Assignment 5: Data Visualization

Yunting Wang

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Monday, February 14 at 7:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and 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_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
getwd()
```

```
## [1] "\\\\homedir.oit.duke.edu/users/y/yw448/RforENV872/Environmental_Data_Analytics_2022/Assignments
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cowplot)
Lake= read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", stringsAs
Litter= read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = TRUE)
#2
Lake$sampledate=as.Date(Lake$sampledate,format= "%Y-%m-%d")
Litter$collectDate=as.Date(Litter$collectDate, format = "%Y-%m-%d")
```

Define your theme

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

```
#3
mytheme= theme_bw(base_size = 14)+
  theme(axis.text=element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

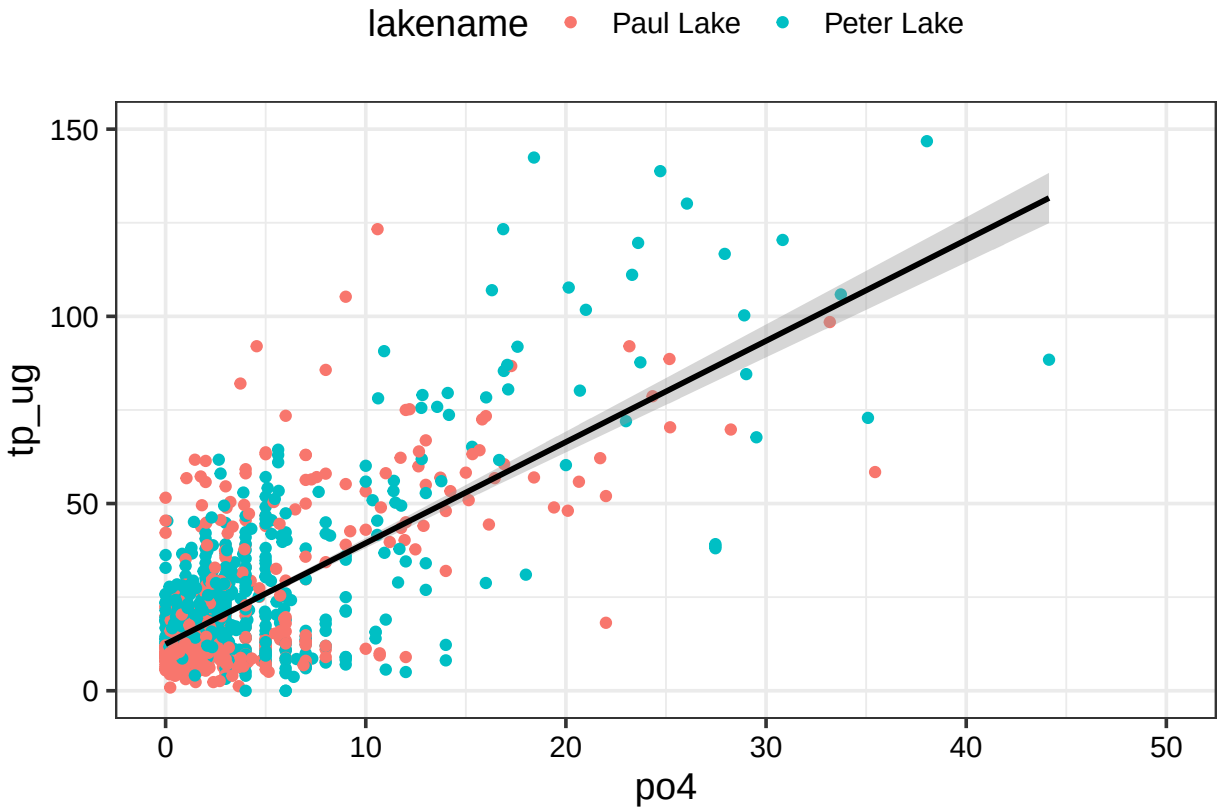
Create graphs

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 `ylim()`).

```
#4
ex4<- ggplot(Lake,aes(x=po4,y=tp_ug,color=lakename))+
  geom_point()+
  xlim(0,50)+ylim(0,150)+geom_smooth(method=lm,color="black")
print(ex4)
```

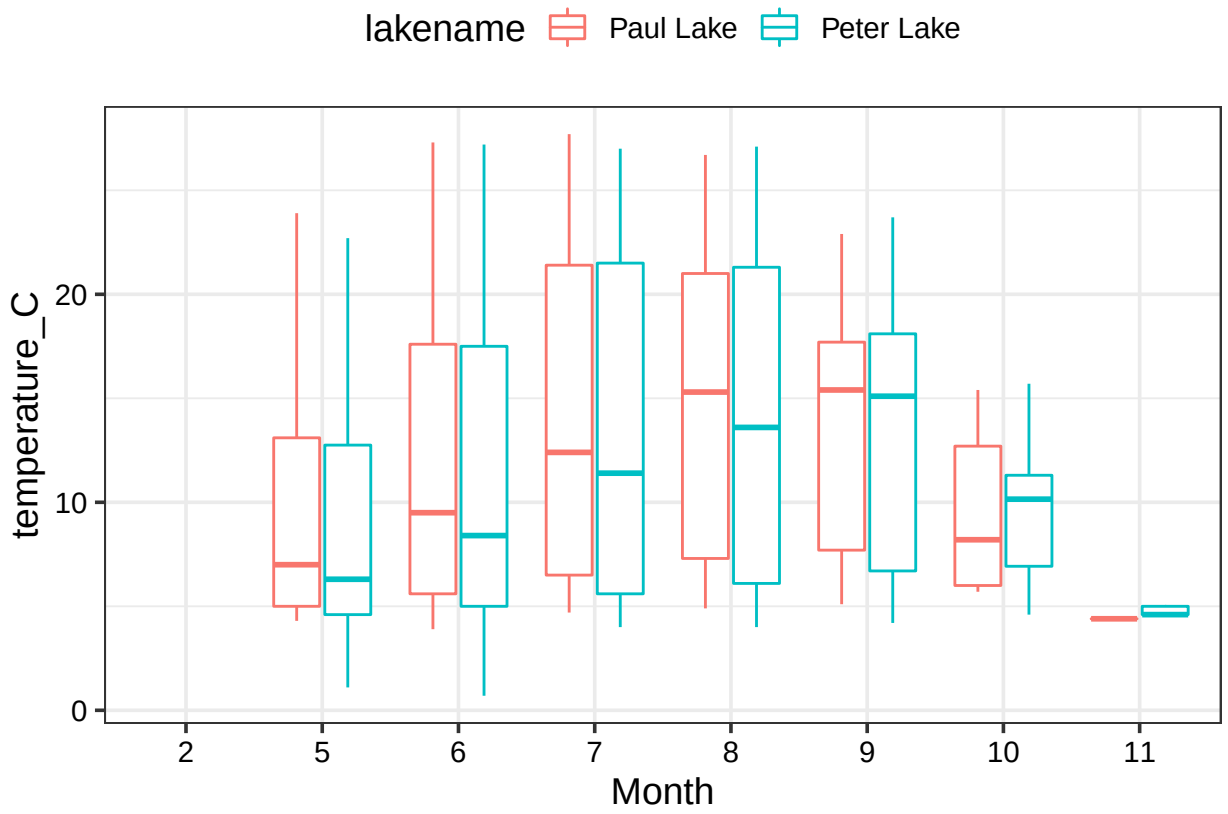
```
## `geom_smooth()` using formula 'y ~ x'

## 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.

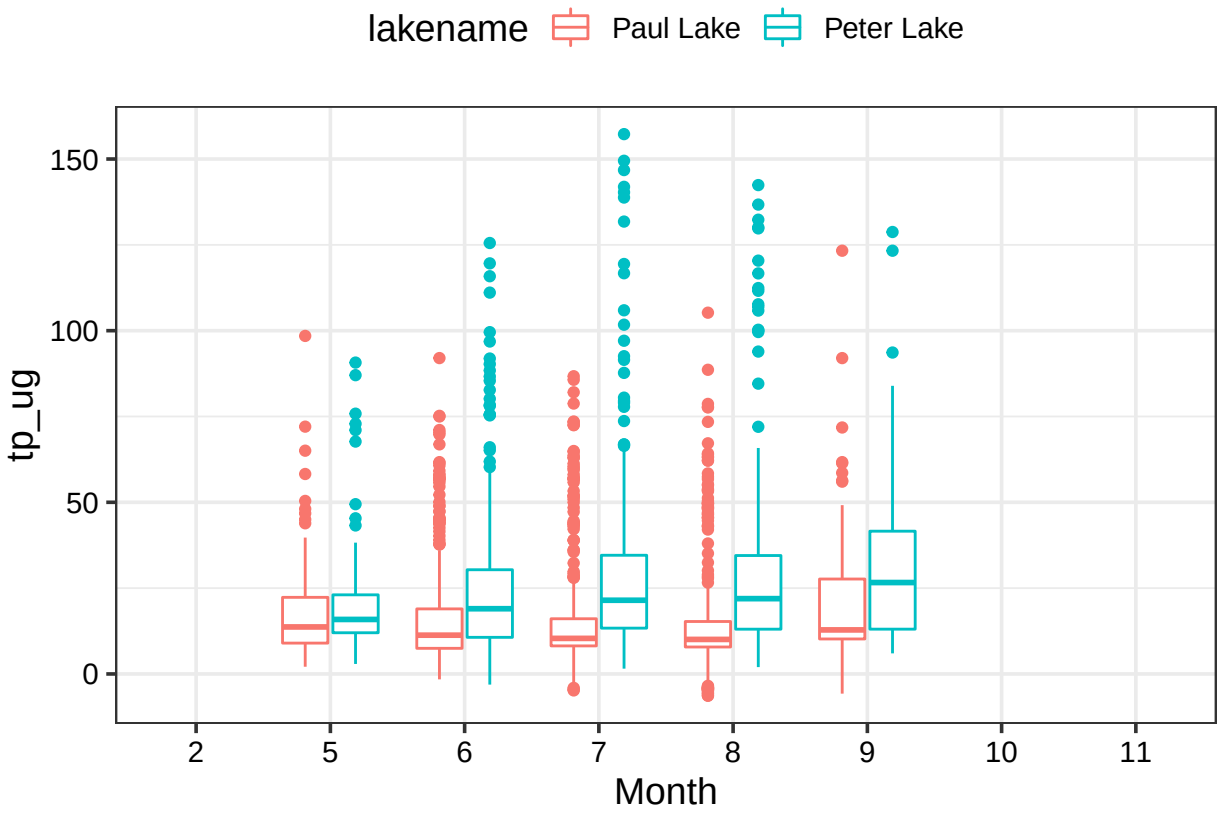
```
#5
ex5.1= ggplot(Lake,aes(x=factor(month),y=temperature_C))+ geom_boxplot(aes(color=lakename))+xlab("Month
print(ex5.1)
```

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



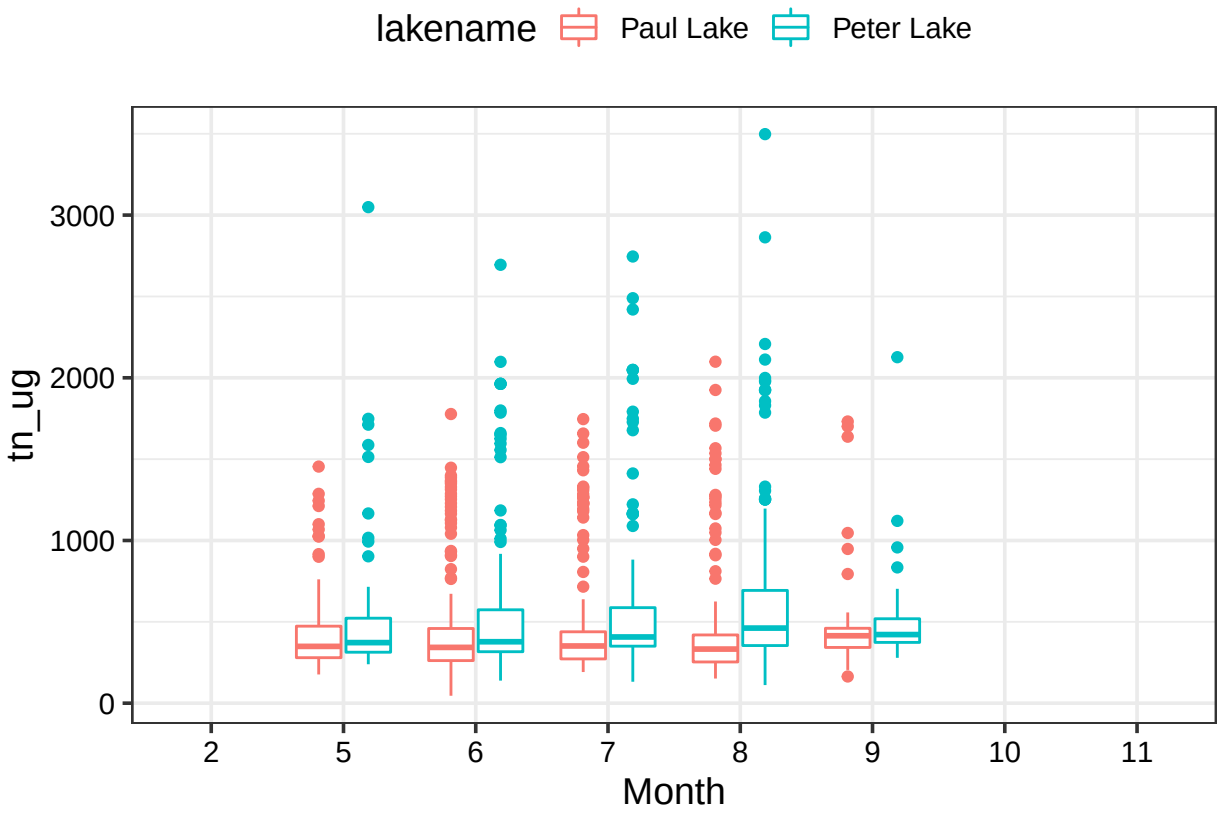

```
ex5.2=ggplot(Lake,aes(x=factor(month),y=tp_ug))+ geom_boxplot(aes(color=lakename))+xlab("Month")
print(ex5.2)
```

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




```
ex5.3=ggplot(Lake,aes(x=factor(month),y=tn_ug))+ geom_boxplot(aes(color=lakename))+xlab("Month")
print(ex5.3)
```

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




```
threeplot=plot_grid(
          ex5.1+theme(legend.position="none"),
          ex5.2+theme(legend.position="none"),
          ex5.3+theme(legend.position="none"),
          nrow=3, align = 'vh')
```

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

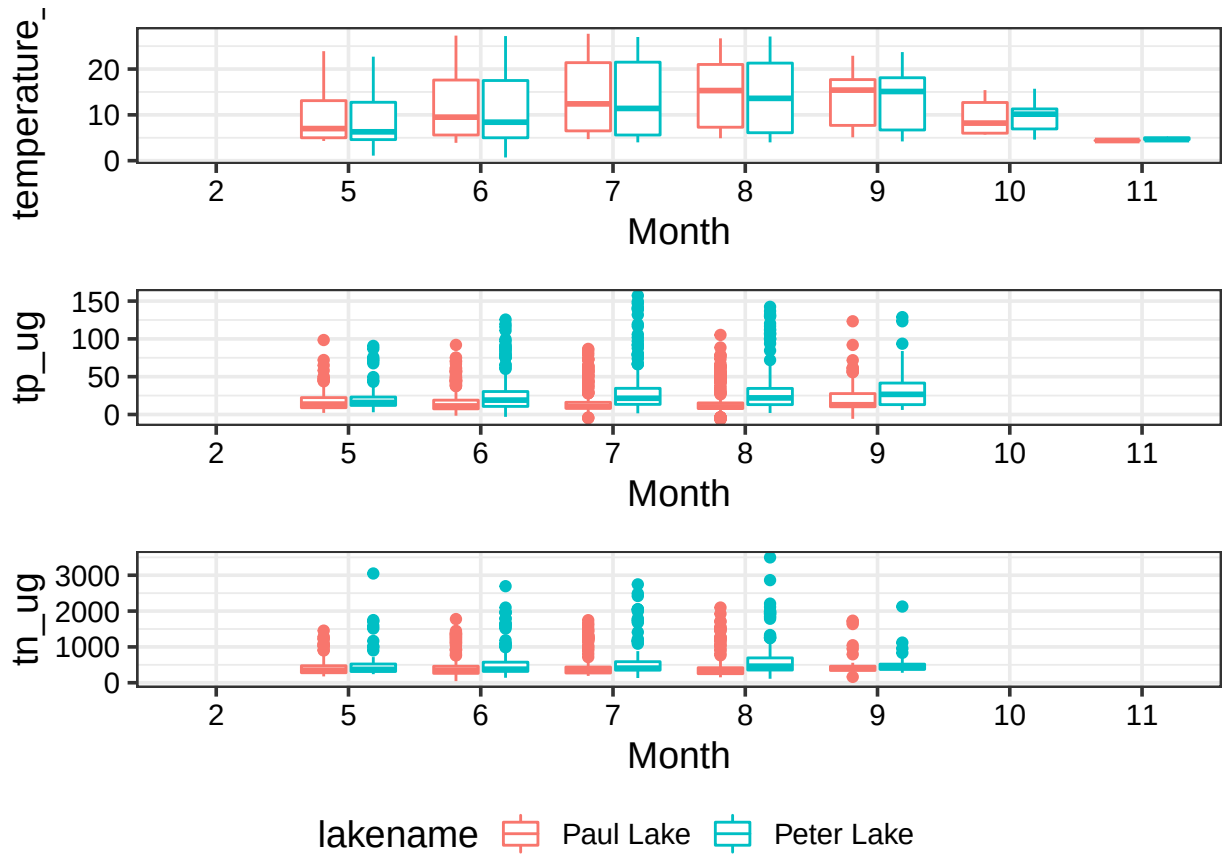
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```

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

```
legend= get_legend(ex5.1)
```

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

```
ex5.4=plot_grid(
  threeplot,legend,ncol = 1,rel_heights = c(1, .1))
print(ex5.4)
```



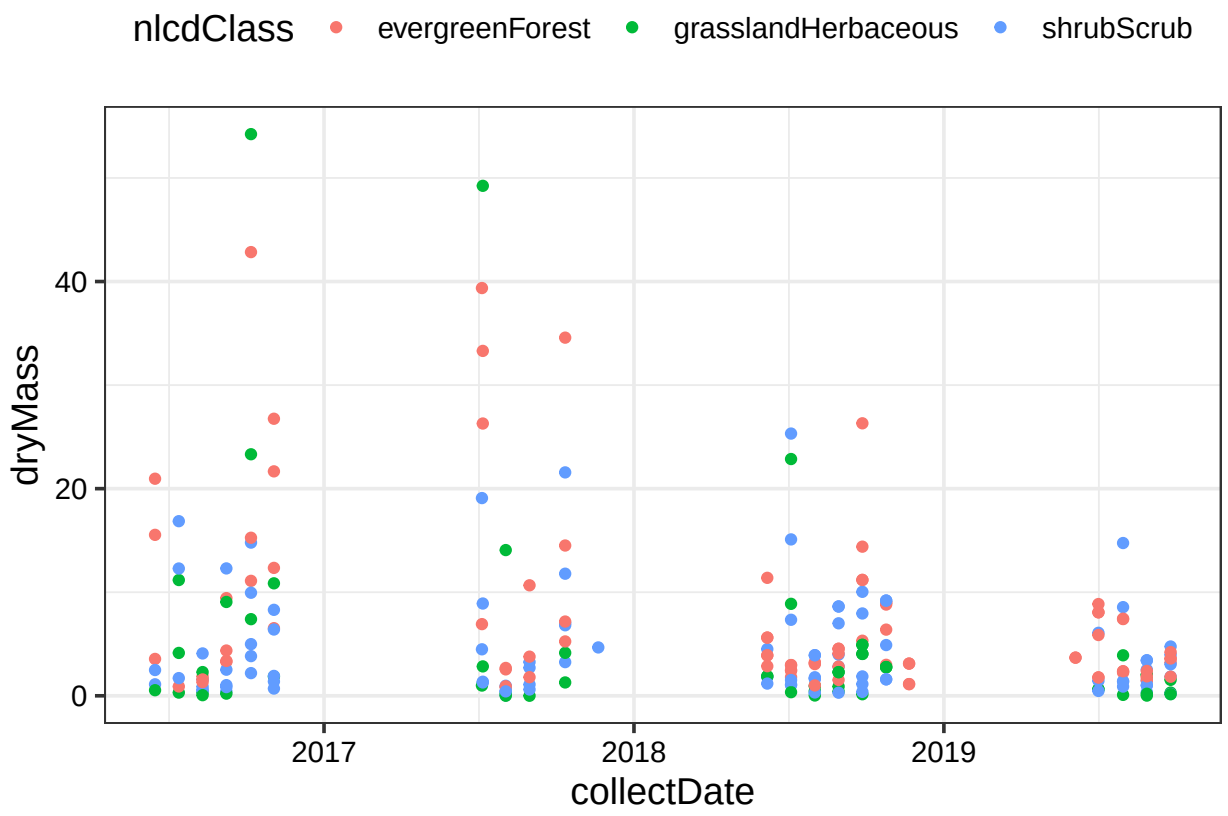
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature is higher in summer, and the overall temperature of Paul lake is slighly higher than in Peter lake; Both concentration of TN and TP are higher in summer, and the overall concentration in Peter lake is slightly higher than in Paul Lake.

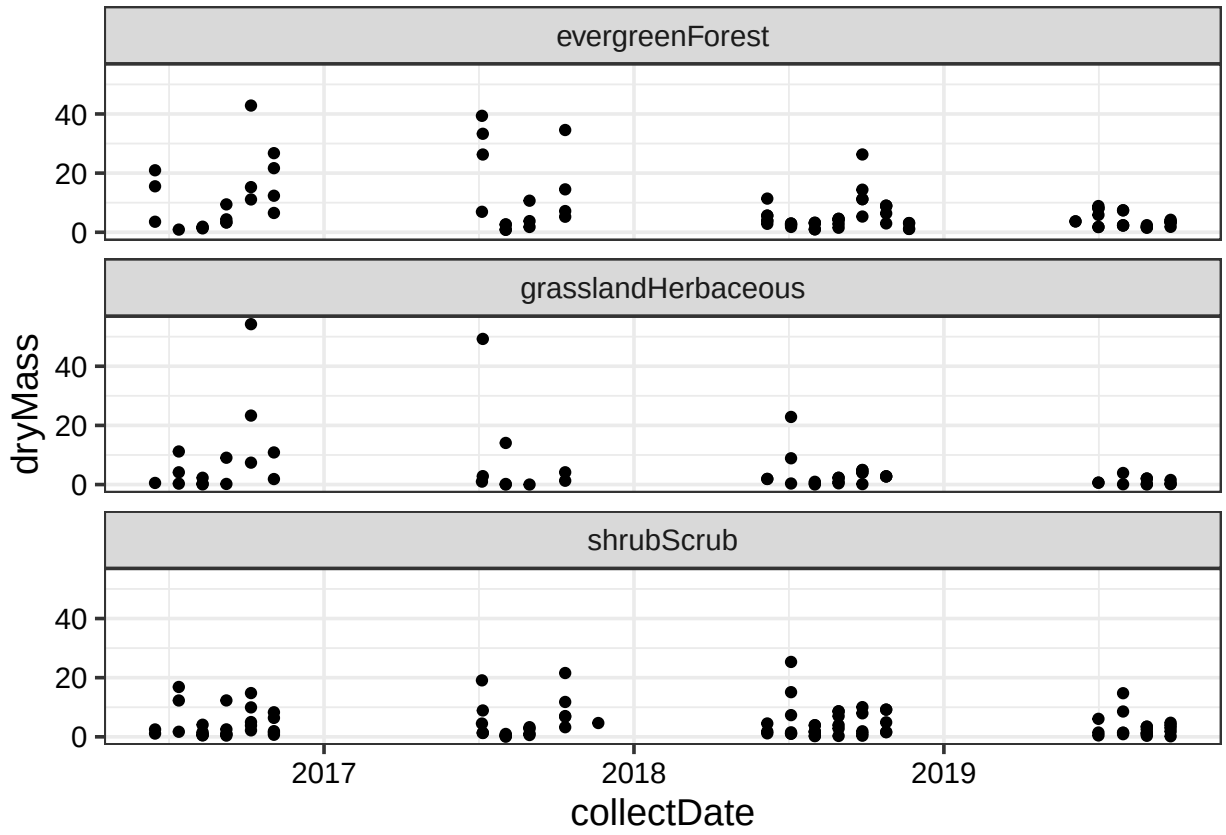
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
ex6= ggplot(subset(Litter,functionalGroup == "Needles"),aes(x=collectDate,y=dryMass,color=nlcdClass))+
  geom_point()
print(ex6)
```




```
#7
ex7= ggplot(subset(Litter,functionalGroup == "Needles"),aes(x=collectDate,y=dryMass))+
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)
print(ex7)
```



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

Answer:Plot 6, It is easier to make comparisons among groups when they're ploted together.