Assignment 5: Data Visualization

Srishti Mutha

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
getwd()
```

```
## [1] "/Users/cherry/Desktop/EDA/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
## -- 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/cherry/Desktop/EDA/EDA-Spring2023
```

```
library(cowplot)
```

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

```
#install.packages("cowplot")
NTLLTER <- read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv", string
NEON <- read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv", stringsAsFactors = TRUE)

NTLLTER$sampledate <- ymd(NTLLTER$sampledate)
NEON$collectDate <- ymd(NEON$collectDate)

class(NTLLTER$sampledate)
```

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

```
class(NEON$collectDate)
```

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

```
#2
```

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

```r
#3
defaulttheme <- theme_light(base_size = 14) +
  theme(axis.text = element_text(color = "orange"))
```

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.

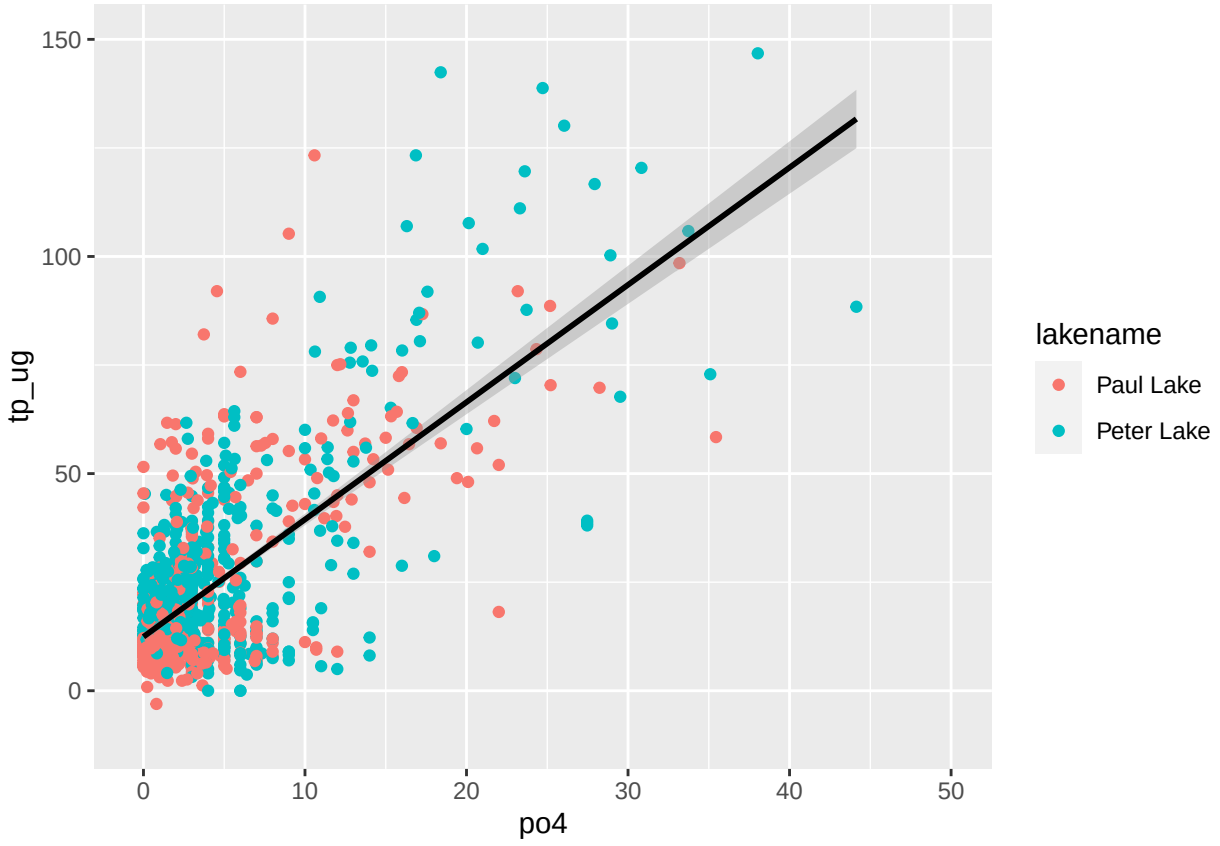
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4
NTLgraph <- ggplot(data = NTLLTER, mapping = aes(x= po4, y = tp_ug, color = lakename)) +
  geom_point() +
  xlim(-0.5,50) +
  ylim(-10,150) +
  geom_smooth(method = lm, color = "black")
  print(NTLgraph)
```

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

```
## Warning: Removed 21947 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21947 rows containing missing values (`geom_point()`).
```

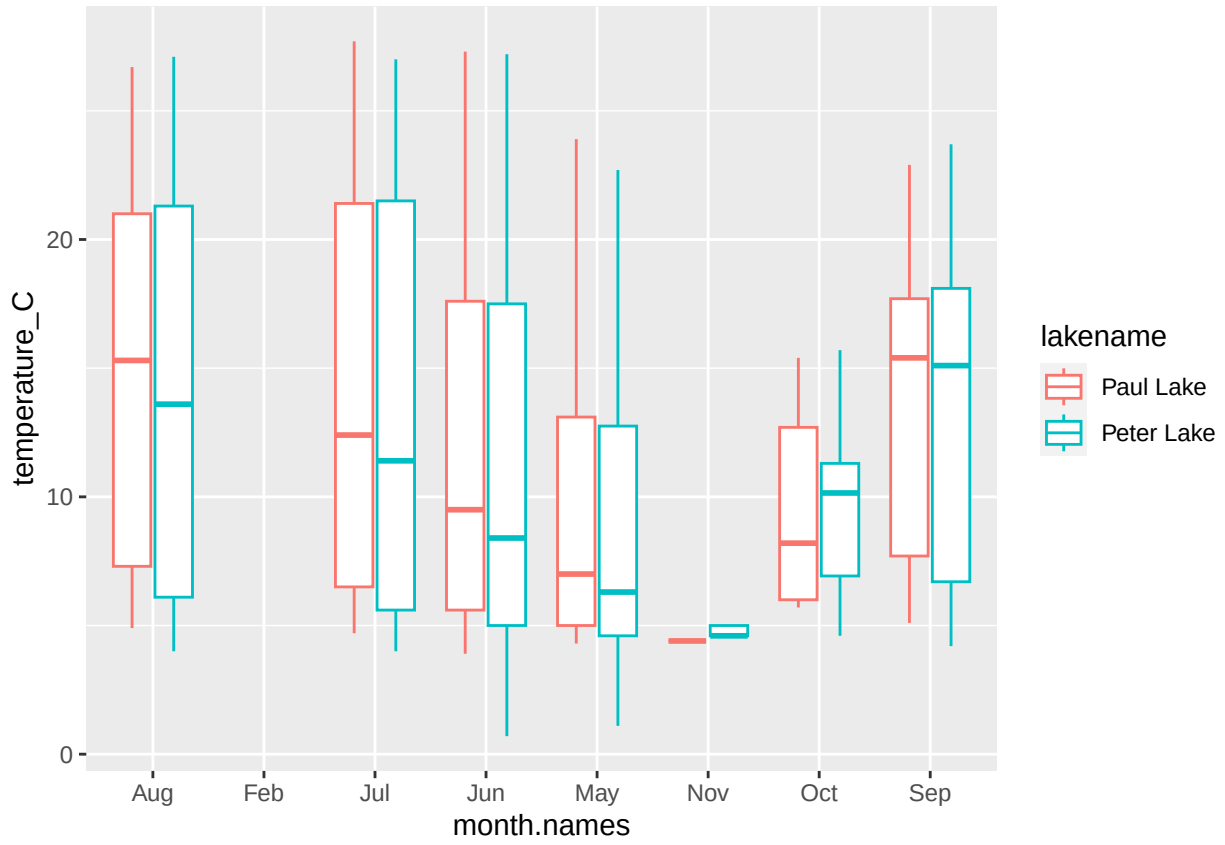


5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

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

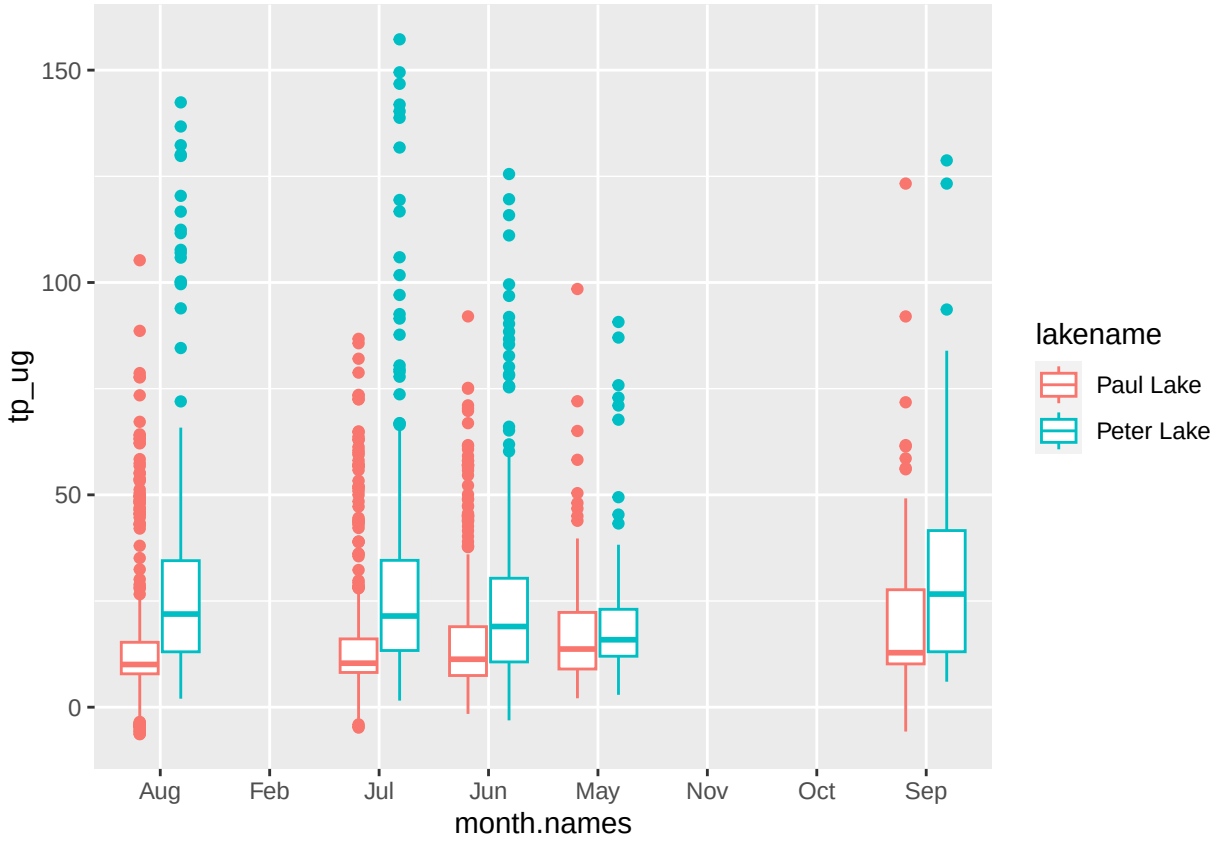
```
#5
month.factor <- factor(NTLLTER$month, levels = c(1:12))
month.names <- month.abb[month.factor]
NTLTEMP <- ggplot(data = NTLLTER, mapping = aes(x= month.names, y = temperature_C)) +
  geom_boxplot( aes(color = lakename))
print(NTLTEMP)
```

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



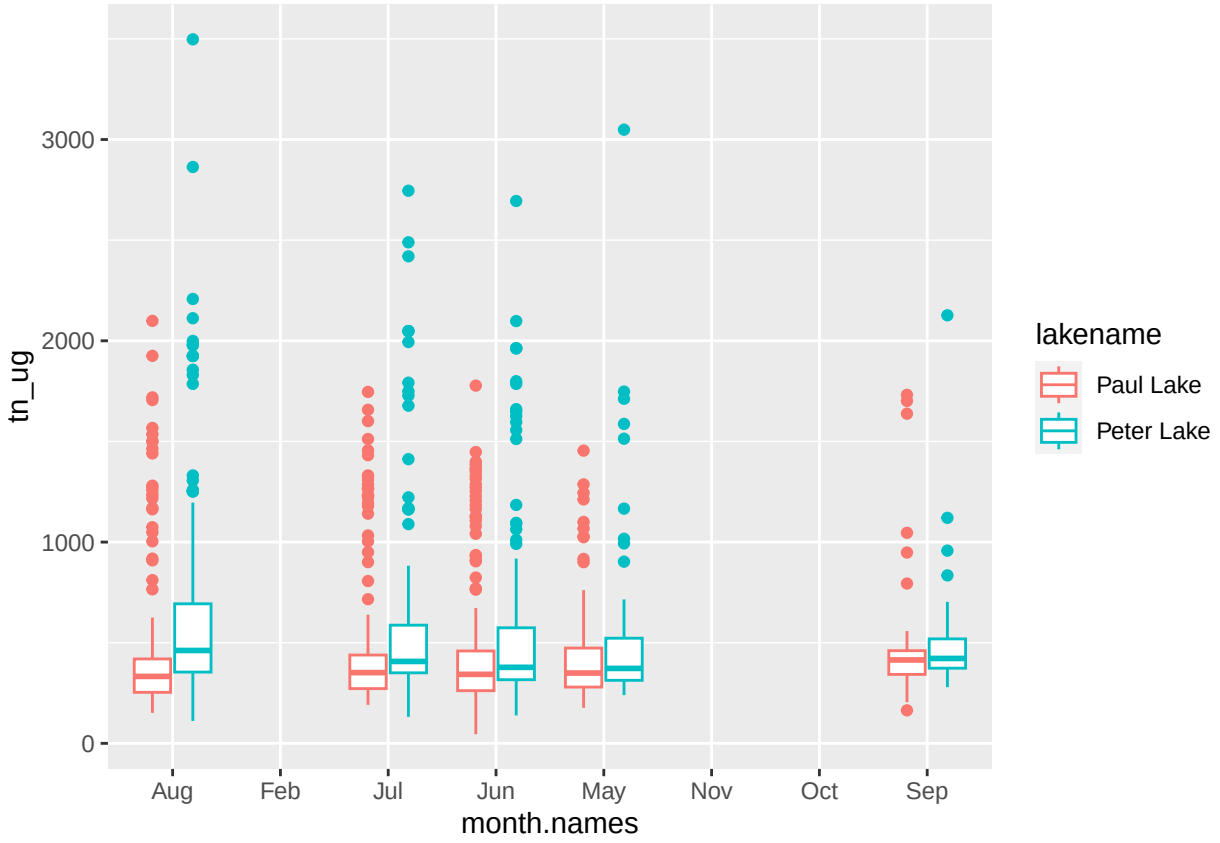

```
NTLTP <- ggplot(data = NTLLTER, mapping = aes(x= month.names, y = tp_ug)) +
  geom_boxplot(aes(color = lakename))
print(NTLTP)
```

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




```
NTLTN <- ggplot(data = NTLLTER, mapping = aes(x= month.names, y = tn_ug)) +
  geom_boxplot(aes(color = lakename))
print(NTLTN)
```

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

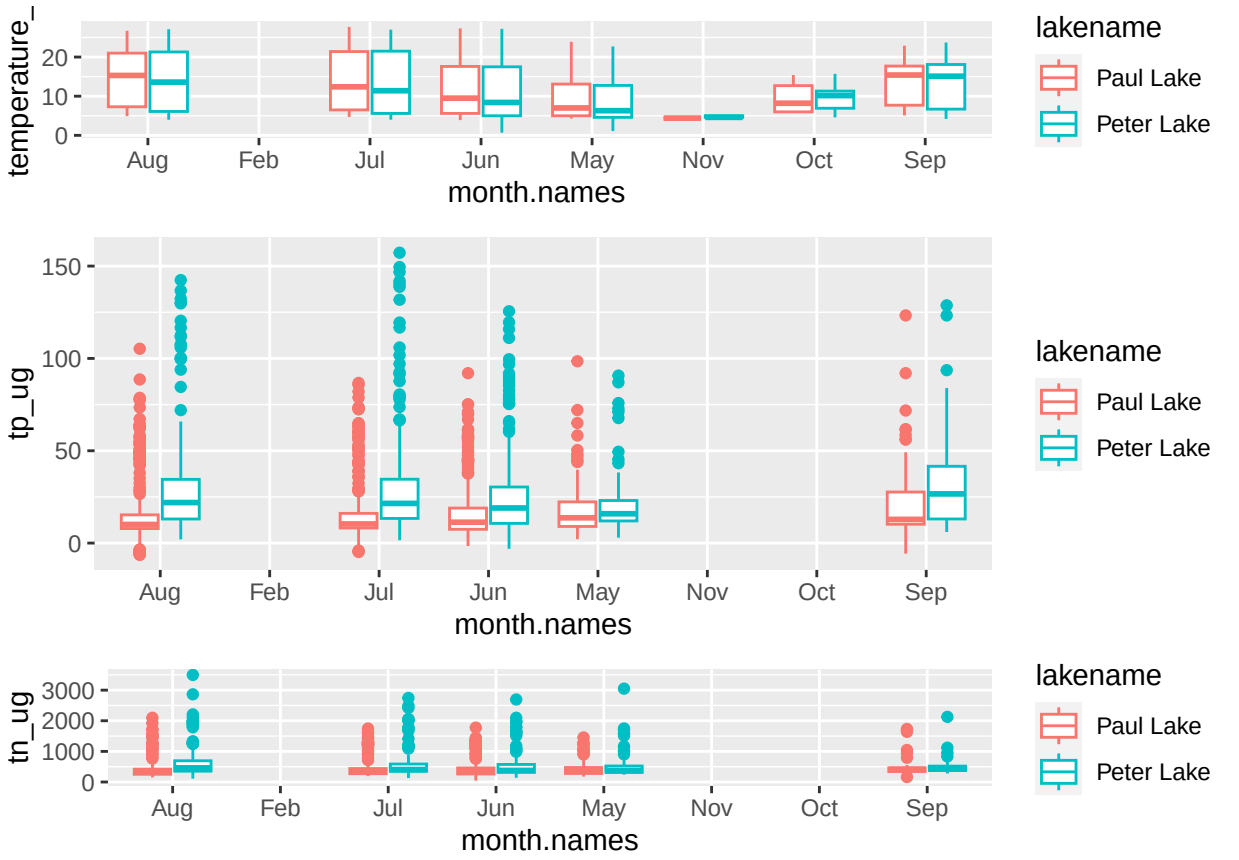



```
plot_grid(NTLTEMP, NTLTP, NTLTN, nrow = 3, align = "h", rel_heights = c(1,2) )
```

```
## 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()‘).
```



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

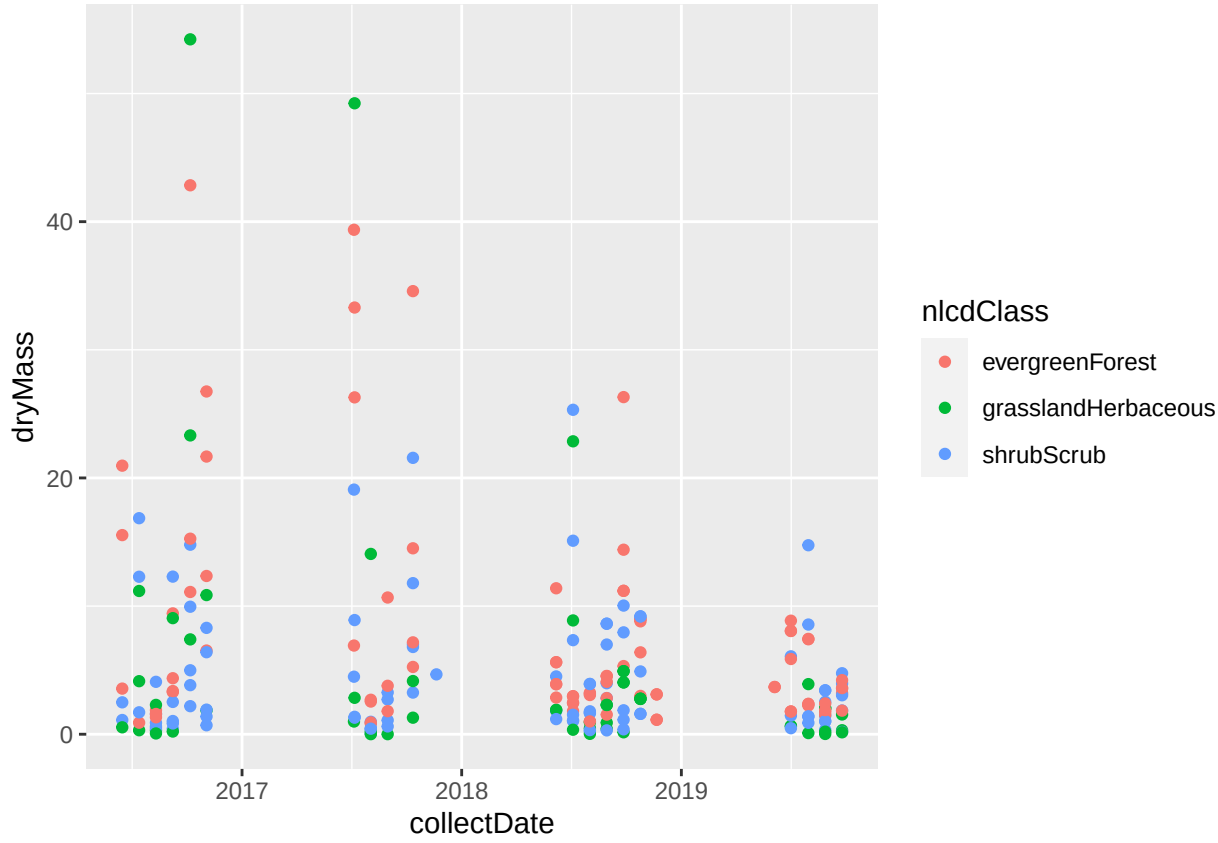
Answer:

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)

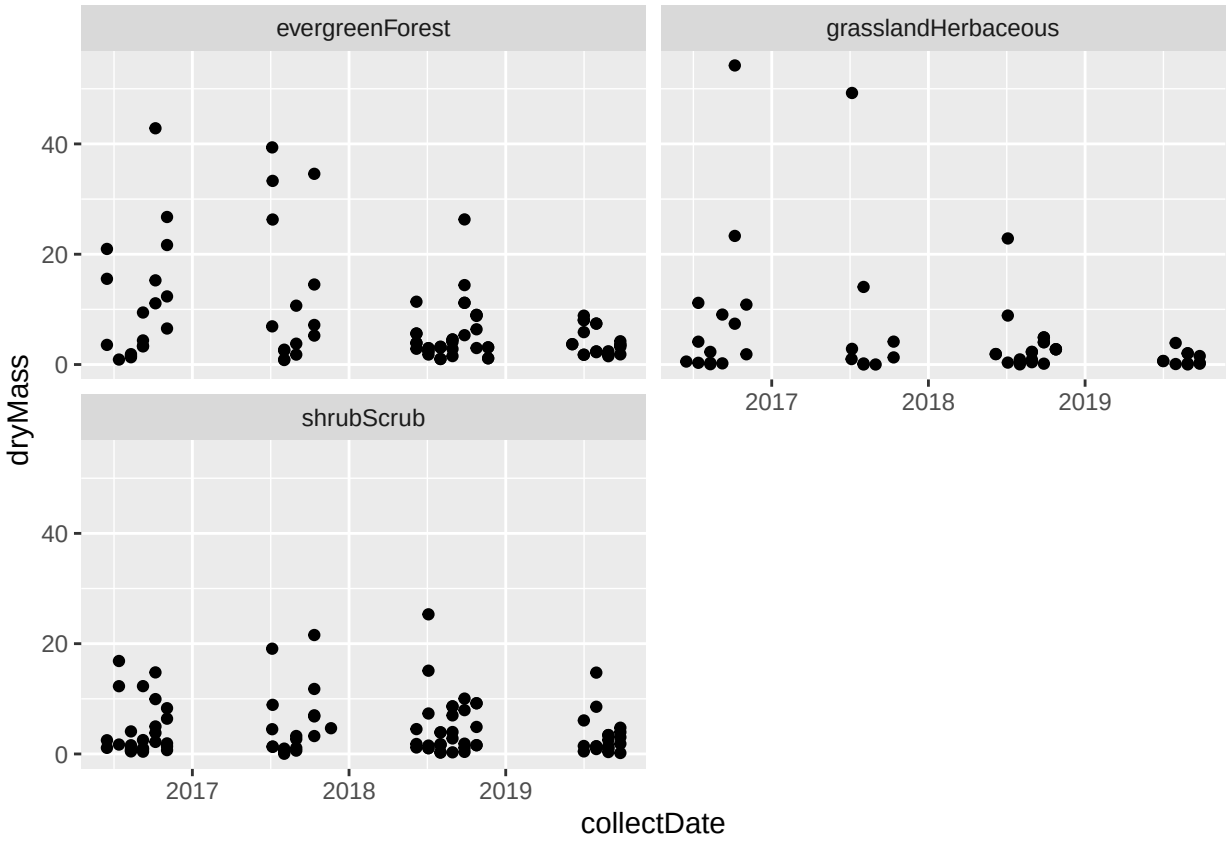
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Neonsubset <- NEON %>%
  filter(functionalGroup == "Needles")

NEON_needle_graph <- ggplot(data = Neonsubset, mapping = aes(y = dryMass, x = collectDate, color = nlcd
  geom_point()
print(NEON_needle_graph)
```




```
#7
NEON_needle_facted <- ggplot(data = Neonsubset, mapping = aes(y = dryMass, x = collectDate)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 2)
print(NEON_needle_facted)
```



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

Answer: 7 is more effective