Assignment 5: Data Visualization

Lauren Shohan

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
library(tidyverse);library(lubridate);library(here);library(ggthemes)
```

```
## -- 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
## here() starts at /home/guest/EDE_Fall2024
```

```
library(cowplot)
```

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

```
getwd()
```

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

```
here()
```

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

```
PeterPaul.Nutrients.Data <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"), string

NiwotRide.Litter.Data <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"), stringsAsFactors = TRUE


#2
class(PeterPaul.Nutrients.Data$sampledate) #factor
```

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

```
class(NiwotRide.Litter.Data$collectDate) #factor
```

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

```
#changing the year columns to date
PeterPaul.Nutrients.Data$sampledate <- ymd(PeterPaul.Nutrients.Data$sampledate)
NiwotRide.Litter.Data$collectDate <- ymd(NiwotRide.Litter.Data$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

```
#3
mytheme <- theme_classic(base_size = 12) + #overall size
  theme(axis.text = element_text(color = "black", face = "bold"), #making axis #s black and bold
        plot.background = element_rect(fill = "darkseagreen"), #making background green
        panel.grid.major = element_line(color = "lightgray", linetype = "solid") #adding grid lines
        )
#set my theme into default for entire sheet
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.

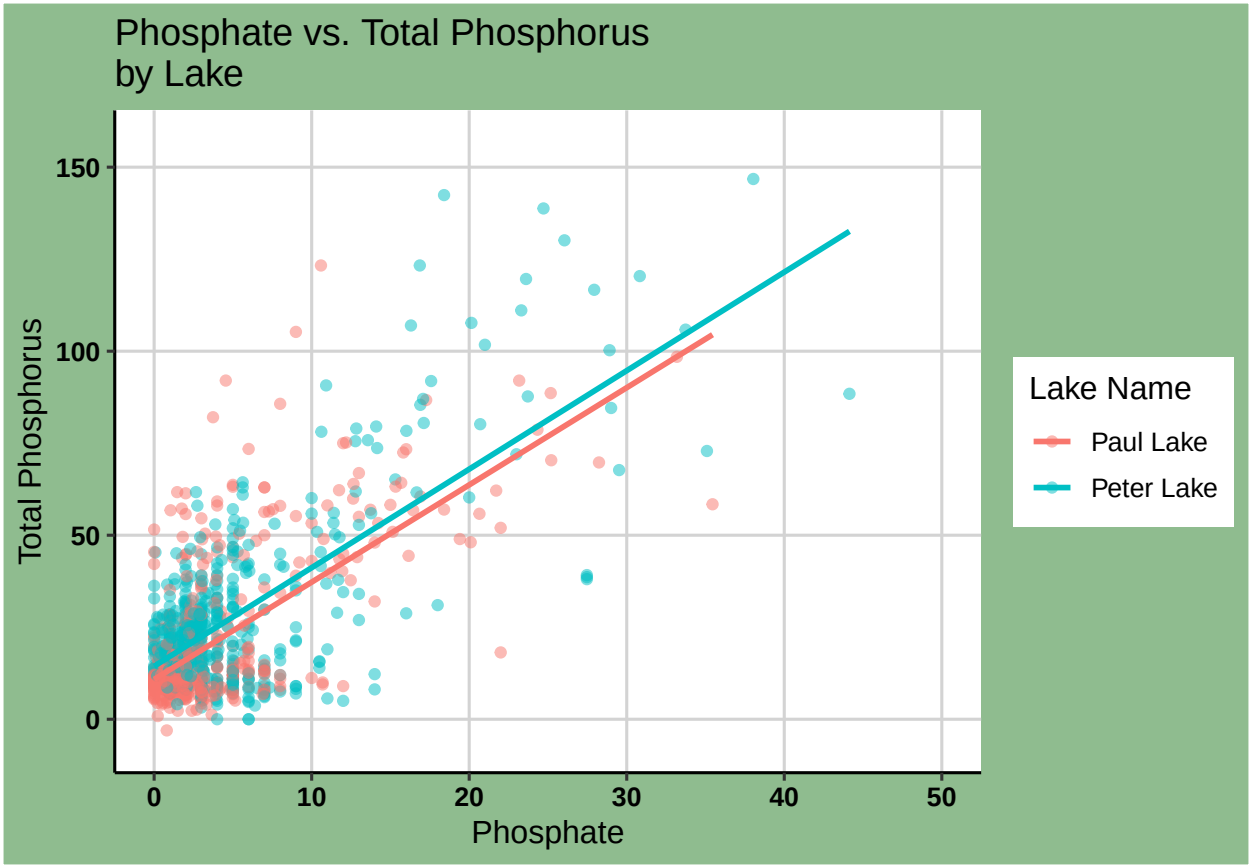
4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

plot04_peterpaul <- PeterPaul.Nutrients.Data %>%
  ggplot(aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = 'lm', se = FALSE) + #creates best fit line
  xlim(0,50) + #limited my x values from 0 to 50
  labs(title = ("Phosphate vs. Total Phosphorus\nby Lake"), # \n separates the line
    x = "Phosphate", y = "Total Phosphorus", color = "Lake Name")

plot04_peterpaul
```

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

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

```
## 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`.

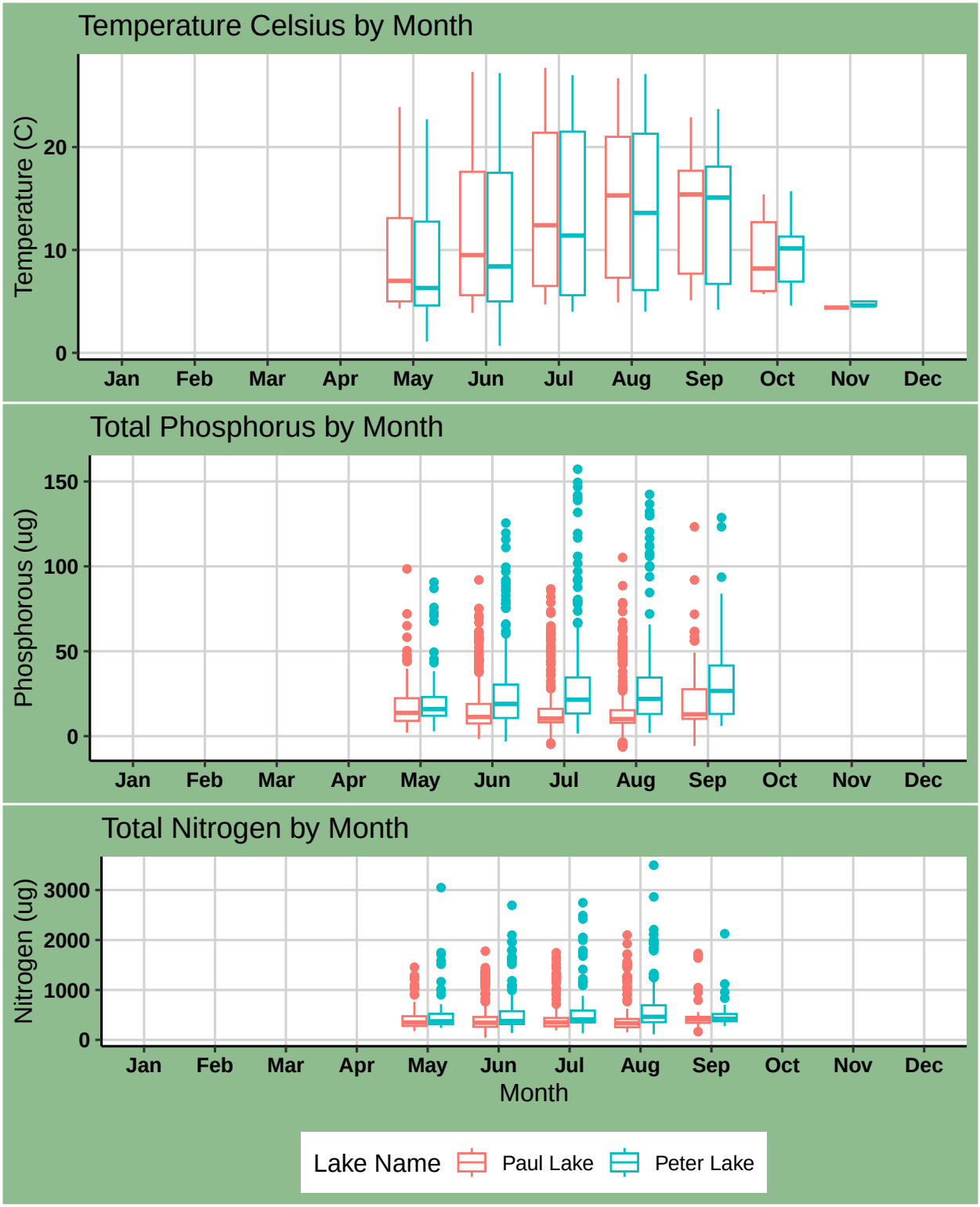
```
#5

PeterPaul.Nutrients.Data$month <- factor(PeterPaul.Nutrients.Data$month,
       levels = 1:12, #12 months
       labels = month.abb) #changing the month column to months abbreviations


#temperature C boxplot
Temperature_boxplot <- PeterPaul.Nutrients.Data %>%
  ggplot(aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = "Month", drop = FALSE) + #helps to add jan and dec
  labs(title = 'Temperature Celsius by Month', y = 'Temperature (C)') +
  theme(legend.position = 'none', axis.title.x = element_blank())
        #elementblank to get rid x labels
```

```r
#total phosphorus
Tp_ug_boxplot <- PeterPaul.Nutrients.Data %>%
  ggplot(aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = "Month", drop = FALSE) +
  labs(title = 'Total Phosphorus by Month', y = 'Phosphorous (ug)') +
  theme(legend.position = "none", axis.title.x = element_blank())

#nitrogen
Tn_ug_boxplot <- PeterPaul.Nutrients.Data %>%
  ggplot(aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale_x_discrete(name = "Month", drop = FALSE) +
  theme(legend.position = 'bottom') +
  labs(title = 'Total Nitrogen by Month', x = 'Month',
       y = 'Nitrogen (ug)', color = 'Lake Name')

#plotting all three box plots
plot_grid(Temperature_boxplot,Tp_ug_boxplot,Tn_ug_boxplot, nrow = 3)
```

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

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

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



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

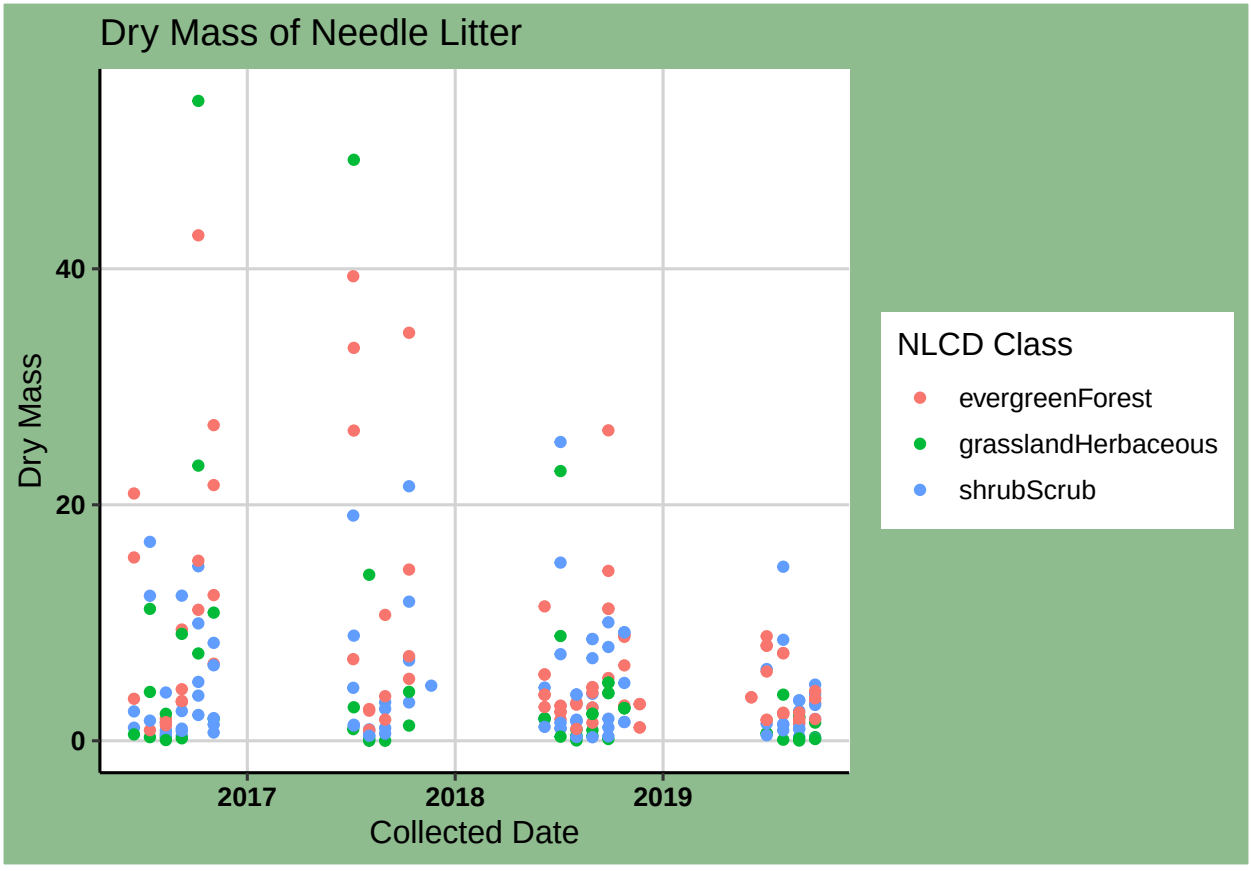
Answer: During the warmer summer months, the variables increase and there is a nice bell curve from when the weather starts to warm up and when it starts to cool down. However, Peter Lake seems to have higher observations of Phosphorous and Nitrogen than Paul Lake during these

warmer months.

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)

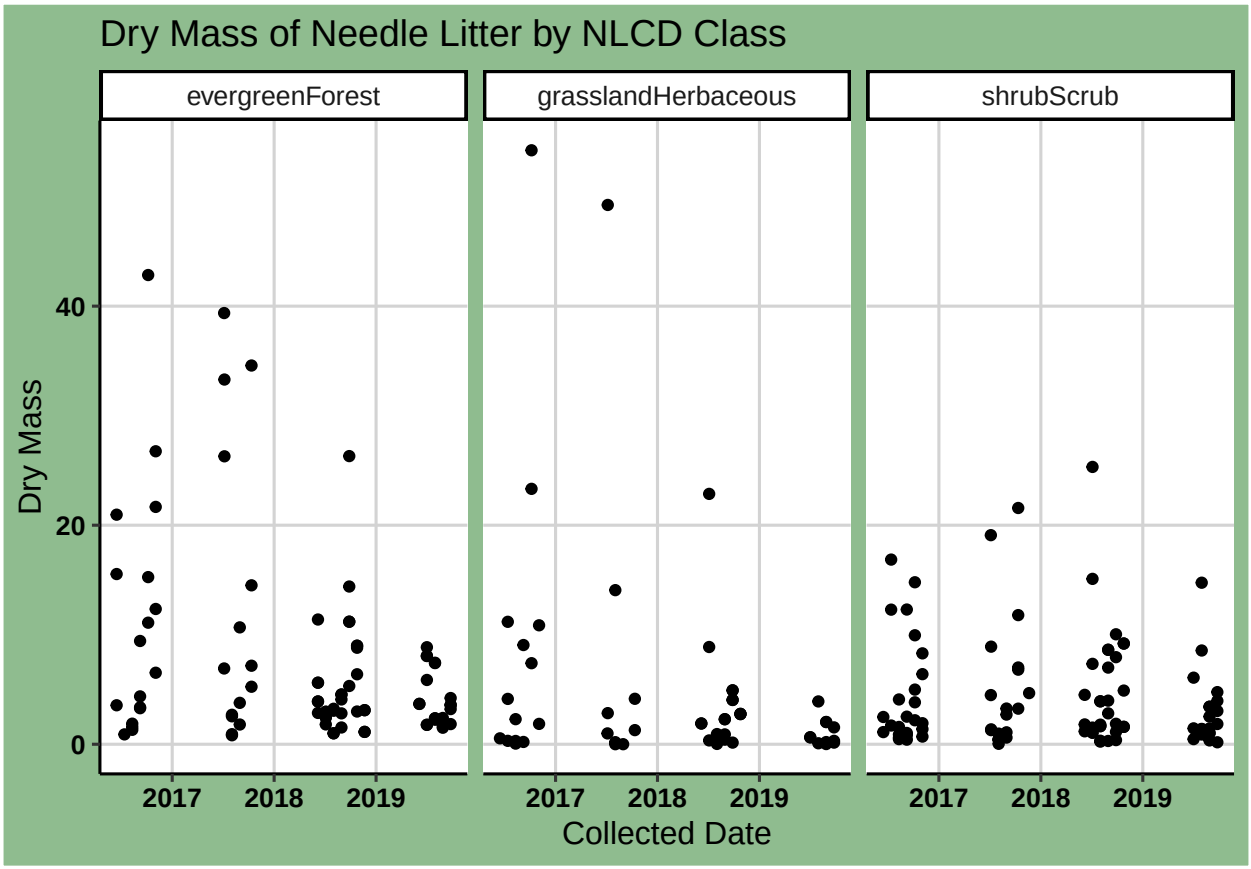
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Needles_litter_plot <- NiwotRide.Litter.Data %>%
  filter(functionalGroup == 'Needles') %>% #filtering Needles out of functional group
  ggplot(aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = 'Dry Mass of Needle Litter', x = 'Collected Date',
       y = 'Dry Mass', color = 'NLCD Class')
Needles_litter_plot
```




```
#7
Needles_litter_faceted <- NiwotRide.Litter.Data %>%
  filter(functionalGroup == 'Needles') %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  facet_wrap(vars(nlcdClass), ncol= 3) +
  geom_point() +
  labs (title = 'Dry Mass of Needle Litter by NLCD Class', x = 'Collected Date',
```

```
        y = 'Dry Mass')
Needles_litter_faceted
```



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

Answer: While I prefer plot 7 and think it is more effective, I think it depends in this situation. There are pros and cons to both. Plot 6 allows us to directly compare the dry mass levels of each NLCD class ontop of one another, but it is a little difficult to compare and make sense of the three colors for each year. Plot 7 spaces all three NLCD class into three graphs so you can compare side by side and it is easier to make sense of each class and their levels, but it is somewhat hard to compare the lower levels of dry mass for each class since they are clustered ontop one another.