Assignment 5: Data Visualization

Ruiqing Li

Spring 2023

OVERVIEW

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

Directions

1. Rename this file `<RuiqingLi>_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   1.2.1      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 /Users/ruiqingli/Desktop/DataAnalytics/RWORK/EDA-Spring2023
```

```
library(cowplot)
```

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

```
getwd()
```

```
## [1] "/Users/ruiqingli/Desktop/DataAnalytics/RWORK/EDA-Spring2023"
```

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

#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Niwot.Ridge.litter$collectDate <- ymd(Niwot.Ridge.litter$collectDate)
```

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.A5 <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "pink"),
        legend.position = "top")

theme_set(mytheme.A5)
```

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.

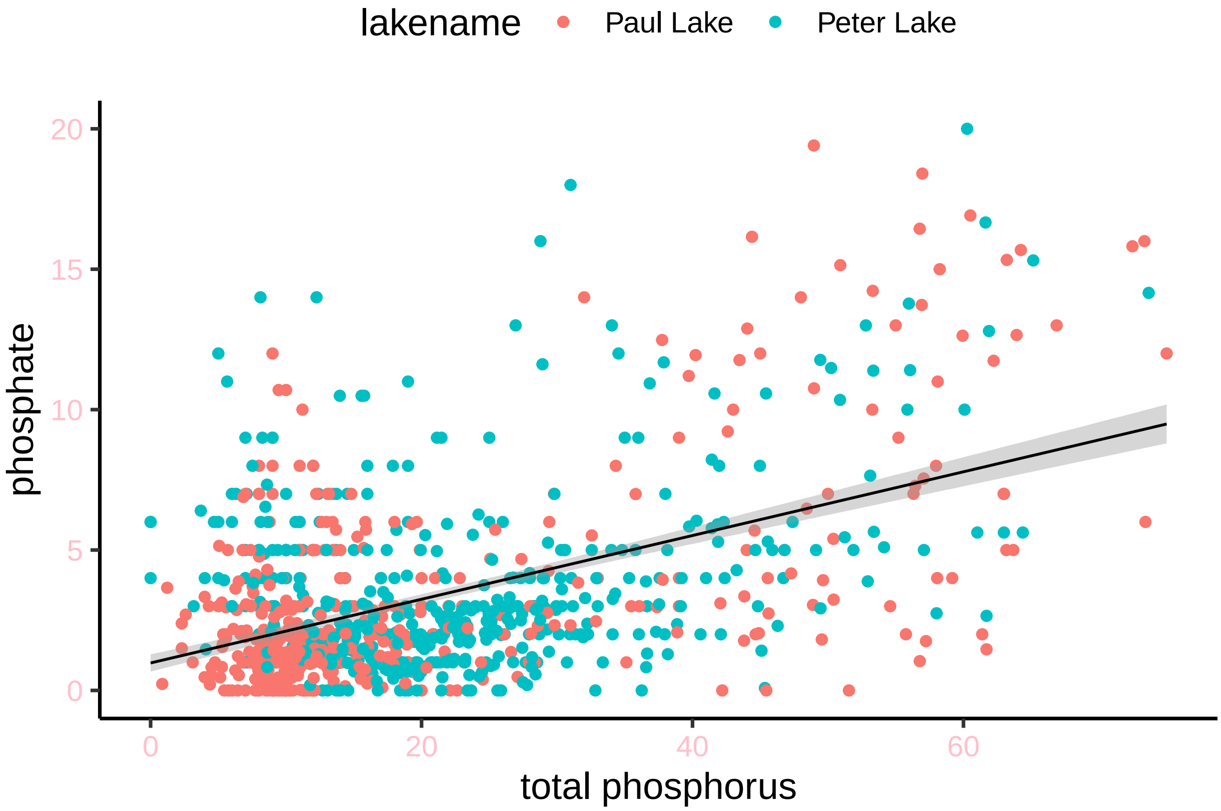
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
Total.PeterPaul.chem.nutrients<-
  ggplot(PeterPaul.chem.nutrients, aes(x = tp_ug, y = po4,color=lakename)) +
  geom_point()+
 xlim(0, 75) +
  ylim(0, 20)+
  geom_smooth(method=lm,color="black",size=0.5)+
                 xlab("total phosphorus")+
                 ylab("phosphate")
print(Total.PeterPaul.chem.nutrients)
```

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

## Warning: Removed 22002 rows containing non-finite values (stat_smooth).

## Warning: Removed 22002 rows containing missing values (geom_point).
```

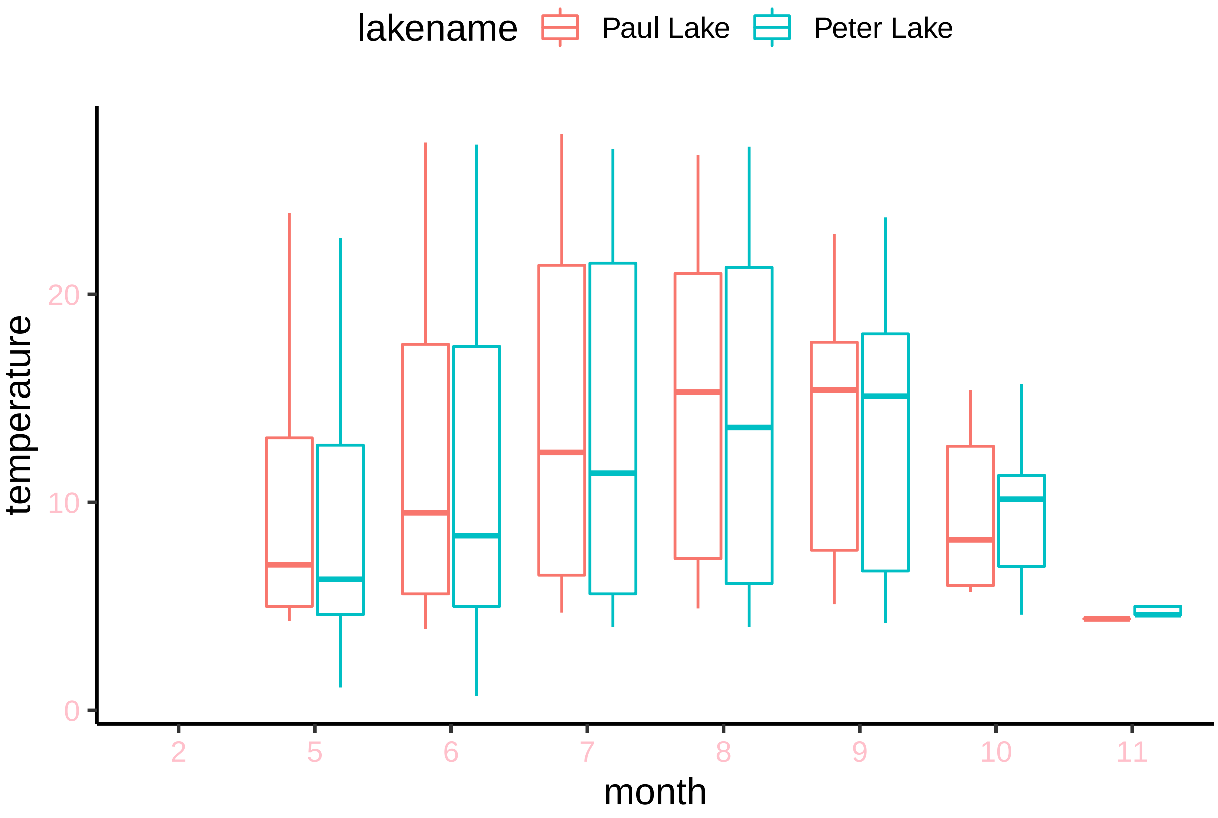


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

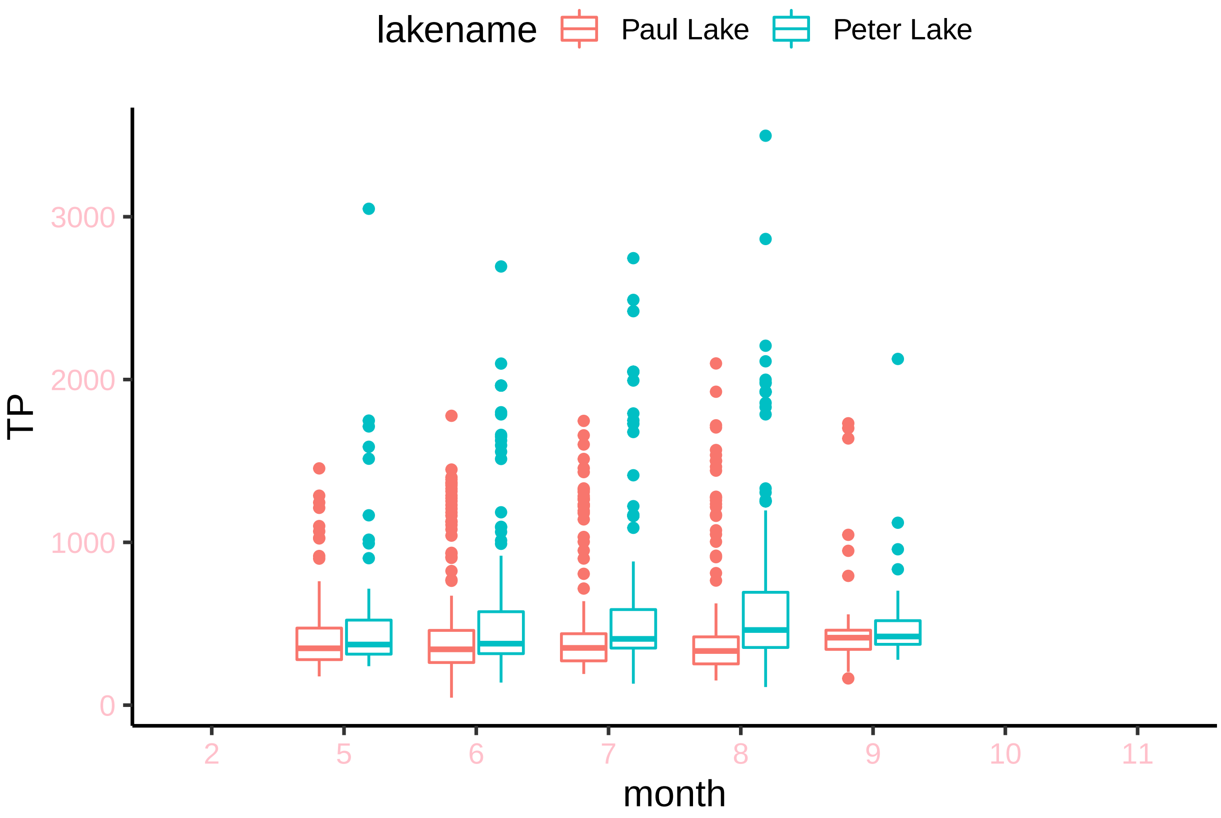
```
#5
temperature.NTL.LTER <-
  ggplot(PeterPaul.chem.nutrients, aes(x = as.factor(month), y=temperature_C)) +
  geom_boxplot(aes(color = lakename))+
               xlab("month")+
               ylab("temperature")
print(temperature.NTL.LTER)
```

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



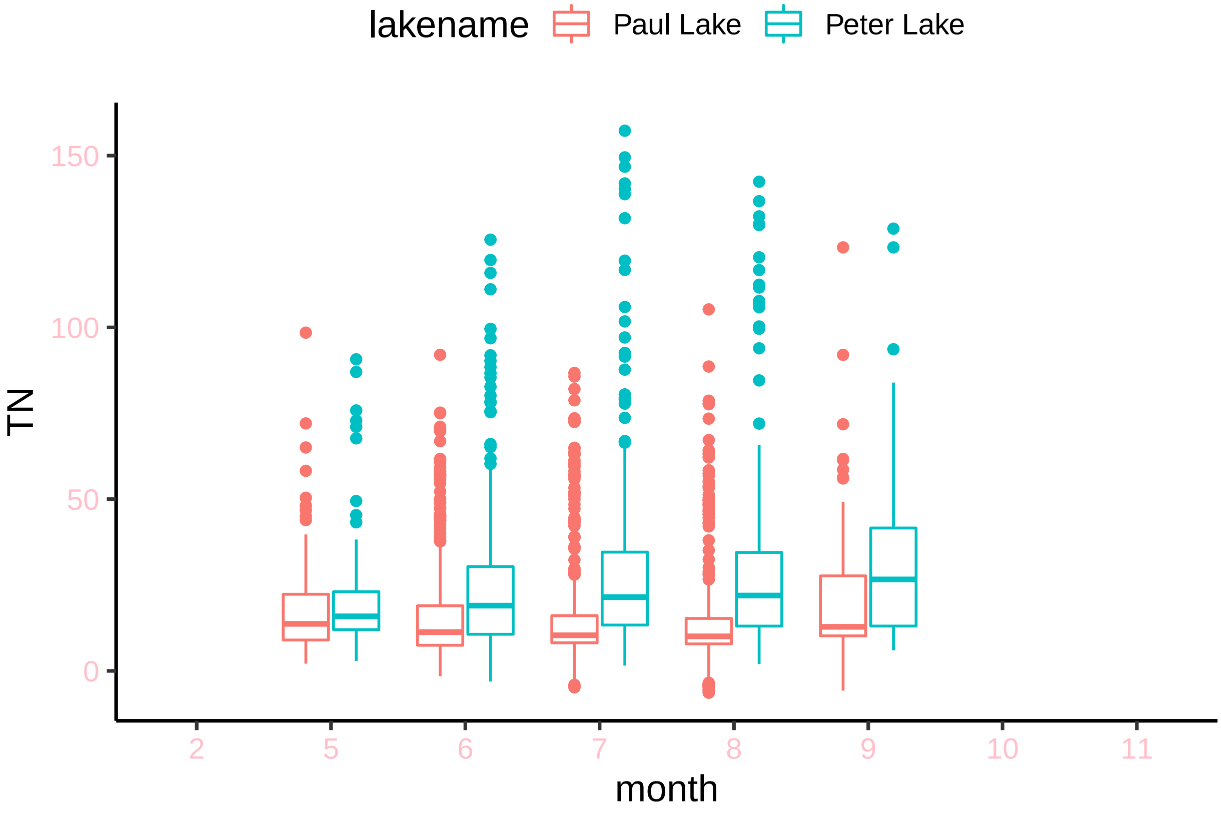
```
tn_ug.NTL.LTER <-
  ggplot(PeterPaul.chem.nutrients, aes(as.factor(month), y=tn_ug)) +
  geom_boxplot(aes(color = lakename))+
               xlab("month")+
               ylab("TP")
print(tn_ug.NTL.LTER)
```

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



```
tp_ug.NTL.LTER <-
  ggplot(PeterPaul.chem.nutrients, aes(as.factor(month), y=tp_ug)) +
  geom_boxplot(aes(color = lakename))+
                 xlab("month")+
                 ylab("TN")
print(tp_ug.NTL.LTER)
```

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

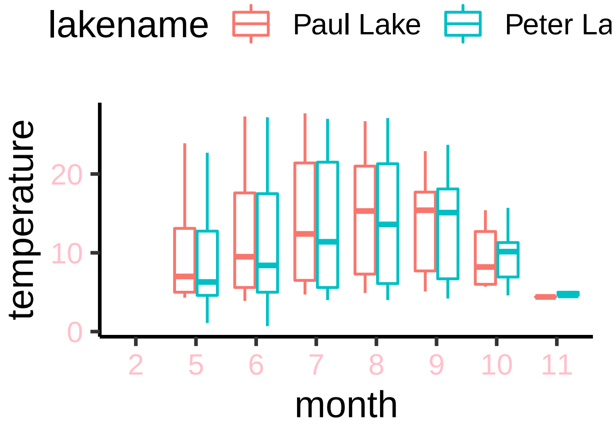


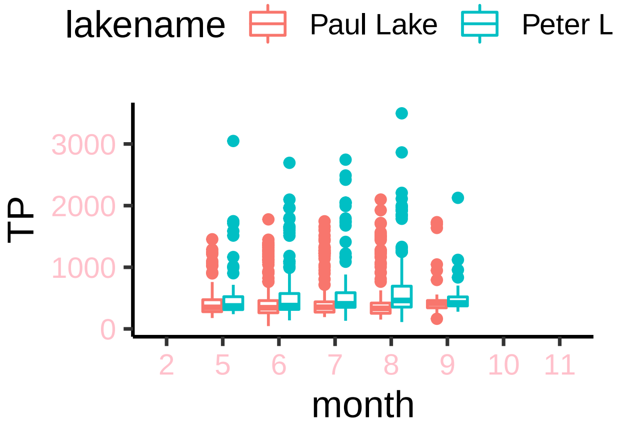
```
library(cowplot)
plot_grid(temperature.NTL.LTER, tn_ug.NTL.LTER, tp_ug.NTL.LTER, nrow = 2, align = 'h', rel_heights = c(
```

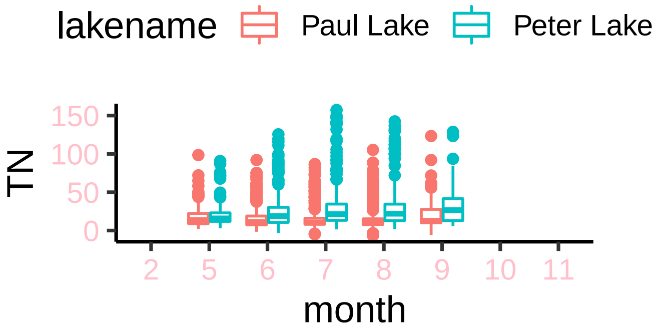
```
## Warning: Removed 3566 rows containing non-finite values (stat_boxplot).
```

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

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





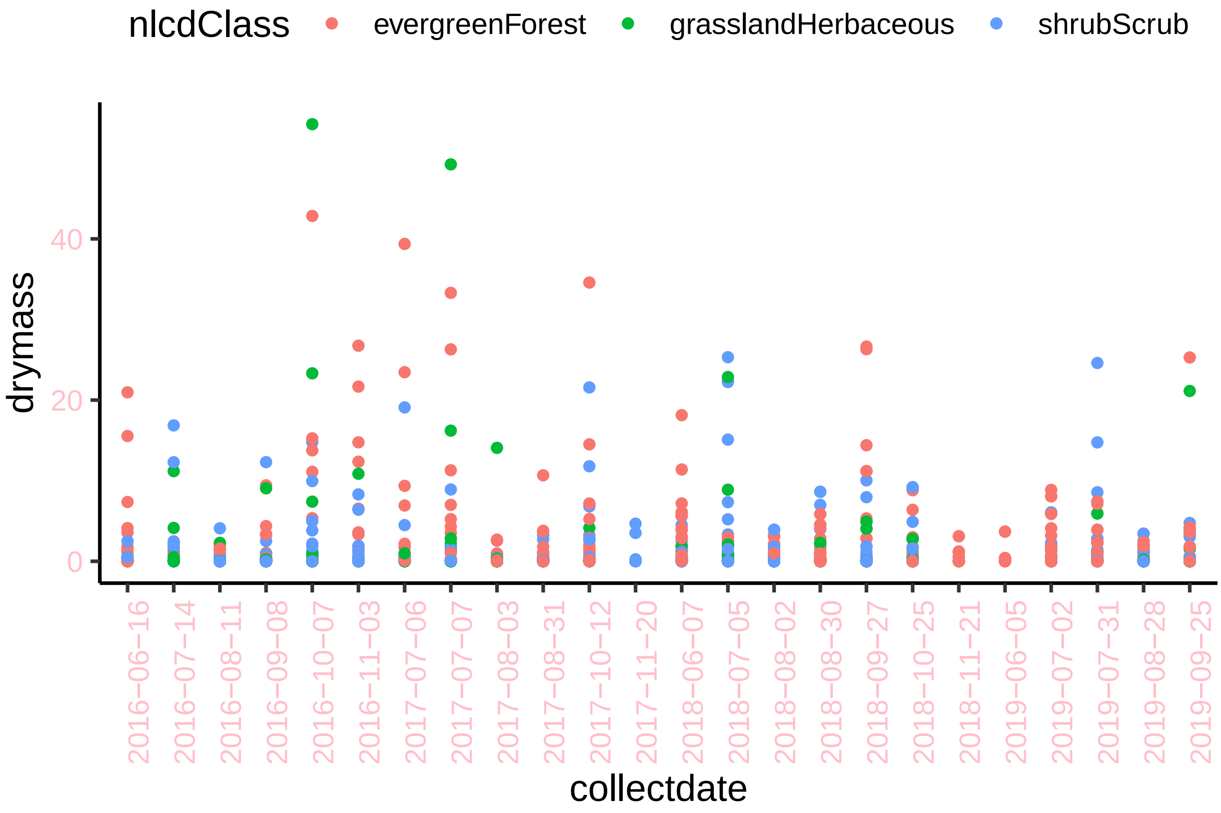


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Over the season, the temperature difference at Peter Lake tend to be greater, the TP and TN at Peter lake have more outliers.

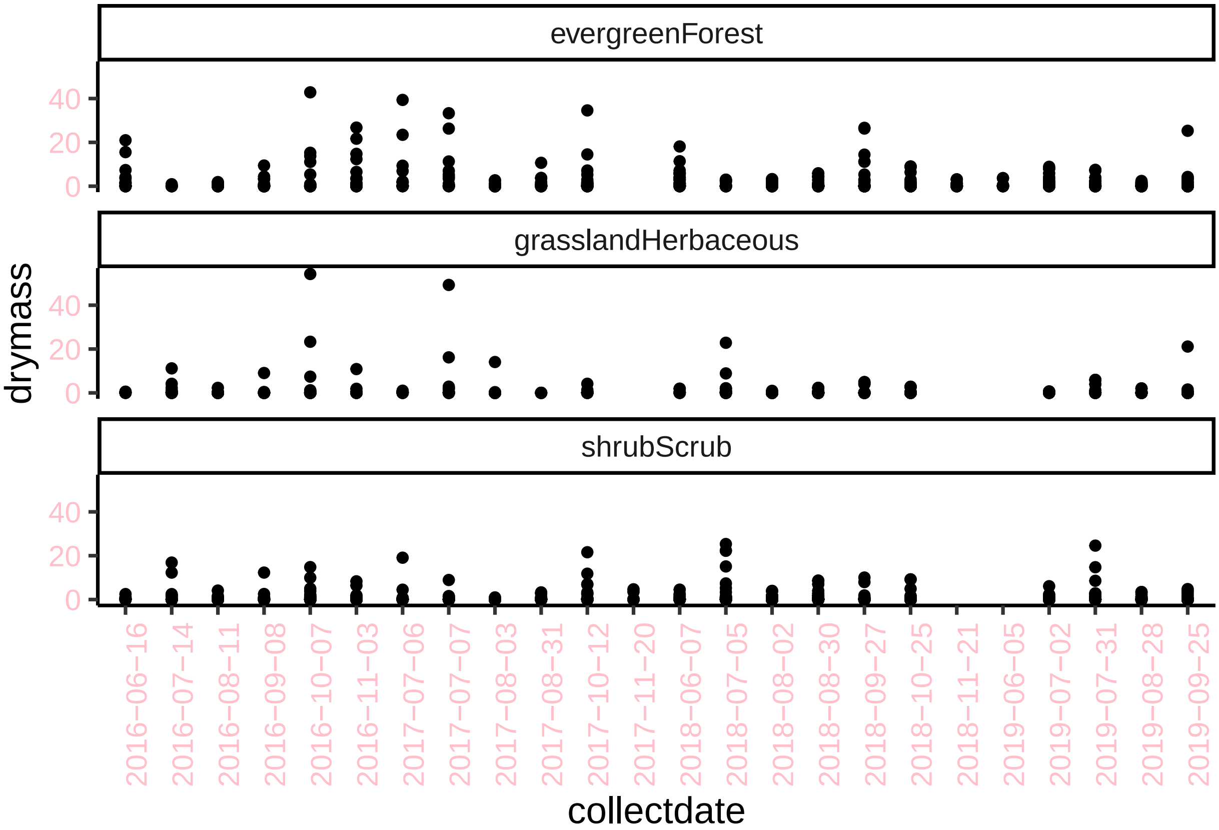
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```r
#6
needles.Niwot.Ridge.litter <-
  ggplot(subset(Niwot.Ridge.litter, functionalGroup = "Needles"),
         aes( x = as.factor(collectDate), y = dryMass)) +
  geom_point(aes(color= nlcdClass))+
                xlab("collectdate")+
                ylab("drymass")+
  theme(axis.text.x = element_text(angle = 90))
print(needles.Niwot.Ridge.litter)
```



```
#7
needles.Niwot.Ridge.litter.faceted =
ggplot(Niwot.Ridge.litter, aes( x = as.factor(collectDate), y = dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)+
                 xlab("collectdate")+
                 ylab("drymass")+
 theme(axis.text.x = element_text(angle = 90))

print(needles.Niwot.Ridge.litter.faceted)
```



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

Answer: 6, beacuse all datas are displayed in one figure makes it easier to compare the outcome of the 3 different classes.