are obscured by others in the first graph.