# Assignment 5: Data Visualization

Shiqi Zheng

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 set up
# load packages and data
library(tidyverse);library(lubridate)
```

```
## -- 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
```

```
library(here);library(cowplot);library(ggthemes)
```

```
## here() starts at /Users/shiqizheng/Desktop/ENV872/EDE_Fall2023
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
##
## Attaching package: 'ggthemes'
##
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
getwd()
```

```
## [1] "/Users/shiqizheng/Desktop/ENV872/EDE_Fall2023"
```

```
here()
```

```
## [1] "/Users/shiqizheng/Desktop/ENV872/EDE_Fall2023"
```

```
# setwd("/Users/shiqizheng/Desktop/ENV872/EDE_Fall2023/Assignments")

PeterPaul.nutrients <-
  read.csv(
  here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)
Niwot.litter <- read.csv(
  here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
                         stringsAsFactors = TRUE)

#2 change date
class(PeterPaul.nutrients$sampledate)
```

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

```
PeterPaul.nutrients$sampledate <- ymd(PeterPaul.nutrients$sampledate)
class(PeterPaul.nutrients$sampledate)
```

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

```
class(Niwot.litter$collectDate)
```

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

```
Niwot.litter$collectDate <- ymd(Niwot.litter$collectDate)
class(Niwot.litter$collectDate)
```

```
## [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

```
#3
mytheme <- theme_gray() +
  theme(axis.text = element_text(color = "black"),
        axis.ticks = element_line(color = "gray", linewidth = 0.5),
        panel.grid = element_line(color = "white", linewidth = 0.2),
        legend.position = "right",
        legend.background = element_rect(fill = "lightyellow"),
        plot.title = element_text(color = "black", size = 16, face = "bold", hjust = 0.5)
        )

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.

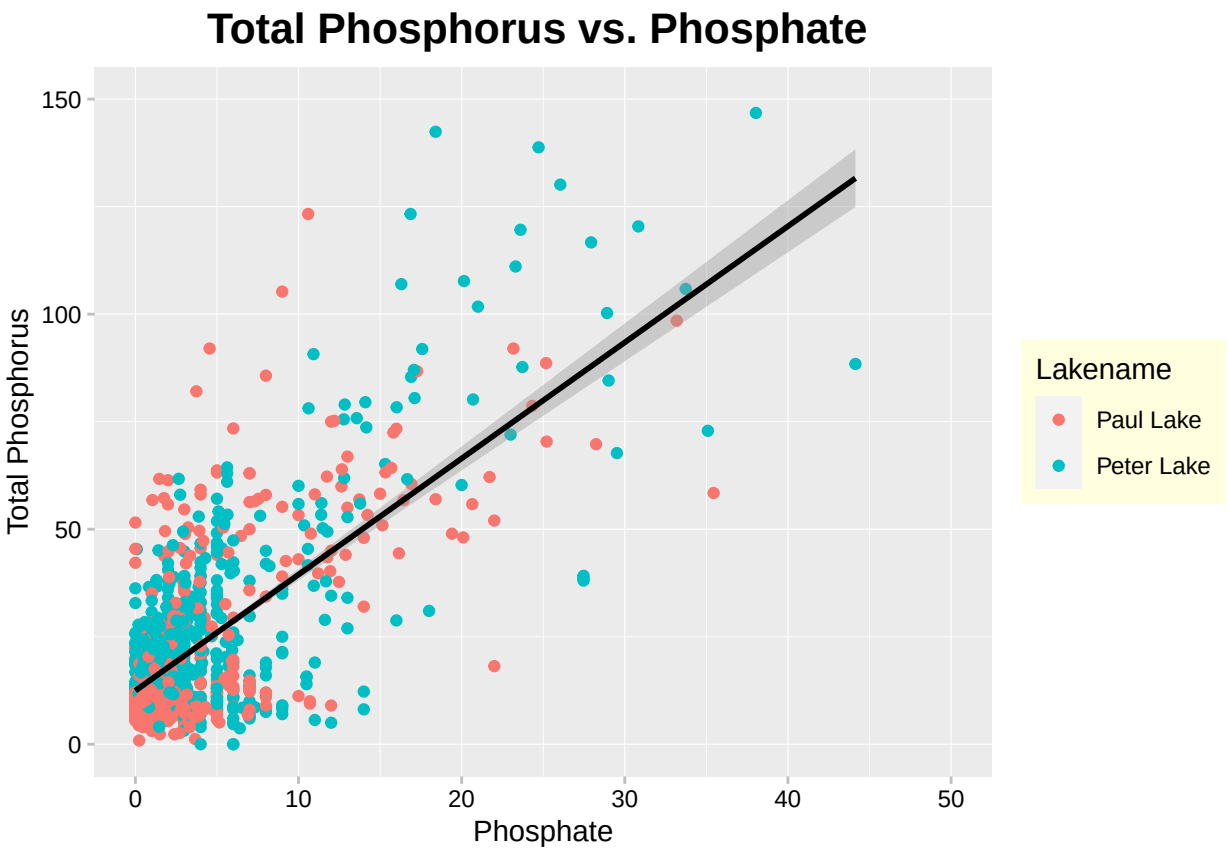
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
p_po <- ggplot(PeterPaul.nutrients, aes(x = po4, y = tp_ug, color = lakename))+
  geom_point()+
  geom_smooth(method = lm, color="black")+
  xlim(0,50)+
  ylim(0,150)+
  labs(x = "Phosphate", y = "Total Phosphorus",
       title = ("Total Phosphorus vs. Phosphate "),
       color = "Lakename")
print(p_po)
```

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

```
## Warning: Removed 21948 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21948 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: * 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

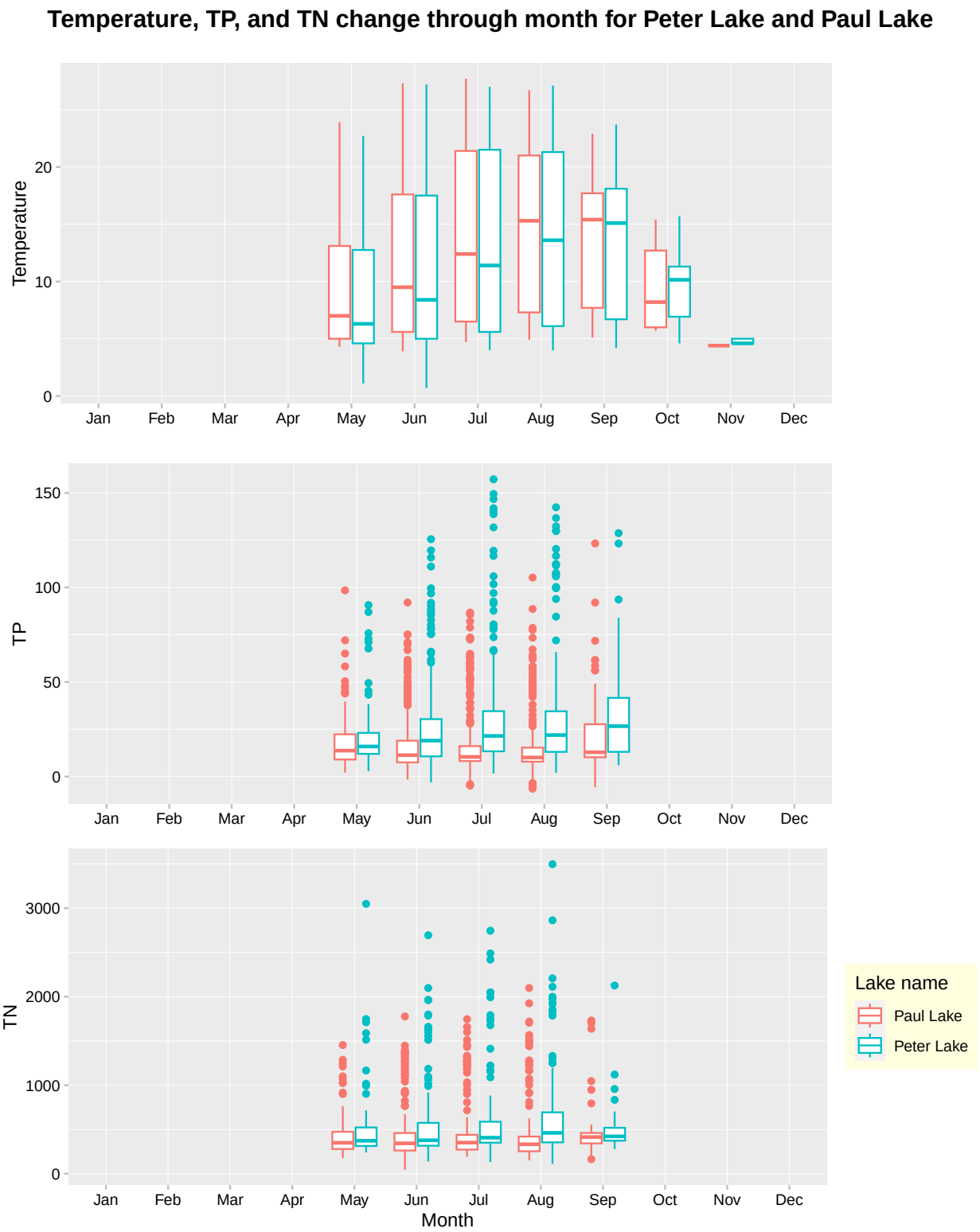
```
#5
# create seperate boxplot
boxplot_temp <- ggplot(PeterPaul.nutrients,aes(x=factor(
  PeterPaul.nutrients$month, level =1:12,
  labels=month.abb),
  y=PeterPaul.nutrients$temperature_C,
color=PeterPaul.nutrients$lakename
))+
  geom_boxplot() +
  scale_x_discrete(name="month", drop=FALSE)+
  labs(y = "Temperature")

boxplot_TP <- ggplot(PeterPaul.nutrients,
                     aes(x=factor(PeterPaul.nutrients$month,
                     level =1:12, labels=month.abb), y=PeterPaul.nutrients$tp_ug,
                    color=PeterPaul.nutrients$lakename
```

```
))+
  geom_boxplot() +
  scale_x_discrete(name="month", drop=FALSE)+
  labs(y = "TP")

boxplot_TN <- ggplot(PeterPaul.nutrients,aes(x=factor(
  PeterPaul.nutrients$month,
   level =1:12, labels=month.abb),
  y=PeterPaul.nutrients$tn_ug,
   color=PeterPaul.nutrients$lakename
))+
  geom_boxplot() +
  scale_x_discrete(name="Month", drop=FALSE)+
  labs(x = "Month", y = "TN", color = "Lake name")

title <- ggdraw() +
  draw_label("Temperature, TP, and TN change through month for Peter Lake and Paul Lake",
    fontface = 'bold')

# plot together
Lake_boxplot <-
plot_grid(title,
          boxplot_temp + theme(legend.position="none",
                               axis.title.x = element_blank(),
                               plot.margin = unit(c(0.2, 1.25, 0.1, 0.15),"in")),
          boxplot_TP + theme(legend.position="none",
                             axis.title.x = element_blank(),
                             plot.margin = unit(c(0.2, 1.25, 0.1, 0.15),"in")),
          boxplot_TN,
          nrow = 4, align = 'v', rel_heights = c(0.1,1,1,1))
```

```
## Warning: Use of ‘PeterPaul.nutrients$month‘ is discouraged.
## i Use ‘month‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$temperature_C‘ is discouraged.
## i Use ‘temperature_C‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$lakename‘ is discouraged.
## i Use ‘lakename‘ instead.

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

## Warning: Use of ‘PeterPaul.nutrients$month‘ is discouraged.
## i Use ‘month‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$tp_ug‘ is discouraged.
## i Use ‘tp_ug‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$lakename‘ is discouraged.
## i Use ‘lakename‘ instead.

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

```
## Warning: Use of ‘PeterPaul.nutrients$month‘ is discouraged.
## i Use ‘month‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$tn_ug‘ is discouraged.
## i Use ‘tn_ug‘ instead.

## Warning: Use of ‘PeterPaul.nutrients$lakename‘ is discouraged.
## i Use ‘lakename‘ instead.

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

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

```r
# add legend
#legend <- get_legend(boxplot_TN)
#Lake_boxplot <-
#plot_grid(Lake_boxplot,legend, nrow = 2, align = 'v', #rel_heights = c(10,0.1))

print(Lake_boxplot)
```




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

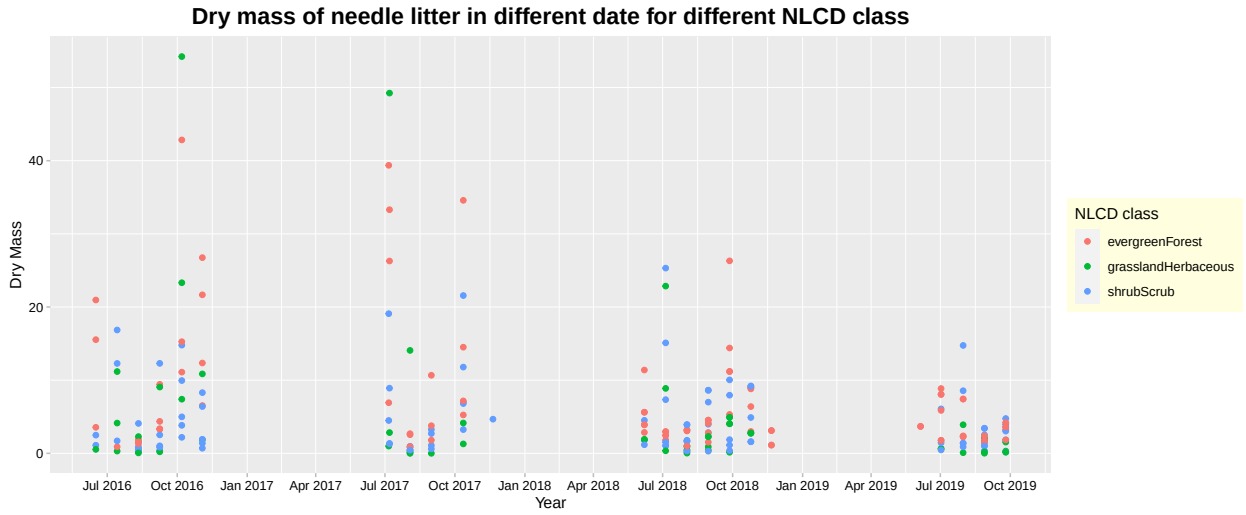
Answer: The plot shows that all data are collected between April to November, there is lack of data from December to April. For temperature of two lakes, it is high during summer. Bbefore

Sepetember Paul lake has higher temperature than Peter Lake, after september, Peter Lake's temeprature is slightly higher than Paul lake. For TP, Peter Lake is generally higher than Paul lake, and the value of TP is increasing from May to September. For TN, Peter Lake is generally slightly higher than Paul lake, and there is no obvious season variation.

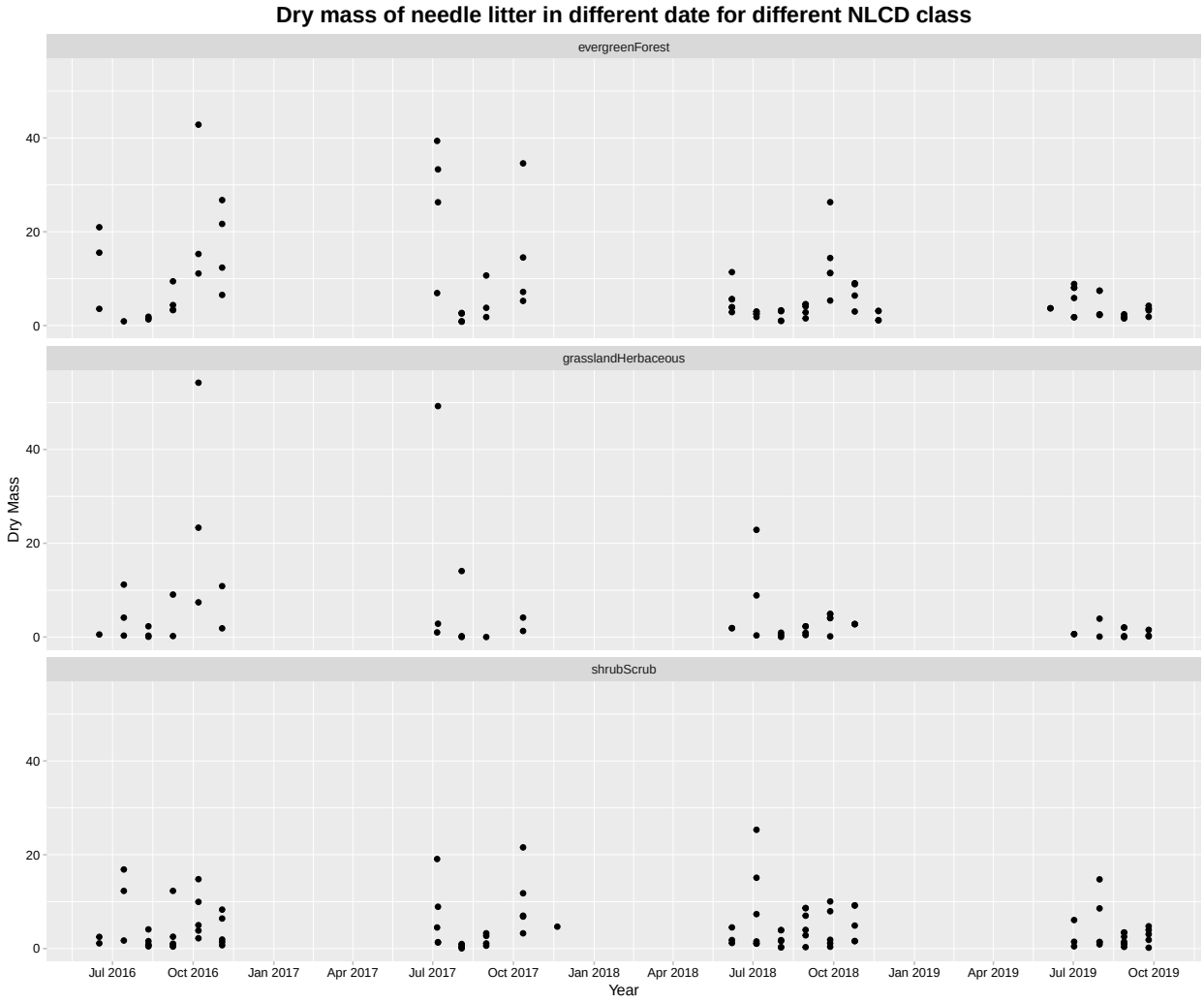
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6 separate by color
# filter needles
Niwot.litter.needles <- filter(Niwot.litter, functionalGroup == "Needles")

# plot
needles.litter <- ggplot(Niwot.litter.needles, aes(x = collectDate, y = dryMass,
                                                   color = nlcdClass))+
  geom_point()+
  scale_x_date(
    date_breaks = "3 months", date_labels = "%b %Y")+
  labs(x = "Year", y = "Dry Mass",
       title =
         ("Dry mass of needle litter in different date for different NLCD class"),
       color = "NLCD class")
print(needles.litter)
```




```
#7 separate by facets
needles.litter.facet <- ggplot(Niwot.litter.needles, aes(x = collectDate, y = dryMass))+
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)+
   scale_x_date(
    date_breaks = "3 months", date_labels = "%b %Y")+
  labs(x = "Year", y = "Dry Mass",
       title = ("Dry mass of needle litter in different date for different NLCD class"))
print(needles.litter.facet)
```




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

Answer: It depends on the goal of visualizing. For observing the relationship between dry mass of needle litter and date within each NLCD class, the plot 6 is more effective. This approach can make it easier to compare the dry mass across different NLCD classes on a single plot. For comparing the dry mass of needle litter over time across different NLCD classes, the plot 7 is more effective. Separate panels for each class allow a clear view of the temporal trend of needle litter dry mass for each NLCD class individually.