Assignment 5: Data Visualization

Rachael Stephan

Fall 2024

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. Read in 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 in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

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

```
#1
#load in library
library(tidyverse); library(lubridate); library(cowplot); library(here)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
##
##
## here() starts at /home/guest/Duke_R/EDE_Fall2024
```

```
#verify home directory
getwd()
```

```
## [1] "/home/guest/Duke_R/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/Duke_R/EDE_Fall2024"
```

```
#load in datasets
pp_chem_nutrients <- read.csv(
  file = here("./Data/Processed/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
  stringsAsFactors = TRUE)

neon_niwot_litter <- read.csv(
  file = here("./Data/Processed/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
#check date formats
str(pp_chem_nutrients)
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : Factor w/ 2 levels "Paul Lake","Peter Lake": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year4          : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ daynum         : int  148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : Factor w/ 1103 levels "1984-05-27","1984-05-28",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ depth          : num  0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num  14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num  9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num  1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num  1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```r
str(neon_niwot_litter)
```

```
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : Factor w/ 12 levels "NIWO_040","NIWO_041",..: 9 8 9 11 7 7 4 4 4 4 ...
##  $ trapID          : Factor w/ 15 levels "NIWO_040_139",..: 11 10 11 13 9 9 5 5 5 5 ...
##  $ collectDate     : Factor w/ 24 levels "2016-06-16","2016-07-14",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ functionalGroup : Factor w/ 8 levels "Flowers","Leaves",..: 6 5 8 6 4 2 2 6 7 8 ...
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 1 1 1 ...
##  $ subplotID       : int  31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num  40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num  -106 -106 -106 -106 -106 ...
##  $ elevation       : num  3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : Factor w/ 3 levels "evergreenForest",..: 3 1 3 1 3 3 2 2 2 2 ...
##  $ plotType        : Factor w/ 1 level "tower": 1 1 1 1 1 1 1 1 1 1 ...
##  $ geodeticDatum   : Factor w/ 1 level "WGS84": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#change dates from factors to correct format
pp_chem_nutrients$sampledate <- ymd(pp_chem_nutrients$sampledate)
neon_niwot_litter$collectDate <- ymd(neon_niwot_litter$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

```r
#3
#build custom plot theme

custom_theme <-
  theme_bw(base_size = 10)+
  theme(axis.title = element_text(size = 10, hjust = 0.5),
        plot.title.position = "panel",
        legend.box = "vertical",
        legend.location = "plot",
        axis.ticks = element_line(color = "black", linewidth = 0.5),
        axis.gridlines = element_line(color = "grey"))
```

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.

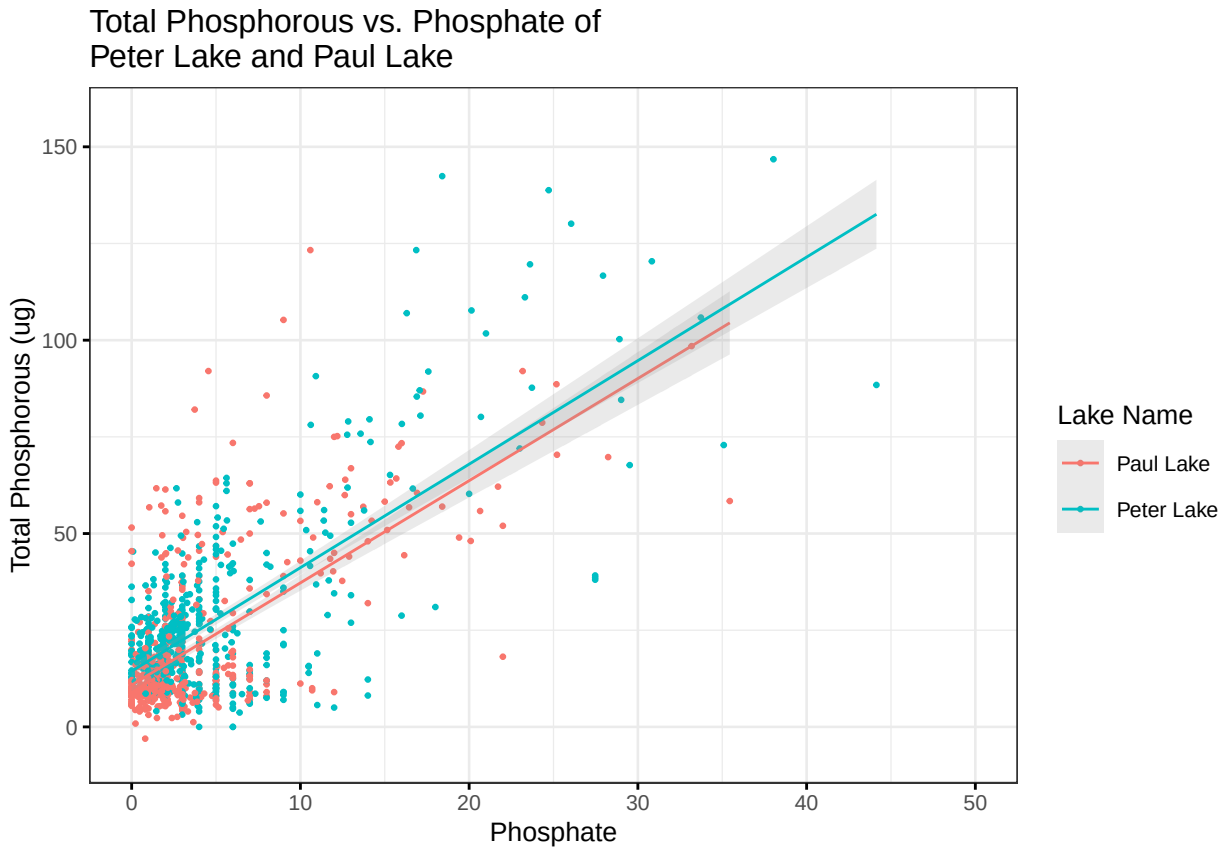
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```r
#4
#create plot
totp_phos <- ggplot(pp_chem_nutrients,
                    aes(x = po4, y = tp_ug, colour = lakename))+
  geom_point(size = 0.5)+
  labs(title = "Total Phosphorous vs. Phosphate of\nPeter Lake and Paul Lake",
       y = "Total Phosphorous (ug)",
       x = "Phosphate",
       color = "Lake Name")+
  xlim(0,50)+
  geom_smooth(method = lm, size = 0.5, alpha = 0.2)+
  custom_theme
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
totp_phos
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 21947 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).

## Warning in plot_theme(plot): The ‘axis.gridlines‘ theme element is not defined
## in the element hierarchy.

## Warning: Removed 21947 rows containing missing values or values outside the scale range
## (‘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.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

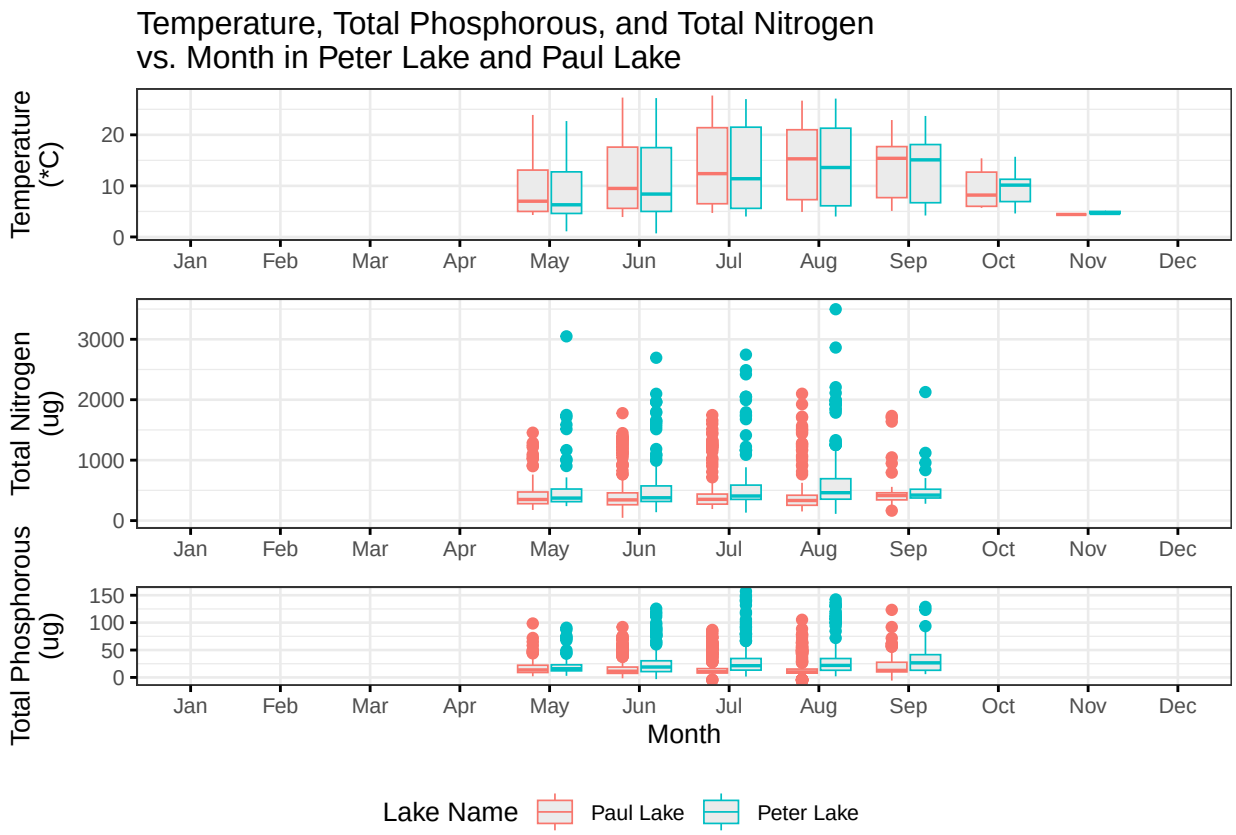
```
#change month to a factor with text labels
pp_chem_nutrients$month <- factor(pp_chem_nutrients$month,
       levels = 1:12,
       labels = month.abb)
```

```
#5
#create temperature plot
temp_plot <- ggplot(
  pp_chem_nutrients,
  aes(x = month,
      y = temperature_C,
      colour = lakename))+
  geom_boxplot(fill = "grey92", size = 0.3)+
  scale_x_discrete(drop=FALSE)+
  labs(title = "Temperature, Total Phosphorous, and Total Nitrogen\nvs. Month in Peter Lake and Paul La
       x = "Month",
```

```
       y = "Temperature\n(*C)",
       color = "Lake Name")+
  custom_theme+
  theme(axis.title.x = element_blank(),
        legend.position = "none")

#create a TN plot
tn_plot <- ggplot(
  pp_chem_nutrients,
  aes(x = month,
      y = tn_ug,
      colour = lakename))+
  geom_boxplot(fill = "grey92", size = 0.3)+
  scale_x_discrete(drop=FALSE)+
  labs(y = "Total Nitrogen\n(ug)",
       color = "Lake Name")+
  custom_theme+
  theme(axis.title.x = element_blank(),
        legend.position = "none",
        axis.title.y = element_text(margin = margin(r = 5)))

#create a TP plot
tp_plot <- ggplot(
  pp_chem_nutrients,
  aes(x = month,
      y = tp_ug,
      colour = lakename))+
  geom_boxplot(fill = "grey92", size = 0.3)+
  scale_x_discrete(drop=FALSE)+
  labs(x = "Month",
       y = "Total Phosphorous\n(ug)",
       color = "Lake Name")+
  custom_theme+
  theme(legend.position = "bottom")

#combine plots using cowplot
plot_grid(temp_plot, tn_plot, tp_plot, nrow = 3,
  align = "v")
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

## Warning in plot_theme(plot): The ‘axis.gridlines‘ theme element is not defined
## in the element hierarchy.

## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

## Warning in plot_theme(plot): The ‘axis.gridlines‘ theme element is not defined
## in the element hierarchy.

## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning in plot_theme(plot): The 'axis.gridlines' theme element is not defined
## in the element hierarchy.
```



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

> Answer: Temperatures slowly increase over the summer, reaching a peak in September. Aftwerwards, there is a decrease. The interquartile range also gets larger in the summer. The smallest interquartile range occurs in November. The mean temperatures in Paul Lake are quicker to change compared to Peter Lake (i.e., Paul Lake has the lowest and highest median temperatures). The total nitrogen remains relatively stable over the course of the sampling period. However, Peter Lake has more nitrogen than Paul Lake. This trend remains the same for total phosphorous.

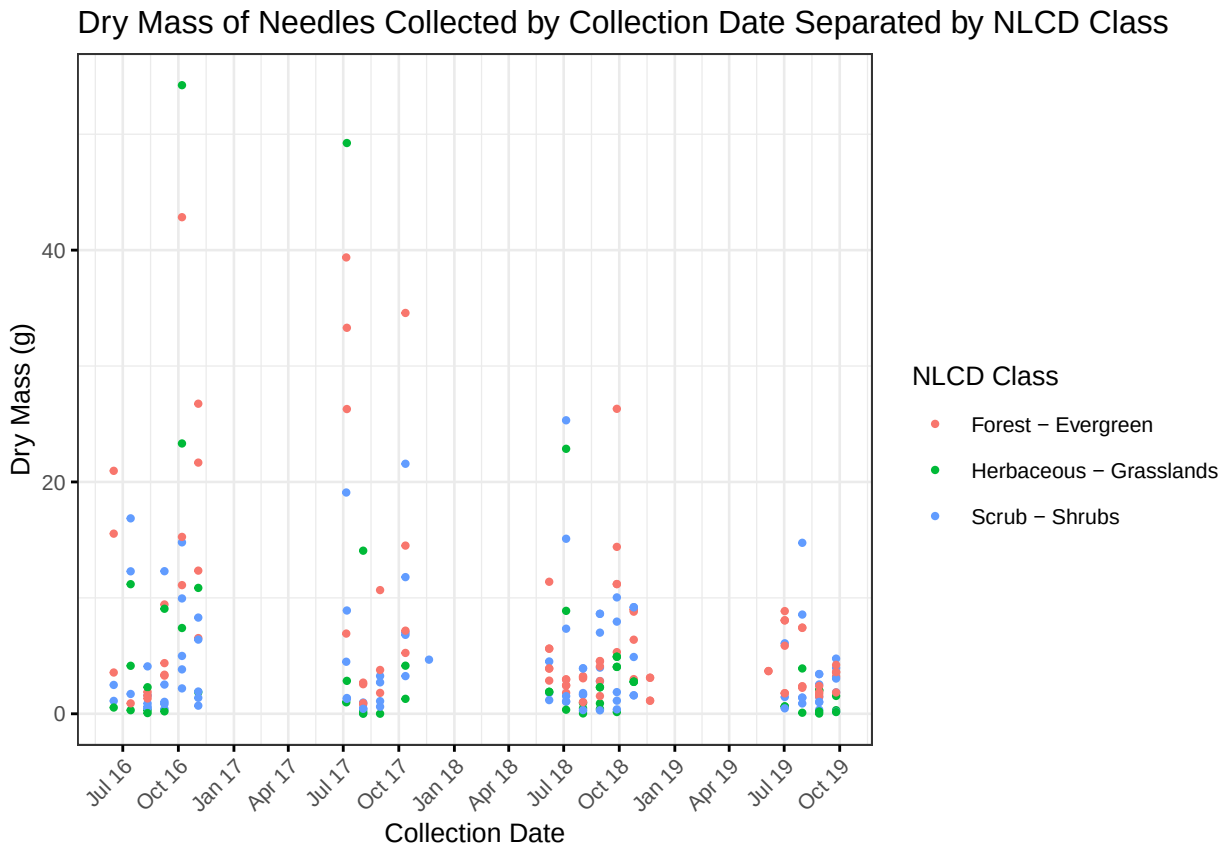
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
#create needles plot
needles_plot <- ggplot(
  subset(neon_niwot_litter, functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass, color = nlcdClass))+
  geom_point(size = 0.8)+
```

```
  labs(y = "Dry Mass (g)",
       x = "Collection Date",
       title = "Dry Mass of Needles Collected by Collection Date Separated by NLCD Class",
       color = "NLCD Class")+
  scale_x_date(date_breaks = "3 months", date_labels = "%b %y") +
  custom_theme+
  scale_color_discrete(labels=c("Forest - Evergreen", "Herbaceous - Grasslands", "Scrub - Shrubs"))+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))

needles_plot
```

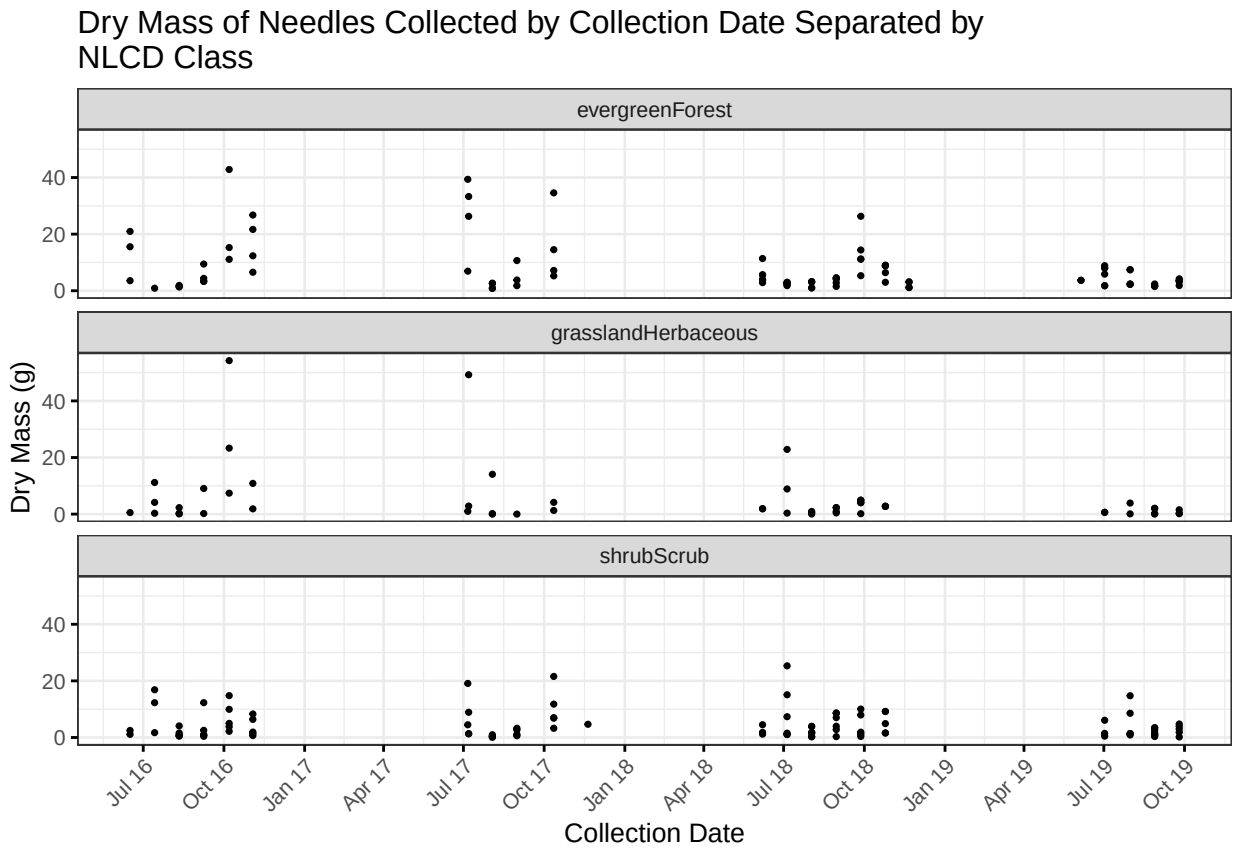
```
## Warning in plot_theme(plot): The 'axis.gridlines' theme element is not defined
## in the element hierarchy.
```




```
#7
#
needles_plot_facet <-  ggplot(
  subset(neon_niwot_litter, functionalGroup == "Needles"),
  aes(x = collectDate, y = dryMass))+
  geom_point(size = 0.6)+
  labs(y = "Dry Mass (g)",
       x = "Collection Date",
       title = "Dry Mass of Needles Collected by Collection Date Separated by\nNLCD Class",
       color = "NLCD Class")+
```

```
  scale_x_date(date_breaks = "3 months", date_labels = "%b %y") +
  facet_wrap(facets =vars(nlcdClass),
             nrow = 3,
             labeller = labeller(group = c("evergreenForest" = "Forest - Evergreen",
                                           "grasslandHerbaceous" = "Herbaceous - Grasslands",
                                           "shrubScrub" = "Scrub - Shrubs")))+
  custom_theme+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))

needles_plot_facet
```

```
## Warning in plot_theme(plot): The 'axis.gridlines' theme element is not defined
## in the element hierarchy.
```



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

Answer: I think plot 6 is more effective. Plotting points on the same set of axes makes it easier to compare between different groups. There are few enough entries in this dataset that the individual data points can still be seen. Also, facetting the graphs compresses the y axes, which makes it harder to distinguish between values are lower ranges where the data is concentrated.