SamanthaSedar>_A05_DataVisualization.Rmd

Samantha Sedar

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); here()
```

```
## -- 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 /home/guest/EDE_Fall2023
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp

## [1] "/home/guest/EDE_Fall2023"
```

```
PeterPaul <-
  read.csv(
    "~/EDE_Fall2023/Data/Processed/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.
    stringsAsFactors = TRUE)
Niwot <- read.csv(
  "~/EDE_Fall2023/Data/Processed/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv",
  stringsAsFactors = TRUE)

#2
Niwot$collectDate <- ymd(Niwot$collectDate)
PeterPaul$sampledate <- ymd(PeterPaul$sampledate)
```

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
#Build theme using a light blue background, dark blue title, with the legend position at the
#bottom of the plot

mytheme <- theme(
  plot.background = element_rect(fill = "lightblue"),
  plot.title = element_text(color = "darkblue"),
  legend.position = "bottom")

theme_set(mytheme)
```

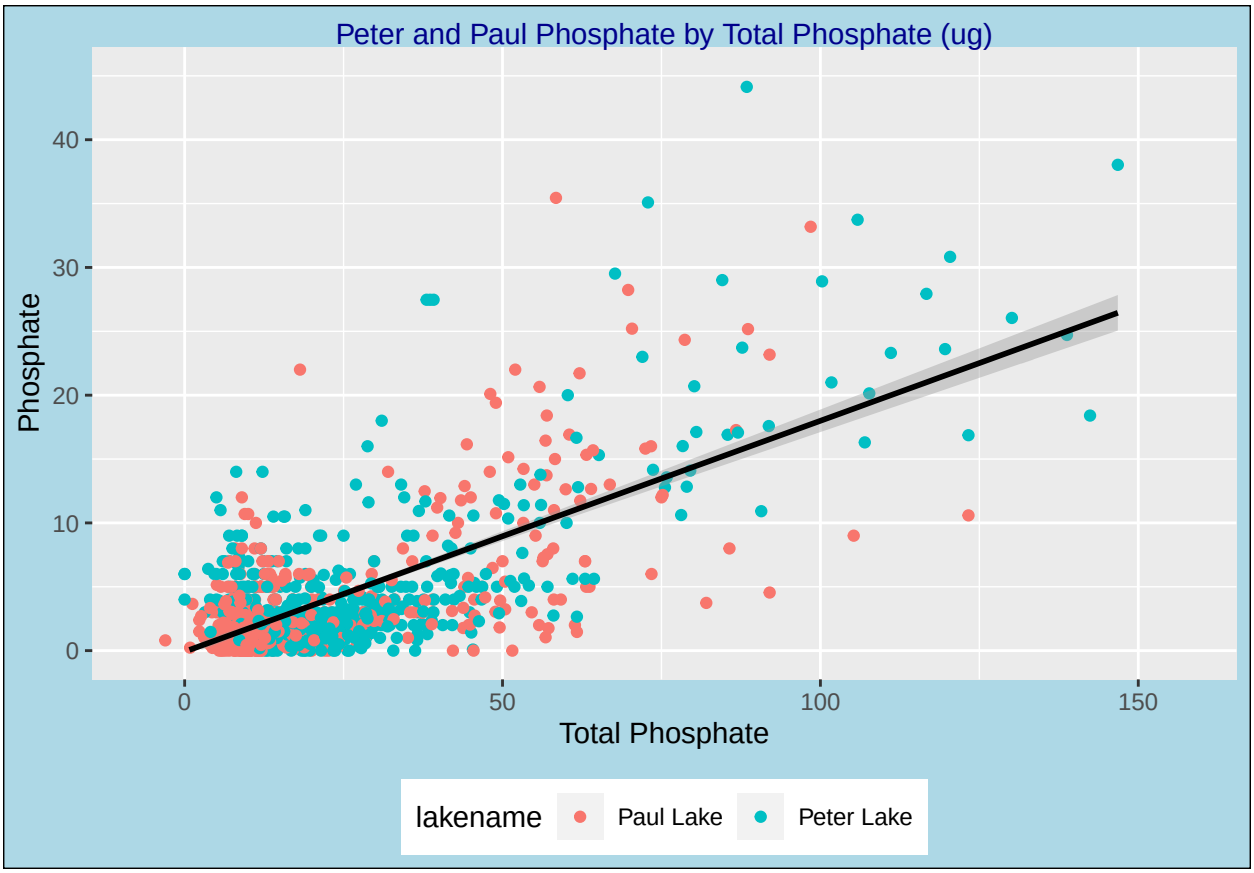
Create graphs

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 Plot, setting y as po4 and x as tp_ug, adding limits to x in order to view clearer trends within the

peter_paul_plot <- ggplot(PeterPaul, aes(x=tp_ug,
                          y=po4,
                          color=lakename
                          ))+
  ylim(0, 45)+
  geom_point()+
  geom_smooth(method = lm, color="black")+
  labs(title = "Peter and Paul Phosphate by Total Phosphate (ug)", x = "Total Phosphate",
       y = "Phosphate")

print(peter_paul_plot)
```

```
## `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()`).

## Warning: Removed 2 rows containing missing values (`geom_smooth()`).
```

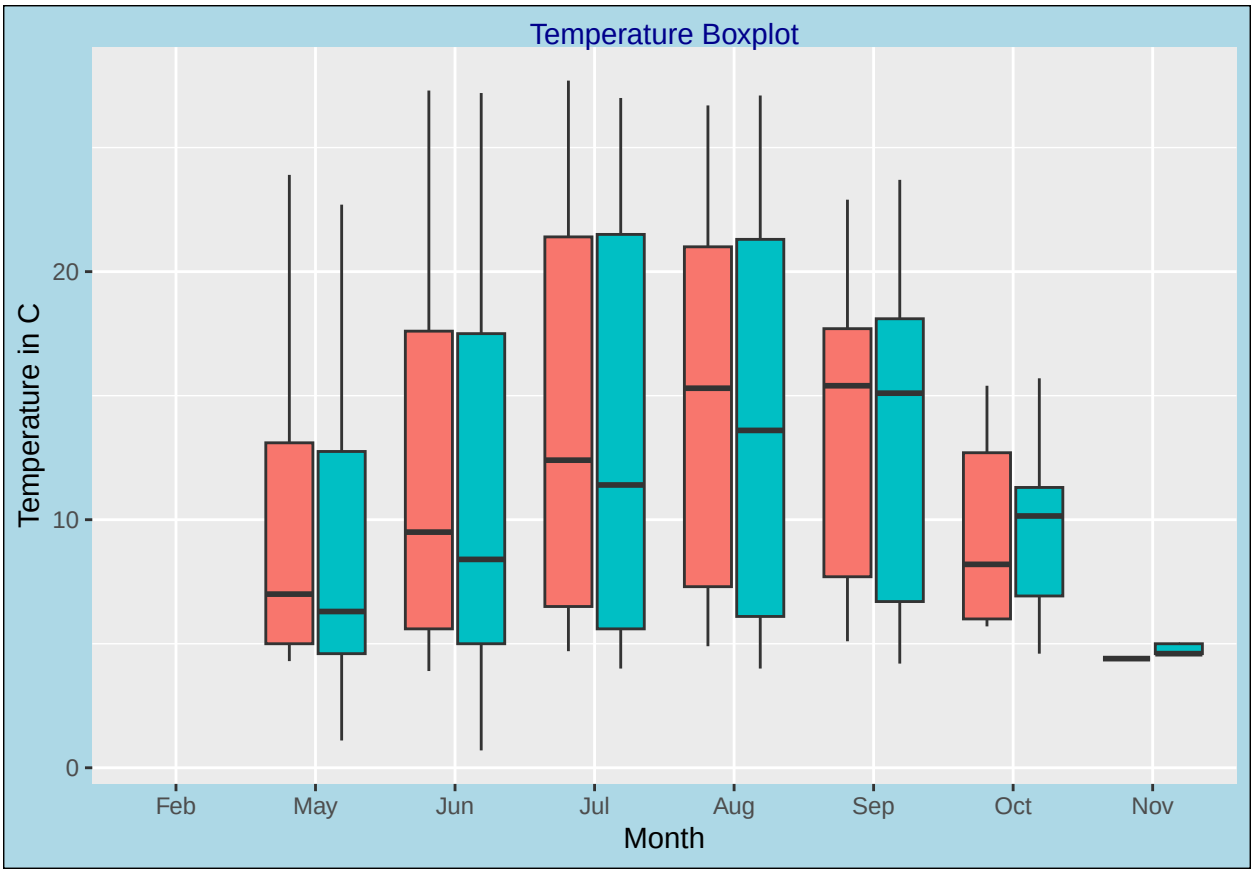


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

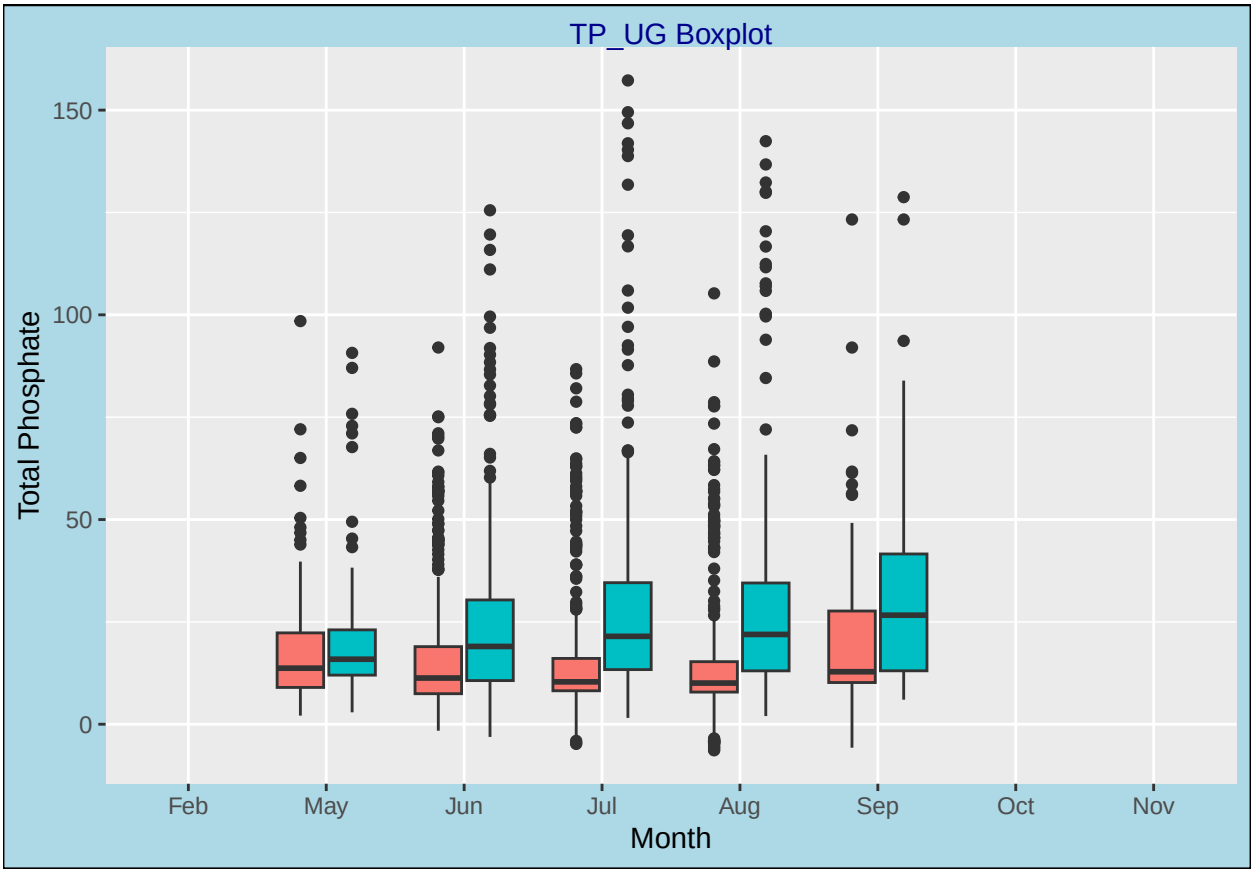
```
#5 Created plots for temp, TP, and TN, setting labels to months and specifying levels 1:12 to indicate
#there are 12 months.

#A- temperature, choosing fill to better visualize the difference between Peter and Paul lakes
temp <- ggplot(PeterPaul,
                aes(x=factor(month,
                    levels=1:12,
                    labels=month.abb),
                    y=temperature_C,
                    fill=lakename))+
  geom_boxplot() +
  scale_x_discrete(name="Month")+
  theme(legend.position = "none")+
  labs(title = "Temperature Boxplot", x = "Month", y = "Temperature in C")

print(temp)
```

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



```
#B - TP Plot
TP <- ggplot(PeterPaul,
               aes(x=factor(month,
                    levels=1:12,
                    labels=month.abb),
                    y=tp_ug,
                    fill=lakename))+
  geom_boxplot() +
  scale_x_discrete(name="Month")+
  theme(legend.position = "none")+
  labs(title = "TP_UG Boxplot", x = "Month", y = "Total Phosphate")

print(TP)
```

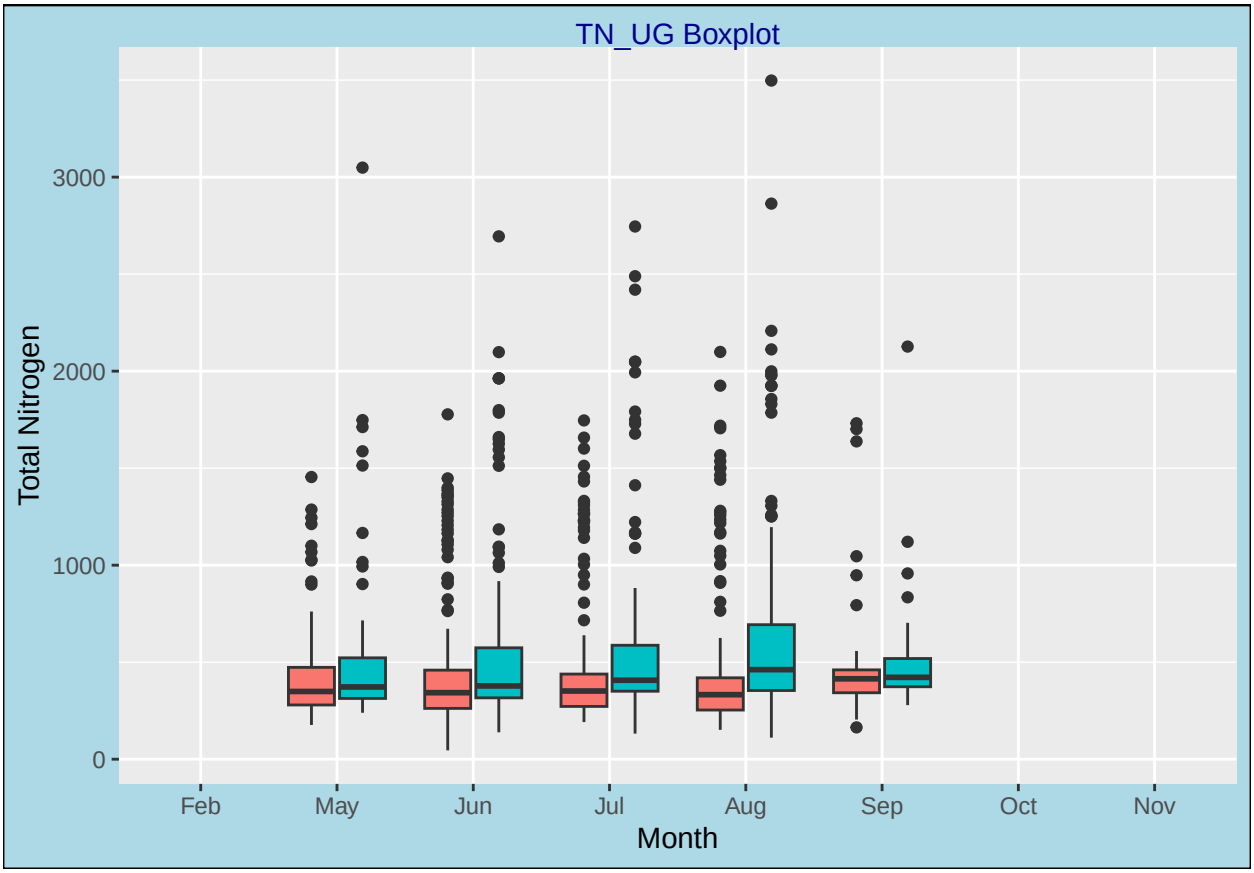
```
## Warning: Removed 20729 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#C - TN Plot
TN <- ggplot(PeterPaul,
               aes(x=factor(month,
                    levels=1:12,
                    labels=month.abb),
                    y=tn_ug,
                    fill=lakename))+
  geom_boxplot() +
  scale_x_discrete(name="Month")+
```

```
  theme(legend.position = "none")+
  labs(title = "TN_UG Boxplot", x = "Month", y = "Total Nitrogen")

print(TN)
```

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



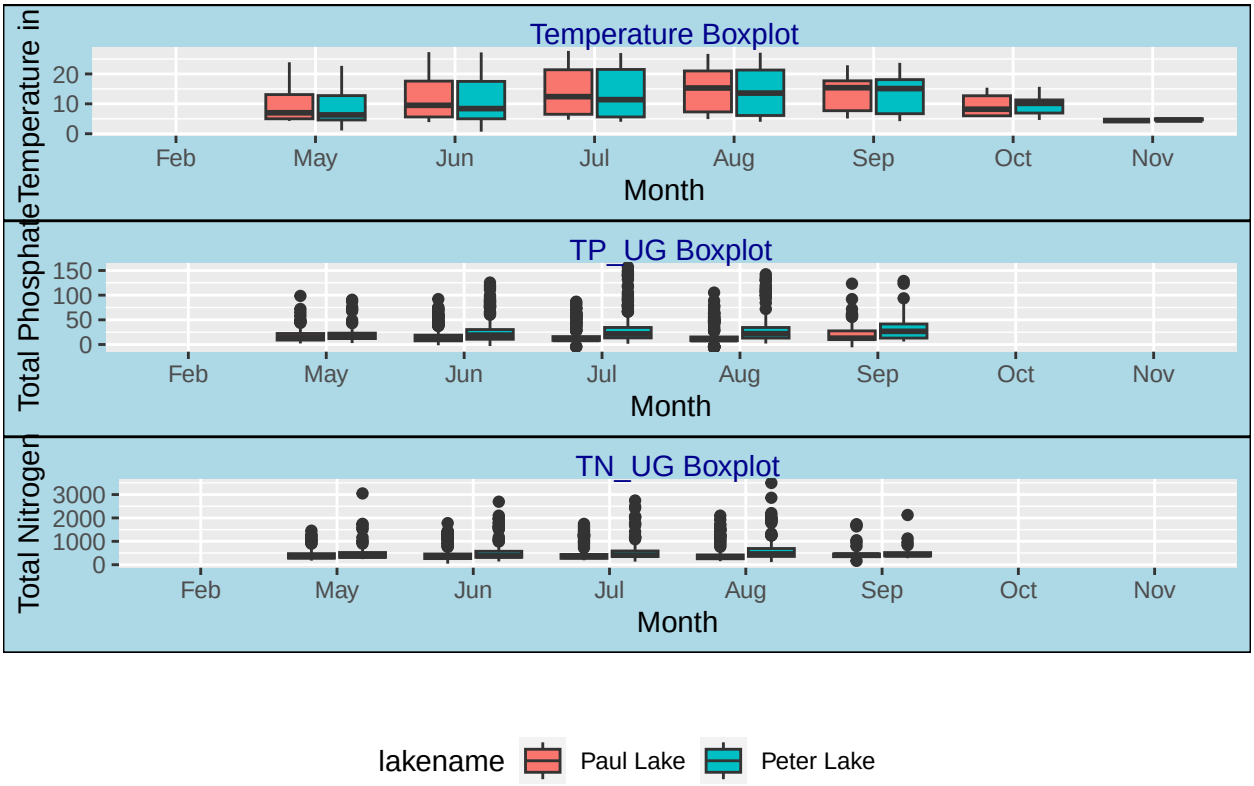

```
#legend made using: https://wilkelab.org/cowplot/articles/shared_legends.html --- used months as axes

legend <- get_legend(temp +
              theme(legend.position = "bottom"))
```

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

```
summary <- plot_grid(temp, TP, TN, legend, ncol = 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(summary)
```



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

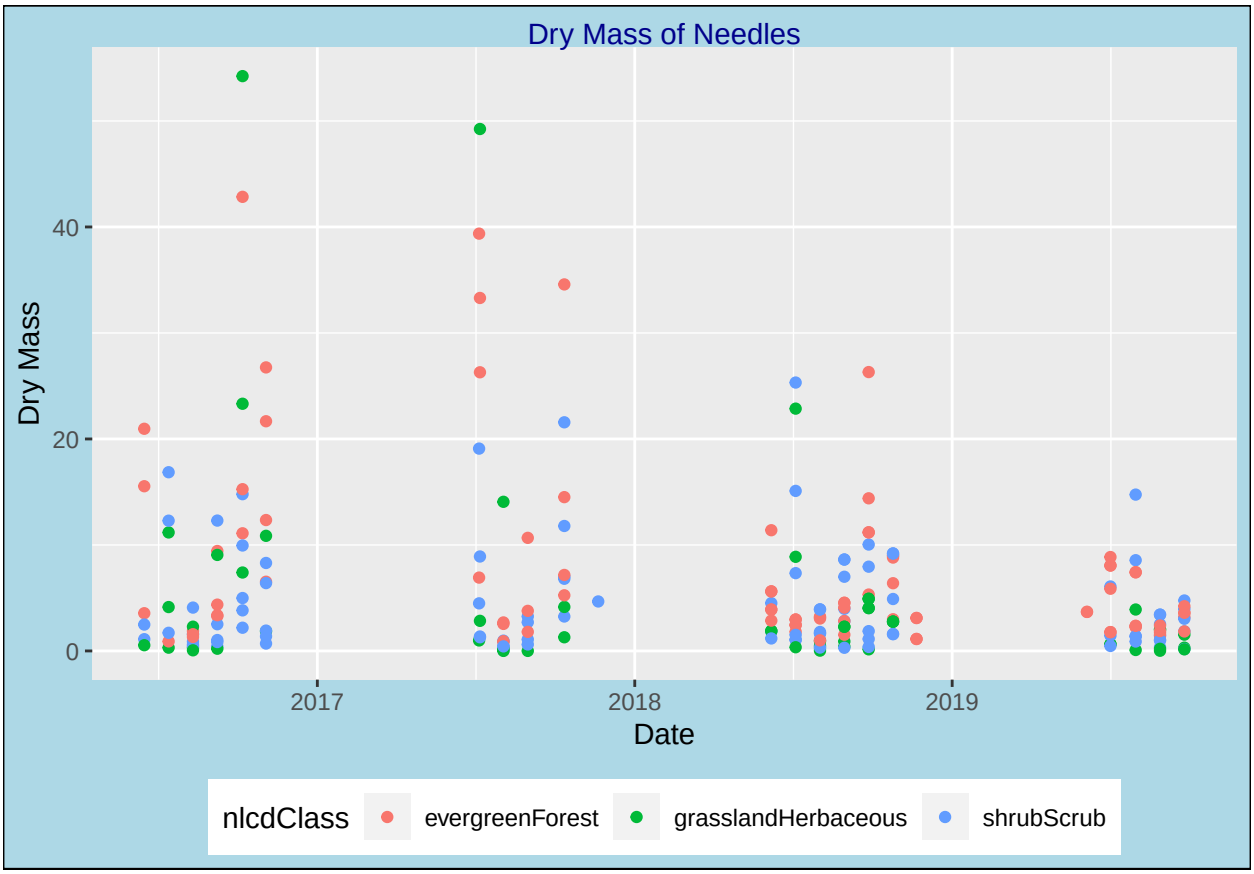
> Answer: We can see that data was primarily collected during the warmer months: May-Sep. Overall, there appears to be larger dispersion in tp_ug for Peter Lake than Paul Lake, where we see shorter boxplots. In observing tn_ug, we see that the two lakes have similar floors, though again we see Peter lake having higher dispersion, its third quartile reaching over 1000 in the month of August.

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)

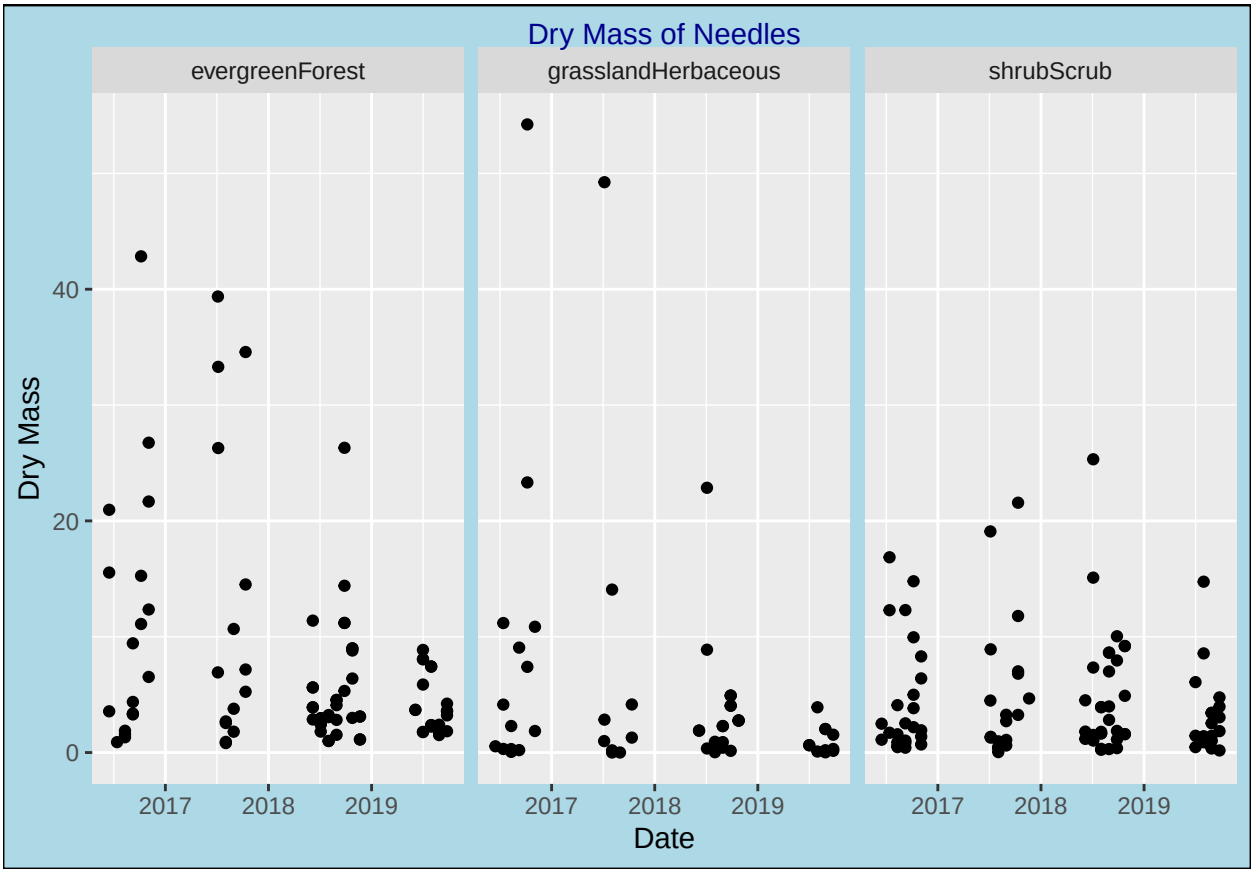
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6 First pipe to select needles only, then use gg plot based on the specified parameters
needles_df <- Niwot %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(
    mapping = aes(
      x=collectDate,
      y=dryMass,
      color=nlcdClass)
```

```
    ) +
  geom_point()+
  labs(title = "Dry Mass of Needles", x = "Date", y = "Dry Mass")
print(needles_df)
```




```
#7 First pipe to select needles only, then use gg plot based on the specified parameters
needles_df_facet <- Niwot %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x=collectDate,
      y=dryMass))+
  facet_wrap(~ nlcdClass, nrow = 1) +
  geom_point()+
  labs(title = "Dry Mass of Needles", x = "Date", y = "Dry Mass")
print(needles_df_facet)
```



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

Answer: Though I can see why the facet approach would be cleaner in instances, I think in this case the plot with color is more effective. Using that approach, we can more easily see the differences between the nlcd class through the use of color on the same chart. In using the facets, it is more difficult to compare the drymass of the different nlcd classes by only looking at the black dots in separate columns.