Assignment 5: Data Visualization

Shana Shapiro Section #1

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] "Z:/EnvironmentalDataAnalytics/Environmental_Data_Analytics_2022/Assignments"
```

```
library("tidyverse","ggplot2")
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- 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
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'purrr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
## Warning: package 'stringr' was built under R version 4.0.5
## Warning: package 'forcats' was built under R version 4.0.5
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("cowplot")
```

```
## Warning: package 'cowplot' was built under R version 4.0.5
```

```
lakes <- read.csv("../Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv")
litter <- read.csv("../Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv")

#2
class(lakes$sampledate) #character
```

```
## [1] "character"
```

```
lakes$sampledate <- as.Date(lakes$sampledate)
class(lakes$sampledate) #date
```

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

```
class(litter$collectDate) #character
```

```
## [1] "character"
```

```
litter$collectDate <- as.Date(litter$collectDate)
class(litter$collectDate) #date
```

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

Define your theme

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

```
#3
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        plot.title = element_text(face = "bold",size=12),
        axis.ticks = element_line(colour="grey70", size = 0.2),
        panel.grid.major = element_line(colour="grey70", size = 0.2),
        panel.grid.minor = element_blank())
```

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
phos <- ggplot(lakes, aes(x=tp_ug, y=po4, color = lakename)) +
  geom_point() +
```

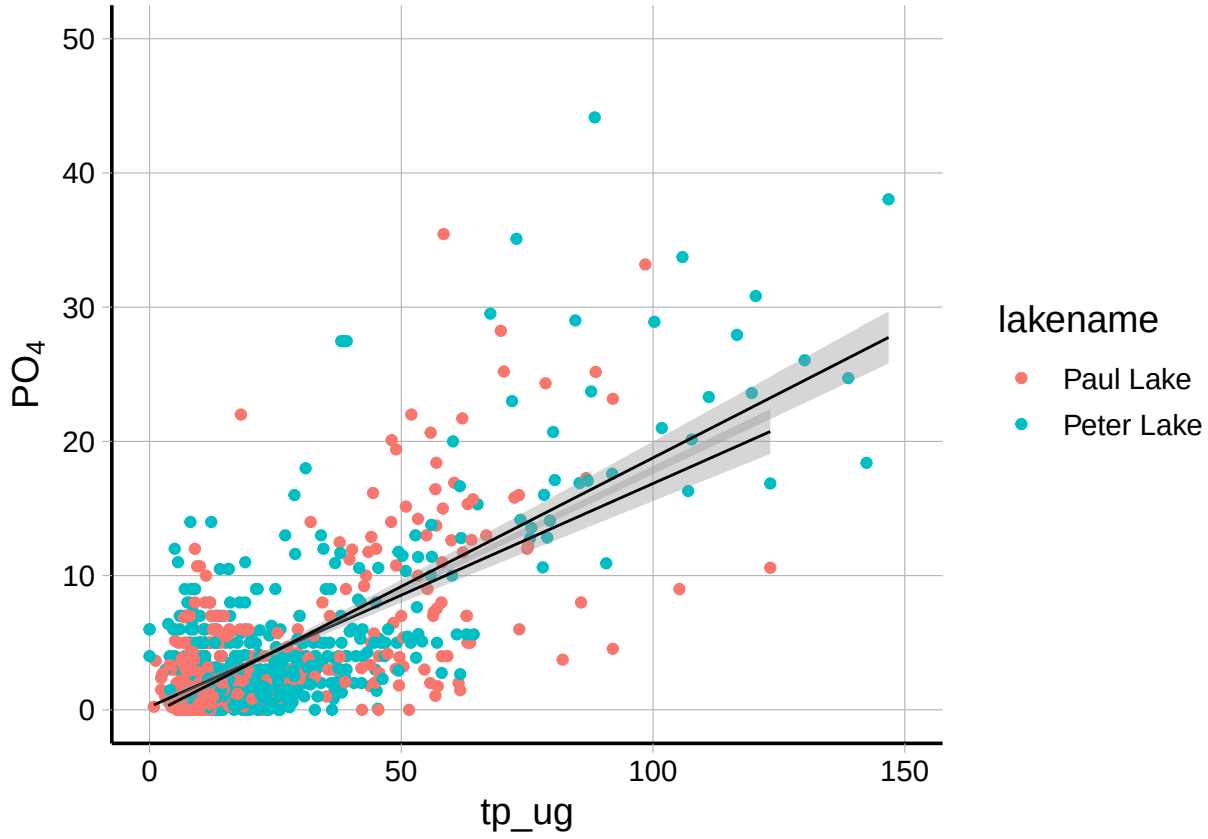
```
  mytheme +
  xlim(0,150) +
  ylim(0,50) +
  geom_smooth(aes(group=lakename), method="lm", color = "black", size=0.5) +
  ylab(expression("PO"[4]))
print(phos)
```

```
## `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).
```

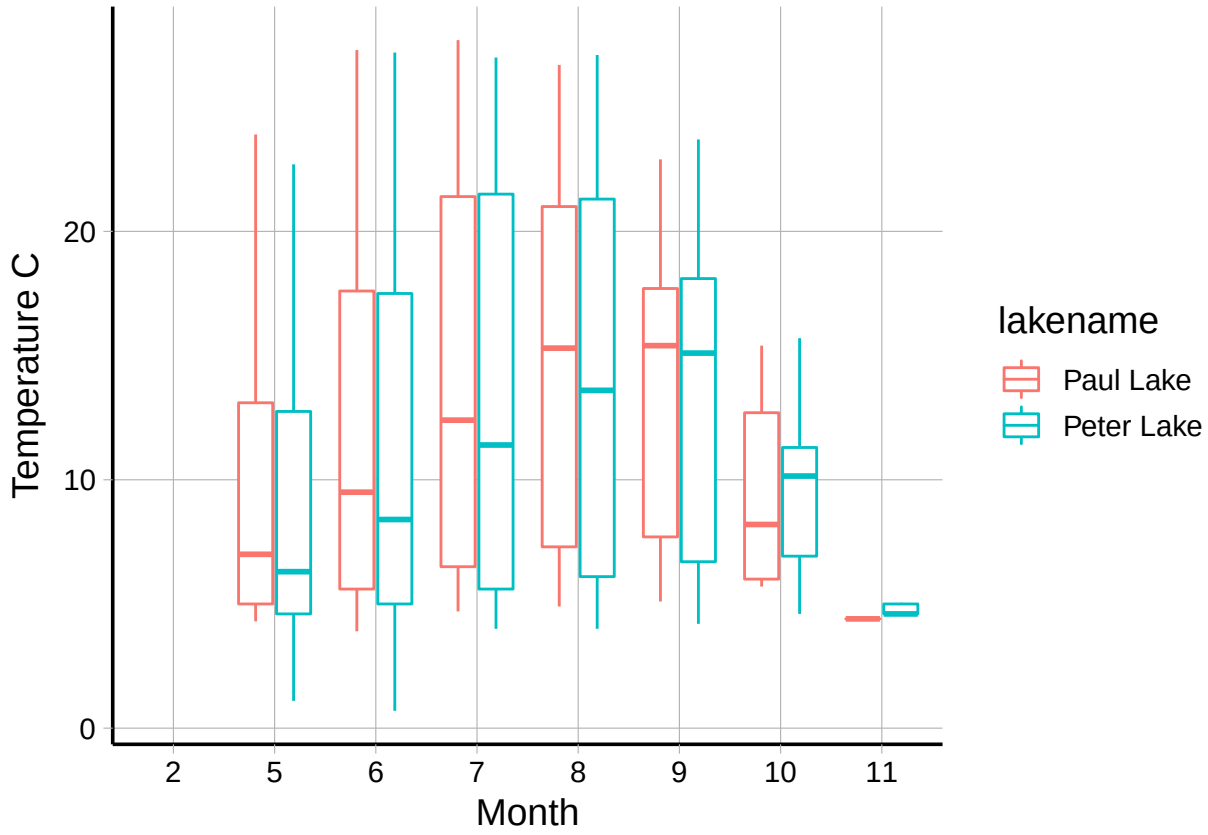
```
## Warning: Removed 2 rows containing missing values (geom_smooth).
```



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.

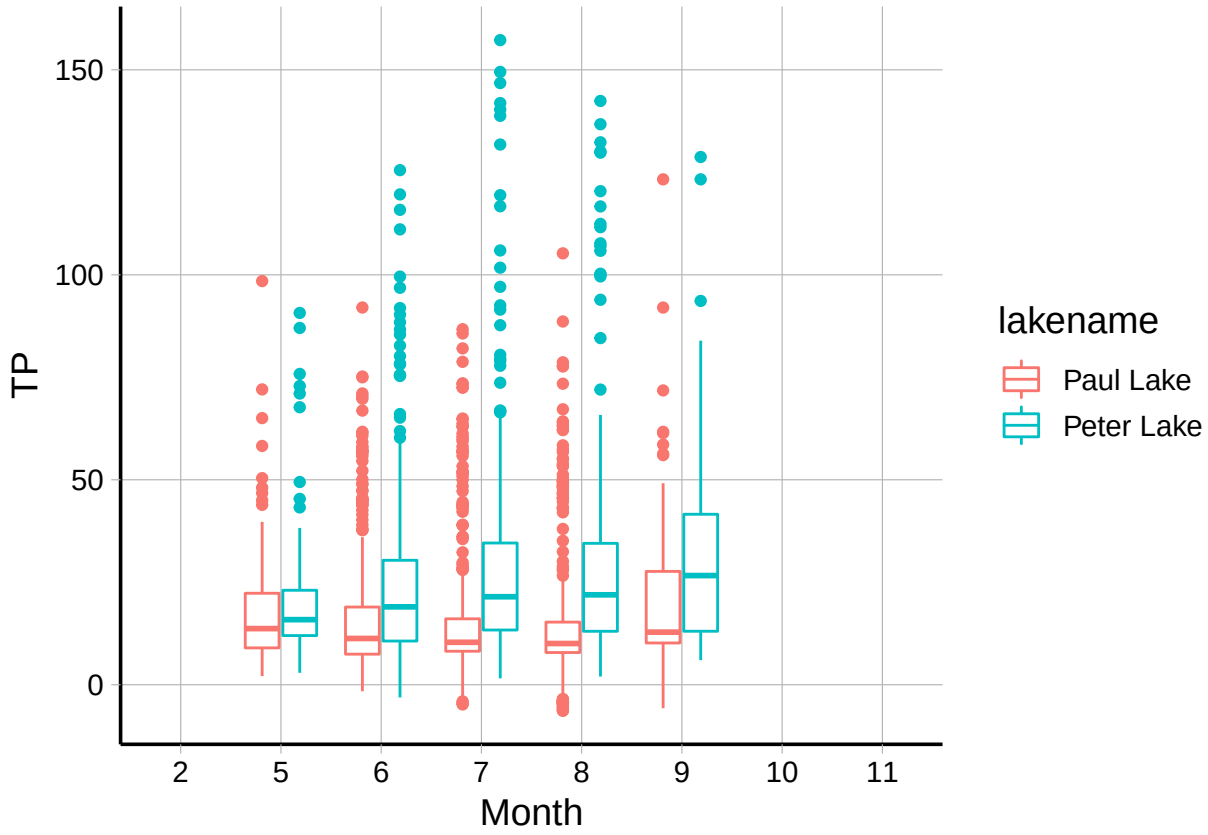
```
#5
temp <- ggplot(lakes, aes(x=as.factor(month), y=temperature_C, color = lakename)) +
  geom_boxplot() +
  ylab(expression("Temperature C")) +
  xlab(expression("Month")) +
  mytheme
print(temp)
```

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



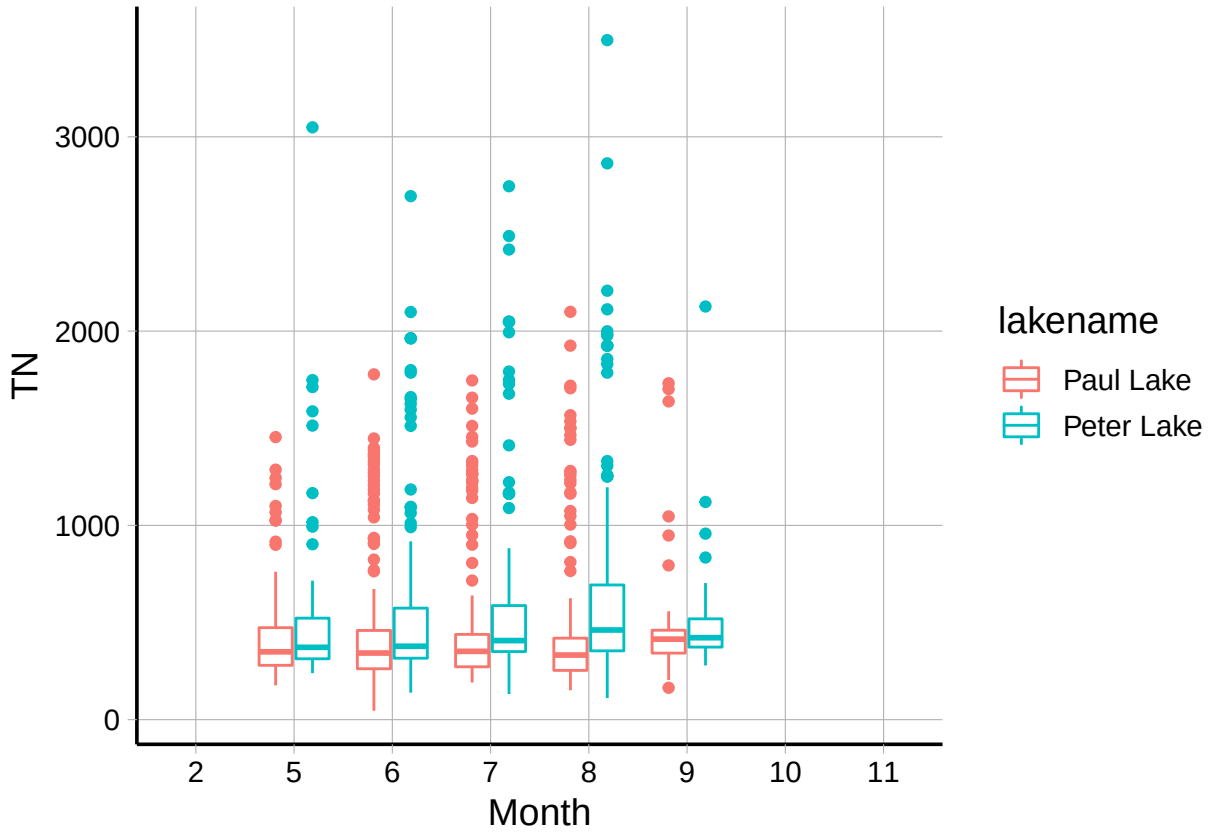

```
TP <- ggplot(lakes, aes(x=as.factor(month),y=tp_ug,color=lakename)) +
  geom_boxplot() +
  ylab(expression("TP")) +
  xlab(expression("Month")) +
  mytheme
print(TP)
```

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




```
TN <- ggplot(lakes, aes(x=as.factor(month),y=tn_ug,color=lakename))+
  geom_boxplot() +
  ylab(expression("TN")) +
  xlab(expression("Month")) +
  mytheme
print(TN)
```

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




```
#fix this
PeterPaul.temp.TP.TN <- plot_grid(temp, TP, TN, theme(legend.position = "bottom"), nrow =3, align = 'h'
```

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

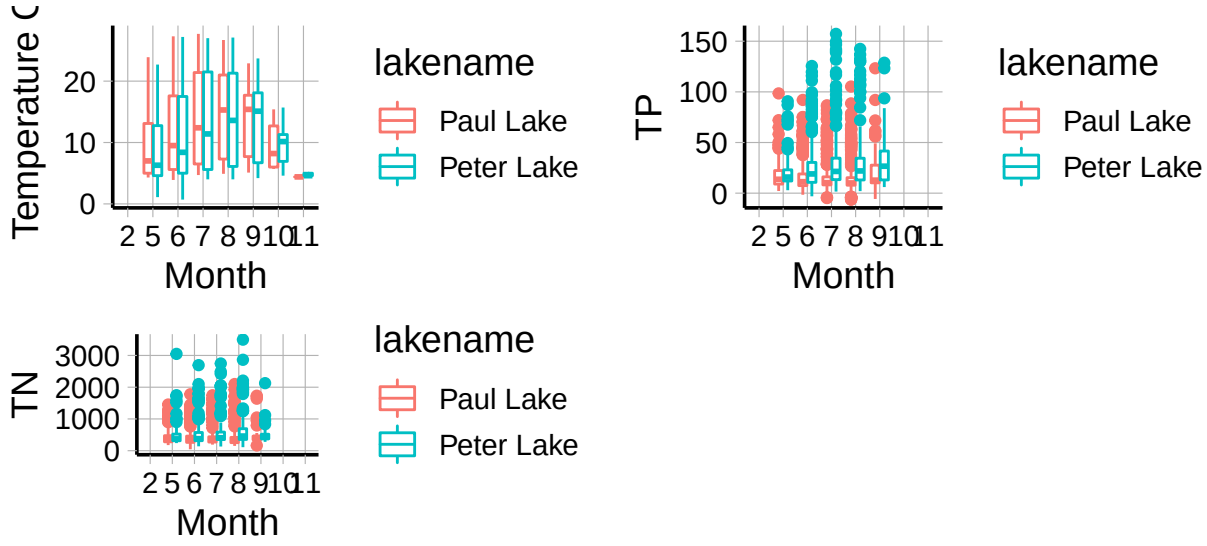
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```

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

```
## Warning in as_grob.default(plot): Cannot convert object of class themegg into a
## grob.
```

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

```
print(PeterPaul.temp.TP.TN)
```



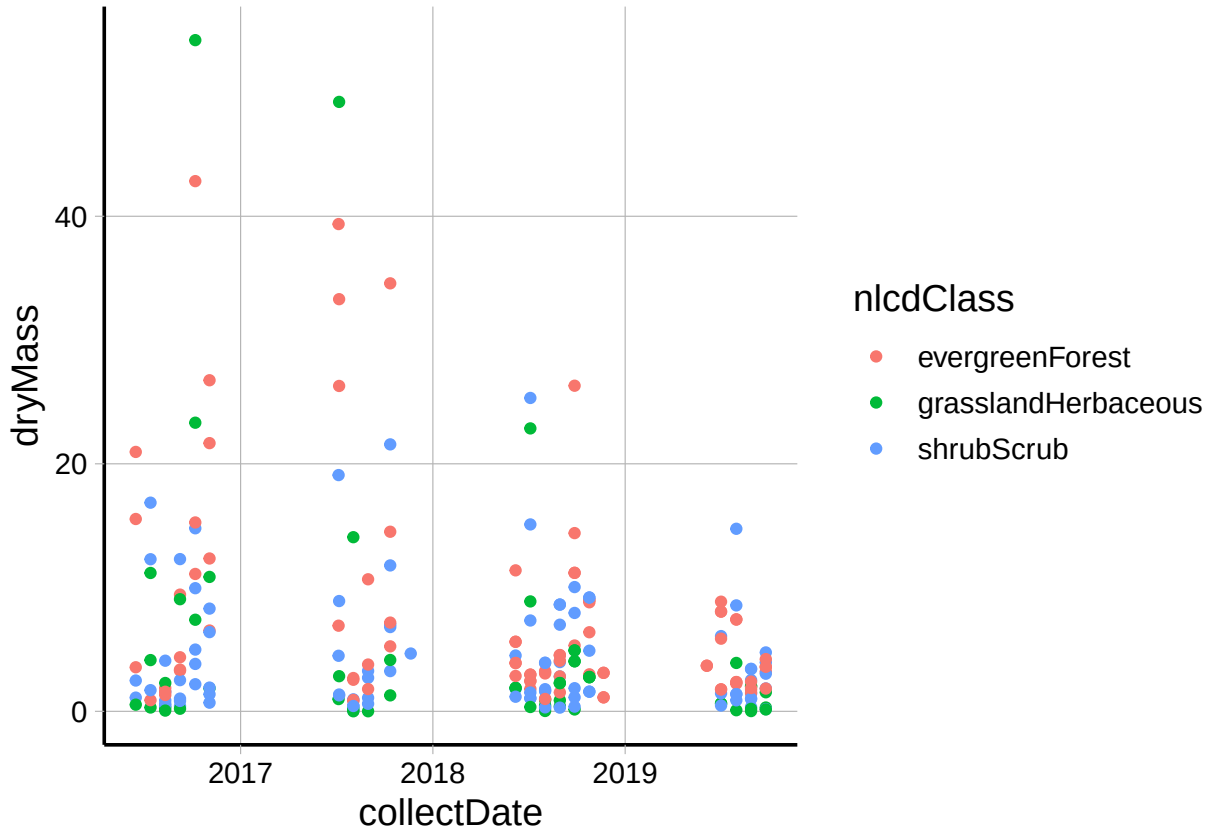
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: Over seasons and between lakes, the variables of interest appear to follow similar trends.

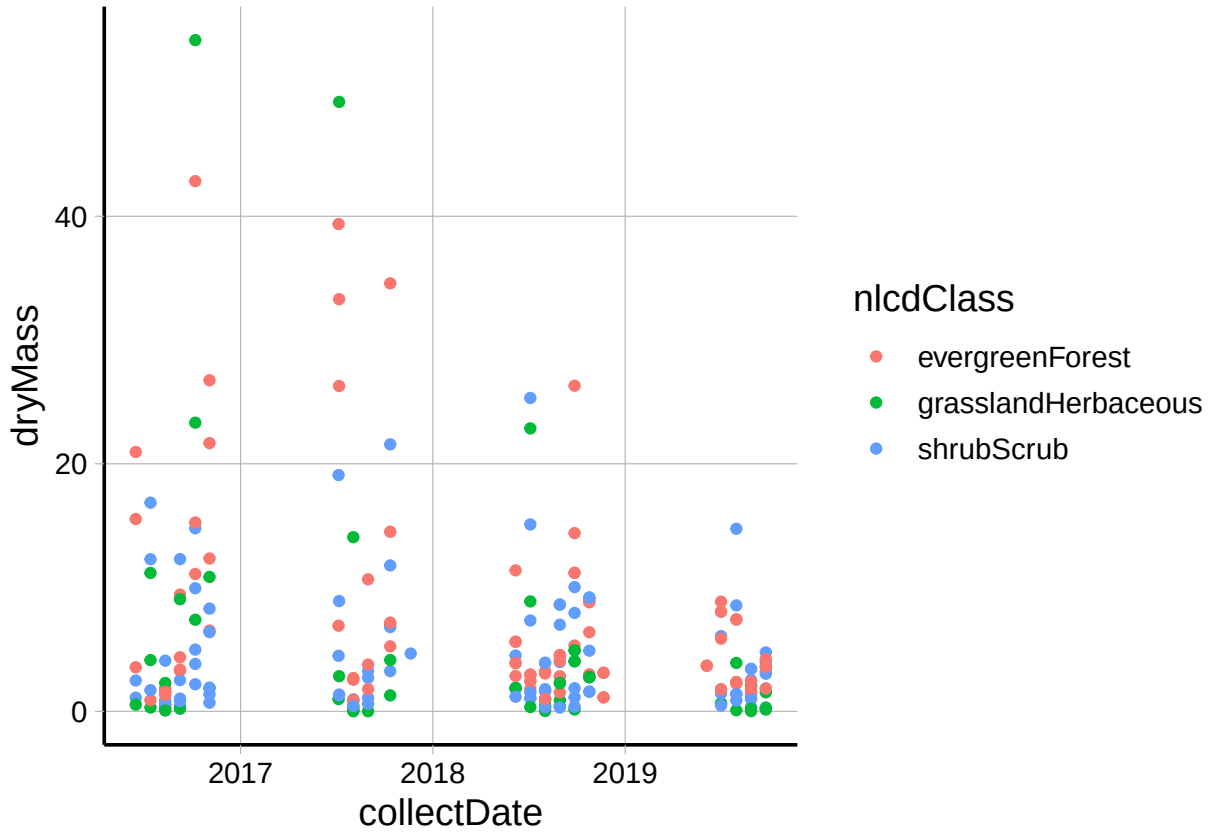
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
#FIX THIS
needles <- ggplot(subset(litter, functionalGroup %in% "Needles"))+
  geom_point(aes(x=collectDate, y=dryMass,color=nlcdClass)) +
  mytheme
print(needles)
```




```
#7
needleSplit <- ggplot(subset(litter, functionalGroup %in% "Needles"))+
  geom_point(aes(x=collectDate, y=dryMass,color=nlcdClass)) +
  mytheme
print(needles)
```



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

Answer: