Assignment 5: Data Visualization

Yilin Zhong

Spring 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. 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_PeterPaul_Processed.csv` version) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version).

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

```
#1
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()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(here)
```

```
## here() starts at C:/Users/victo/Desktop/Spring 2023/EDA/EDA-Spring2023
```

```
#install.packages("cowplot")
library(cowplot)
```

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

```
library(ggthemes)
```

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

```
getwd()
```

```
## [1] "C:/Users/victo/Desktop/Spring 2023/EDA/EDA-Spring2023"
```

```
NTL_PeterPaul<-read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
                        stringsAsFactors = T)
NEON_Litter_Mass<-read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                           stringsAsFactors = T)

#2
NTL_PeterPaul$sampledate<-as.Date(NTL_PeterPaul$sampledate, format = "%Y-%m-%d")
class(NTL_PeterPaul$sampledate)
```

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

```
NEON_Litter_Mass$collectDate<-as.Date(NEON_Litter_Mass$collectDate, format = "%Y-%m-%d")
class(NEON_Litter_Mass$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
my_theme<-theme_base()+
  theme(
     legend.background = element_rect(
      color='grey',
      fill = 'white'),
    plot.background = element_rect(
      color = 'white'),
     plot.title = element_text(
      color = 'lightblue'),
    legend.title = element_text(
      color = 'red')
  )

theme_set(my_theme)
```

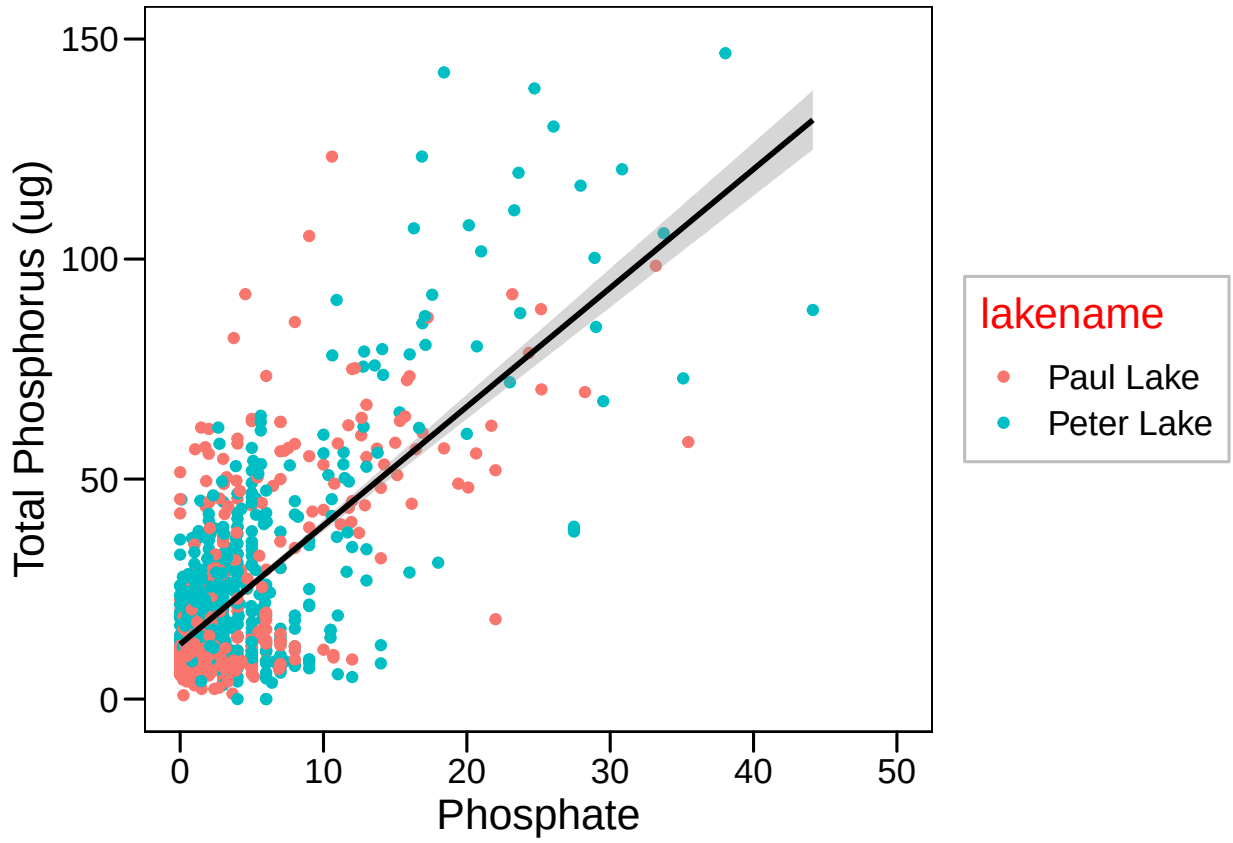
Create graphs

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
phosphorus.vs.phosphate <-
  ggplot(NTL_PeterPaul, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point()+
  labs(x= "Phosphate",
       y= "Total Phosphorus (ug)")+
  xlim(0, 50) +
  ylim(0, 150)+
  geom_smooth(method = lm, color = 'black')
print(phosphorus.vs.phosphate)
```

```
## `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).
```



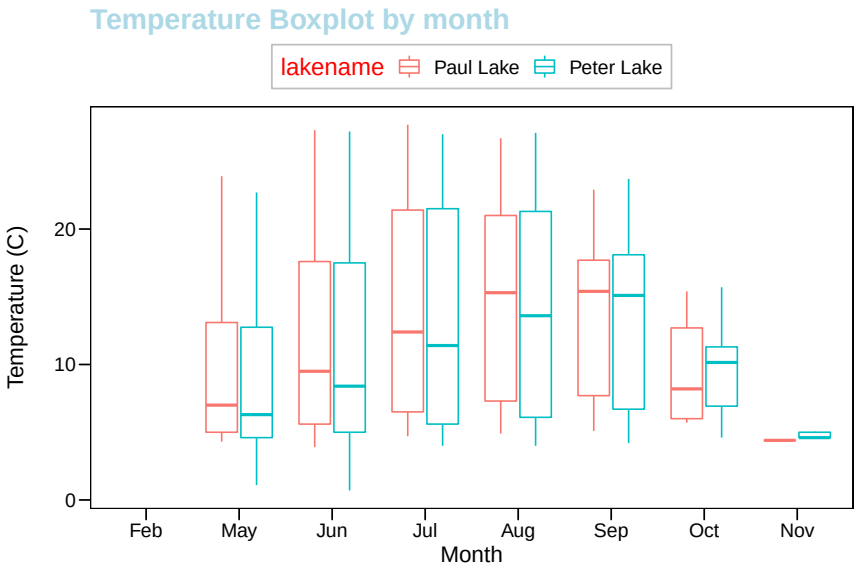
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

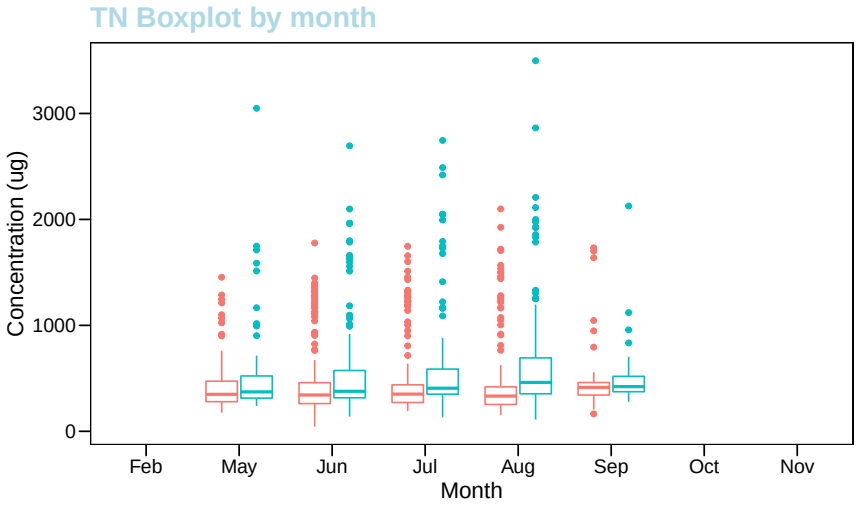
Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

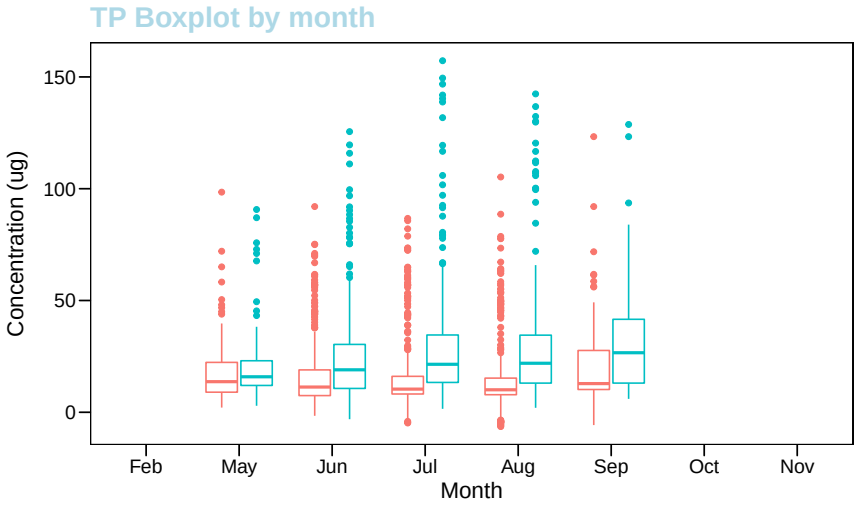
```
#5
#factor(NTL_PeterPaul$month, levels(1:12))

box_temp <-
  ggplot(NTL_PeterPaul, aes(x=factor(month, levels = 1:12, labels = month.abb), y= temperature_C,
                            color = lakename)) +
  geom_boxplot()+
  labs(x = "Month",
       y = "Temperature (C)",
       title = "Temperature Boxplot by month")+
  theme(legend.position = "top",
```

```
        axis.title.y = element_text(size = 15),
        axis.title.x = element_text(size = 15))

box_TP <-
  ggplot(NTL_PeterPaul, aes(x=factor(month, levels = 1:12, labels = month.abb), y= tp_ug,
                            color = lakename)) +
  geom_boxplot()+
  labs(x = "Month",
       y = "Concentration (ug)",
       title = "TP Boxplot by month")+
  theme(legend.position = "none",
        axis.title.y = element_text(size = 15),
        axis.title.x = element_text(size = 15))

box_TN <-
  ggplot(NTL_PeterPaul, aes(x=factor(month, levels = 1:12, labels = month.abb), y= tn_ug,
                            color = lakename)) +
  geom_boxplot()+
  labs(x = "Month",
       y = "Concentration (ug)",
       title = "TN Boxplot by month")+
  theme(legend.position = "none",
        axis.title.y = element_text(size = 15),
        axis.title.x = element_text(size = 15))

plot_grid(box_temp, box_TN, box_TP,nrow = 3, align = 'hv', axis = "bt")
```

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

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

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


Temperature Boxplot by month
lakename
Paul Lake
Peter Lake
Temperature (C)
0
10
20
Feb
May
Jun
Jul
Aug
Sep
Oct
Nov
Month


TN Boxplot by month
Concentration (ug)
0
1000
2000
3000
Feb
May
Jun
Jul
Aug
Sep
Oct
Nov
Month


TP Boxplot by month
Concentration (ug)
0
50
100
150
Feb
May
Jun
Jul
Aug
Sep
Oct
Nov
Month

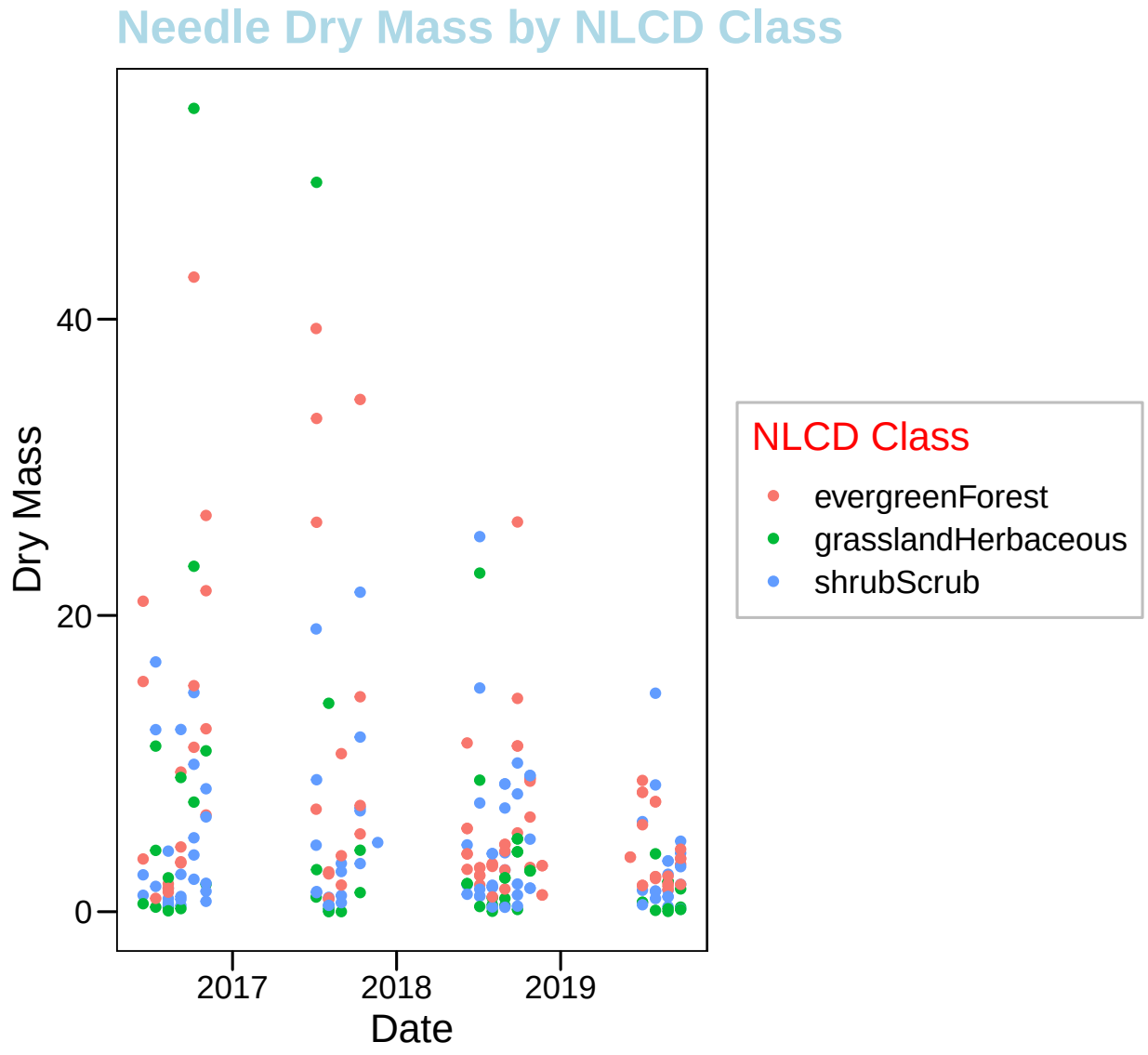
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: By observing the boxplot for temperature, we can clearly see the temperature in Paul lake is higher than temperature in Peter lake except for October and November. However, since we don't have data for October and November for TN and TP, we will not take this into our analysis. Furthermore, by looking at the boxplots for TN and TP, the concentration for TN and TP are both higher in Peter lake than in Paul lake. This might due to lower temperature in Peter lake. Moreover, over the seasons, the concentration for TN in Peter lake increases steadily from May to August and drops a little in September. As for concentration for TN in Paul lake, it does not have significant variations from May to August, but it increase a little in September. Further on, TP concentration in Peter lake, have a increasing trend from May to September. On the other hand, TP concentration in Paul lake have a decreasing trend from May to August, but start to increase in September.

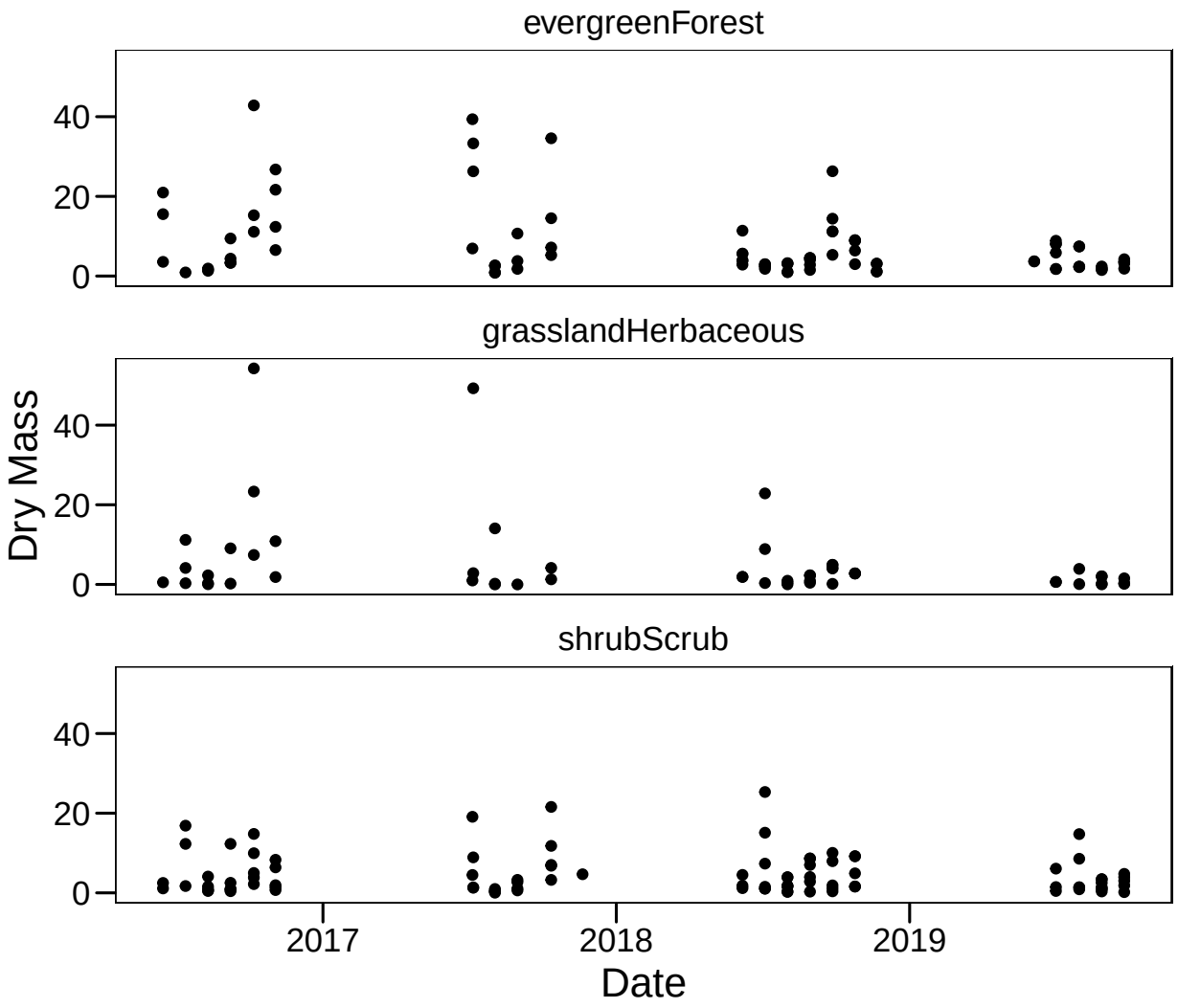
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
Niwot.needles<-ggplot(subset(NEON_Litter_Mass, functionalGroup == "Needles"),
                      aes(x=collectDate, y=dryMass, color= nlcdClass))+
  geom_point()+
  labs(x= "Date",
       y= "Dry Mass",
       title = "Needle Dry Mass by NLCD Class",
        color = "NLCD Class")
print(Niwot.needles)
```




```
#7
Niwot.needles.facet<-ggplot(subset(NEON_Litter_Mass, functionalGroup == "Needles"),
                      aes(x=collectDate, y=dryMass))+
  geom_point()+
  labs(x= "Date",
       y= "Dry Mass",
       title = "Needle Dry Mass by NLCD Class")+
  facet_wrap(vars(nlcdClass), nrow = 3)
print(Niwot.needles.facet)
```



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

Answer: Plot 7 would be more effective because we want to see the distribution of Needle dry mass for different NLCD class. Plot 6 only have one graph that show the distribution of needle dry mass and have different aesthetic for NLCD class, which is so messy that we can not clearly see the distribution of needle dry mass for each NLCD class. On the other hand, plot 7 divide the plot into 3 graphs by each NLCD class, which allow us to clearly see the distribution of needle dry mass of each NLCD class.