Assignment 5: Data Visualization

Cristiana Falvo

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., "Salk_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Tuesday, February 11 at 1:00 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 (tidy and gathered) and the processed data file for the Niwot Ridge litter dataset.

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

```
#1
getwd()
```

```
## [1] "/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_2020/Assignments"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.3.0

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

```
library(cowplot)
```

```
##
## ********************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ********************************************************
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
chem_nutrients <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_2
chem_physics <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_202
nutrients_wide <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_2
nutrients_gathered <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analyti
NIWOT <- read.csv("/Users/cristiana/Documents/Duke/DataAnalytics/Environmental_Data_Analytics_2020/Data/


#2
class(chem_nutrients$sampledate)
```

```
## [1] "factor"
```

```
class(chem_physics$sampledate)
```

```
## [1] "factor"
```

```
class(nutrients_gathered$sampledate)
```

```
## [1] "factor"
```

```
class(nutrients_wide$sampledate)
```

```
## [1] "factor"
```

```
class(NIWOT$collectDate)
```

```
## [1] "factor"
```

```
chem_nutrients$sampledate <- as.Date(chem_nutrients$sampledate, format = "%Y/%m/%d")
chem_physics$sampledate <- as.Date(chem_physics$sampledate, format = "%Y-%m-%d")
nutrients_wide$sampledate <- as.Date(nutrients_wide$sampledate, format = "%Y/%m/%d")
nutrients_gathered$sampledate <- as.Date(nutrients_gathered$sampledate, format = "%Y/%m/%d")
NIWOT$collectDate <- as.Date(NIWOT$collectDate, format = "%Y/%m/%d")
```

Define your theme

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

```
#install.packages("devtools")
devtools::install_github("katiejolly/nationalparkcolors")
```

```
## Skipping install of 'nationalparkcolors' from a github remote, the SHA1 (df8cd15d) has not changed si
##   Use `force = TRUE` to force installation
```

```r
library(nationalparkcolors)

pal <- park_palette("SmokyMountains")

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

mytheme2 <- list(mytheme, scale_color_manual((values = pal)))

theme_set(mytheme)
# when I did this and tried to make a plot it said many elements were missing
# went back to default for now
```

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.

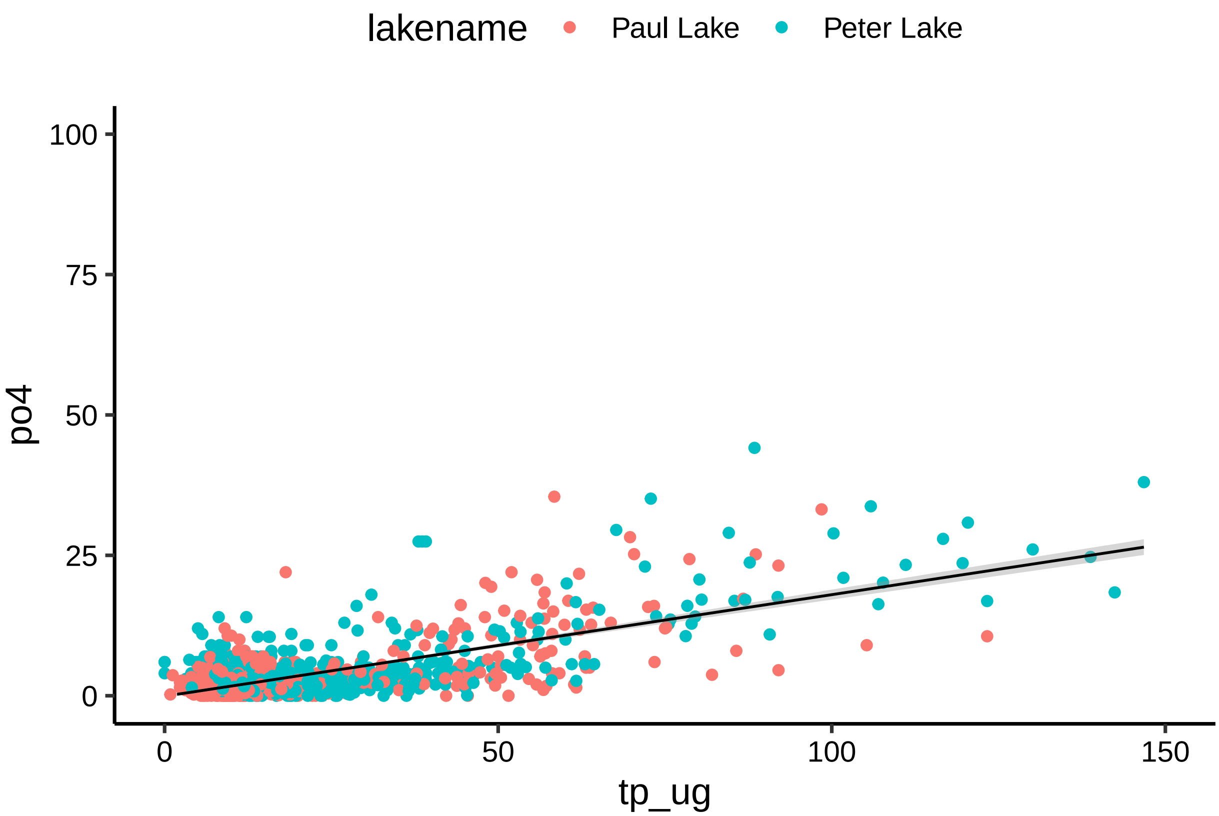
4. [NTL-LTER] Plot total phosphorus by phosphate, with separate aesthetics for Peter and Paul lakes. Add a line of best fit and color it black. Adjust your axes to hide extreme values.

```r
Phosphorus_Phosphate <-
  ggplot(nutrients_wide, aes(x = tp_ug, y = po4, color = lakename)) +
  geom_point() +
  xlim(0, 150) +
  ylim(0, 100) +
  geom_smooth(method = lm, color = "black", size = .5)
print(Phosphorus_Phosphate)
```

```
## Warning: Removed 1346 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1346 rows containing missing values (geom_point).
```

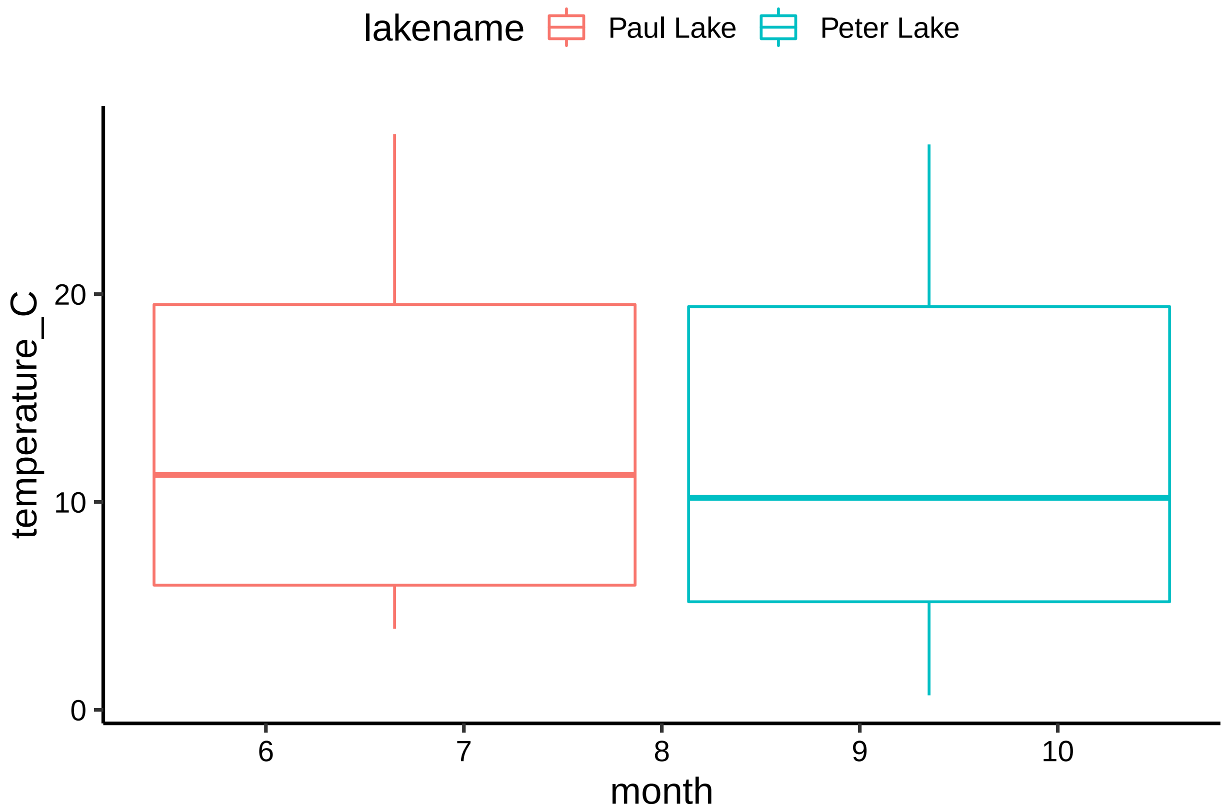
```
## Warning: Removed 1 rows containing missing values (geom_smooth).
```



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.

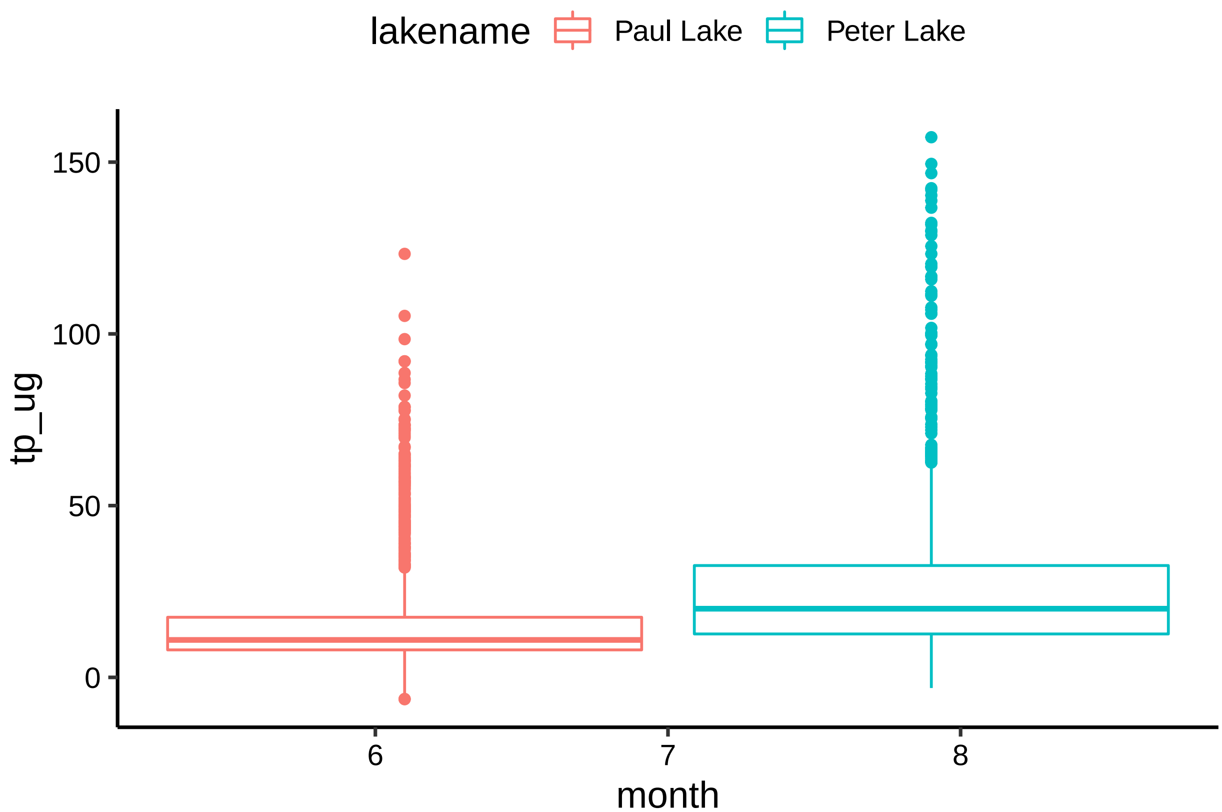
```
# 5a
temp_box <-
  ggplot(chem_nutrients) +
  geom_boxplot(aes(x = month, y = temperature_C, color = lakename))
print(temp_box)
```

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



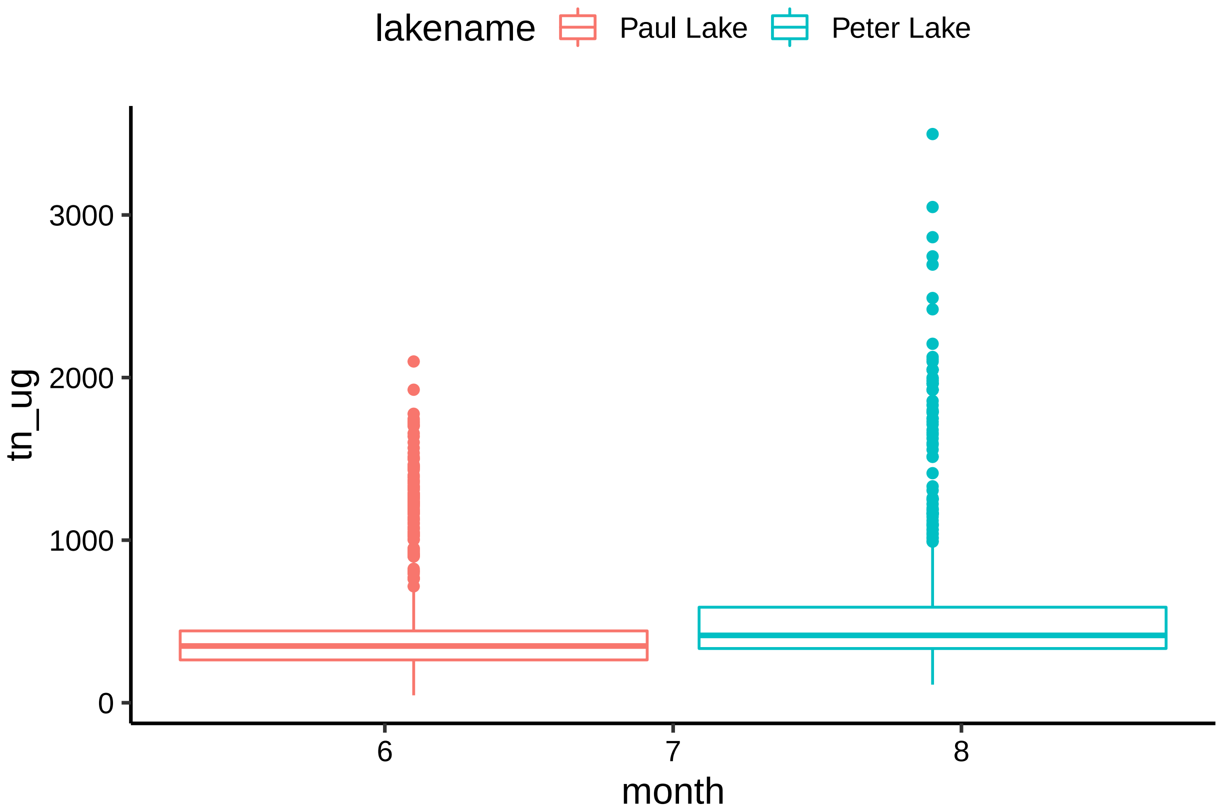

```r
# 5b
TP_box <-
  ggplot(chem_nutrients) +
  geom_boxplot(aes(x = month, y = tp_ug, color = lakename))
print(TP_box)
```

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




```
# 5c
TN_box <-
  ggplot(chem_nutrients) +
  geom_boxplot(aes(x = month, y = tn_ug, color = lakename))
print(TN_box)
```

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

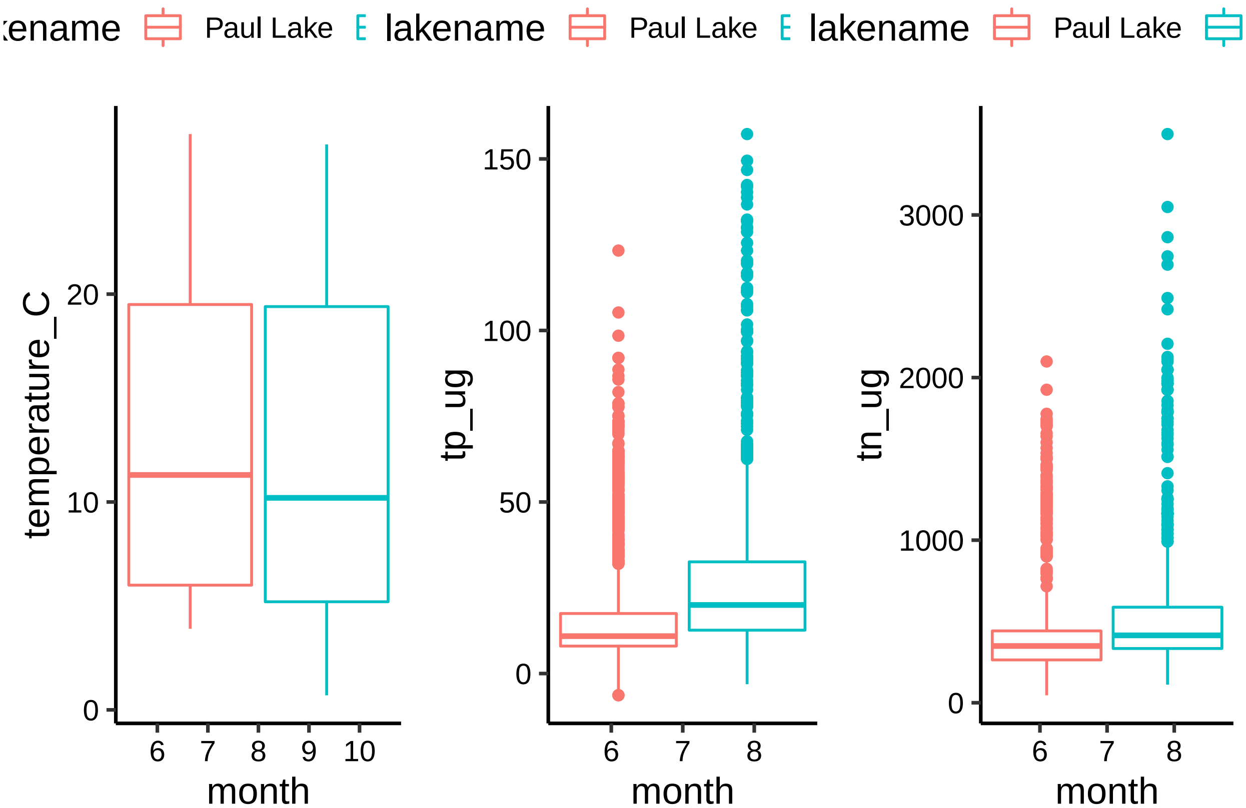



```
# cowplot

all_box <- plot_grid(temp_box, TP_box, TN_box, ncol = 3, nrow = 1)
```

```
## 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).
```

```
print(all_box)
```



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

> Answer: The nutrient concentrations are higher in Peter Lake than in Paul Lake. Change over time is hard to discern from these box plots..

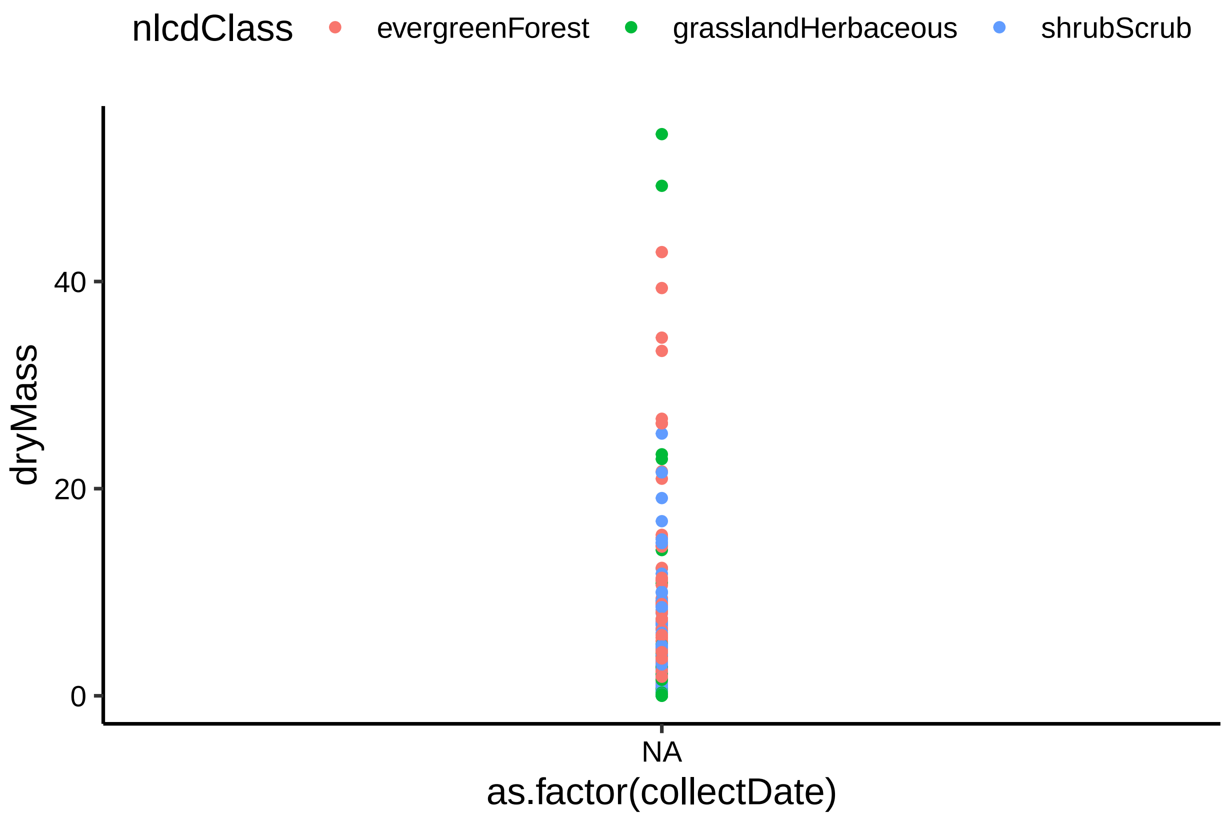
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes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

```
class(NIWOT$collectDate)
```

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

```
S <- subset(NIWOT, functionalGroup == "Needles")

Needle_plot <-
  ggplot(S, aes(x = as.factor(collectDate), y = dryMass, color = nlcdClass))  +
  geom_point()
print(Needle_plot)
```




```
Needle_plot_facet <-
  ggplot(S, aes(x = as.factor(collectDate), y = dryMass))  +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3)
print(Needle_plot_facet)
```


```
## don't know why collectDate can't plot on the x axis
  # without as.factor it gives error message
  # with as.factor it plots all dates as one (NA)
```

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

> Answer: I think looking at the plots separately in 3 facets is most effective because they are lined up for comparison but you can also see each individual dataset more clearly than when the colors are overlapping.