Assignment 5: Data Visualization

Lexi Nelson

Spring 2026

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, 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
#set up session
#load tidyverse, here, & cowplot packages
library(tidyverse)
```

```
## -- 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
```

```
library(here)
```

```
## here() starts at /home/guest/ENV872/EDE_Spring2026/Assignments
```

```
library(cowplot)
```

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

```
#verify home directory:
#two commands below return the same path where I am currently working
getwd()
```

```
## [1] "/home/guest/ENV872/EDE_Spring2026/Assignments"
```

```
here()
```

```
## [1] "/home/guest/ENV872/EDE_Spring2026/Assignments"
```

```
#read in processed data files
#first I copied them into the Processed folder under the home directory
Nutrients <- read.csv(
  file = here("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
  stringsAsFactors = TRUE)
Litter <- read.csv(
  file = here("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv"),
  stringsAsFactors = TRUE)

#2
#make sure R is reading dates as date format
class(Nutrients$sampledate)#factor
```

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

```
# Format as date
Nutrients$sampledate <- ymd(Nutrients$sampledate)
class(Nutrients$sampledate) #now it is a date
```

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

```
class(Litter$collectDate)#factor
```

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

```r
# Format as date
Litter$collectDate <- ymd(Litter$collectDate)
class(Litter$collectDate) #now it is a date
```

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

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 a theme with at least 2 customized elements
#adding a plot title and gridlines to our example code from Lab 5
mytheme <- theme_classic(base_size = 14) +
  theme(
    axis.text = element_text(color = "black"),
    legend.position = "bottom",
    plot.title = element_text(size = 18),
    panel.grid.major = element_line(color = "gray80")
  )
#set as default
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()`).

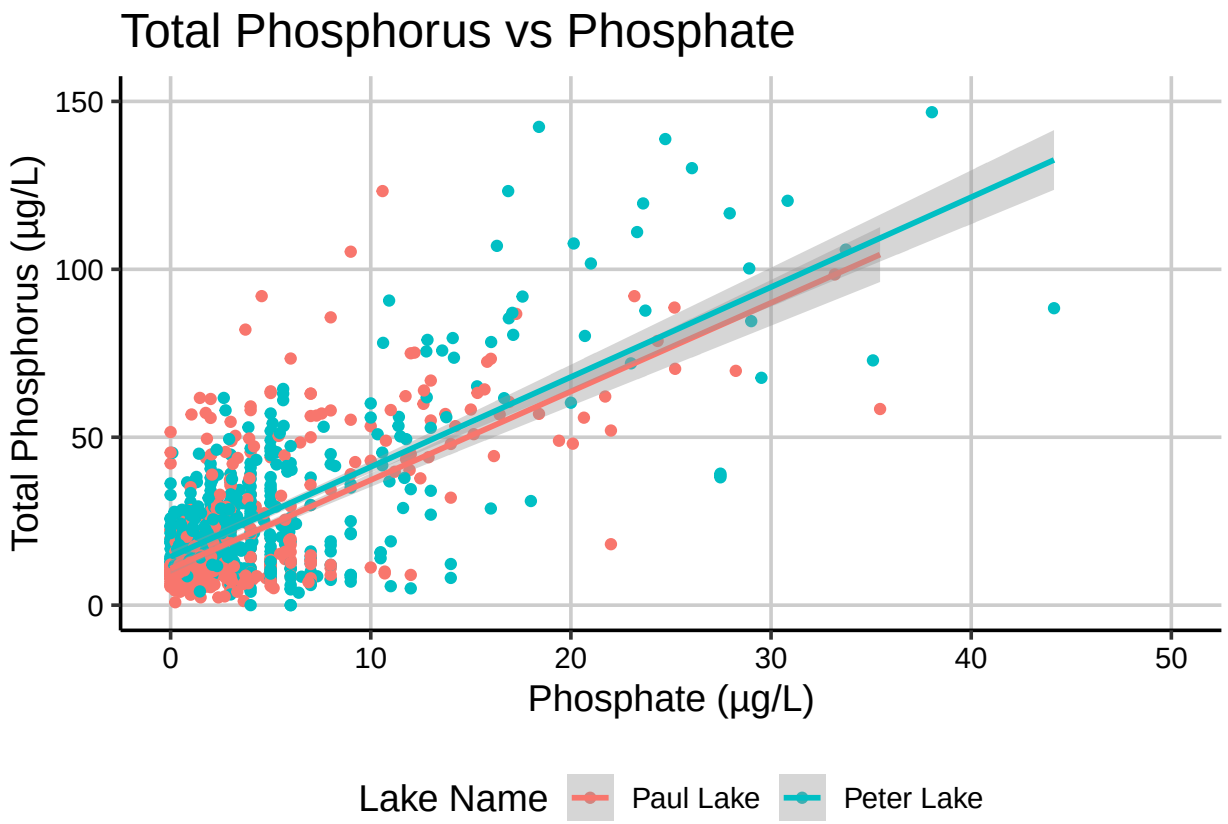
```r
#4
#plot total P vs PO4
PvsPO4 <-
  ggplot(Nutrients, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = lm) + #line of best fit
  xlim(0, 50) +
  ylim(0, 150) + #adjust axes to hide extreme values
  labs(
    title = "Total Phosphorus vs Phosphate",
    x = "Phosphate (µg/L)",
    y = "Total Phosphorus (µg/L)",
```

```
    color = "Lake Name"
  )
print(PvsPO4)
```

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

```
## Warning: Removed 21948 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21948 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. Show all months, even those where no data was collected.

Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis text in your theme to `element_blank()` removes the axis text (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 relative sizes when combined using `cowplot`.

```r
#I edited the line above to make the overall cowplot output more proportional

#5
#use factors to make sure months 1:12 show for all plots
Nutrients$month <- factor(Nutrients$month, levels = 1:12)
scale <- scale_x_discrete(drop = FALSE)

#create boxplot of temperature
Temp_boxplot <-
  ggplot(Nutrients, aes(x = month, y = temperature_C)) +
  geom_boxplot(aes(color = lakename)) +
  scale +
  labs(y = "Temperature (C)") +
  theme(legend.position = "none", #remove legend
  axis.title.x = element_blank(),
  axis.ticks.x = element_blank()) #remove axis text & legends
#print(Temp_boxplot) used to check boxplot

#create boxplot of TP
TP_boxplot <-
  ggplot(Nutrients, aes(x = month, y = tp_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale +
  labs(y = "Total Phosphorus (µg/L)") +
  theme(legend.position = "none",
  axis.title.x = element_blank(),
  axis.ticks.x = element_blank()) #remove axis text & legends
#print(TP_boxplot) used to check boxplot

#create boxplot of TN
#keep legend here since we want one total, shift its position from default
TN_boxplot <-
  ggplot(Nutrients, aes(x = month, y = tn_ug)) +
  geom_boxplot(aes(color = lakename)) +
  scale +
  labs(
   x = "Month",
  y = "Total Nitrogen (µg/L)",
  color = "Lake Name") +
  theme(legend.position = "bottom")
#print(TN_boxplot) used to check boxplot

#combine using cowplot
three_boxplots <- plot_grid(Temp_boxplot, TP_boxplot, TN_boxplot, ncol=1,
                            align = 'v', axis = "l",
                            rel_heights = c(1, 1, 1.75))
```

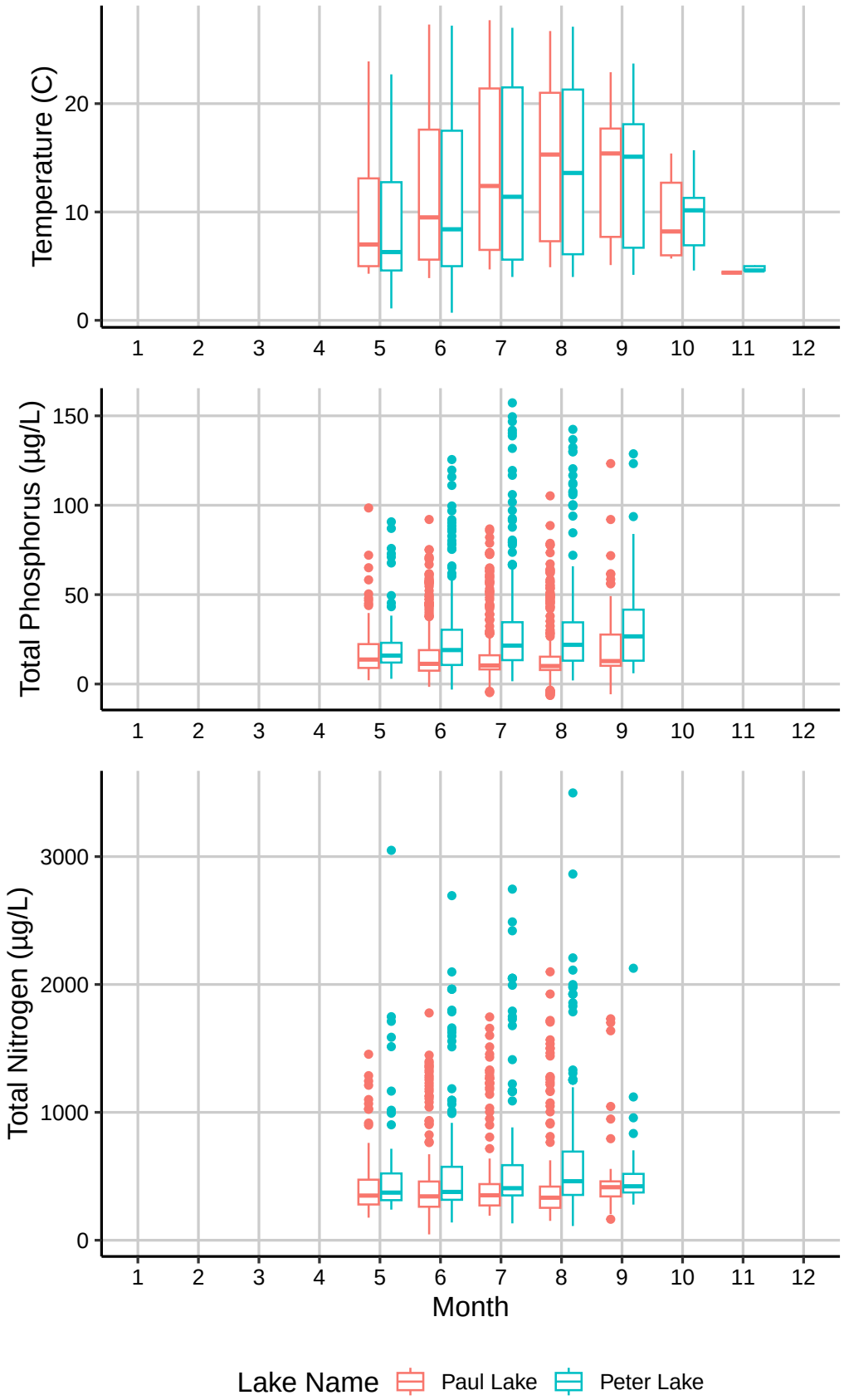
```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## 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()`).
```

```
#use different relative heights to compensate for legend
print(three_boxplots)
```


Temperature (C)
0
10
20
Total Phosphorus (µg/L)
0
50
100
150
Total Nitrogen (µg/L)
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month
Lake Name
Paul Lake
Peter Lake

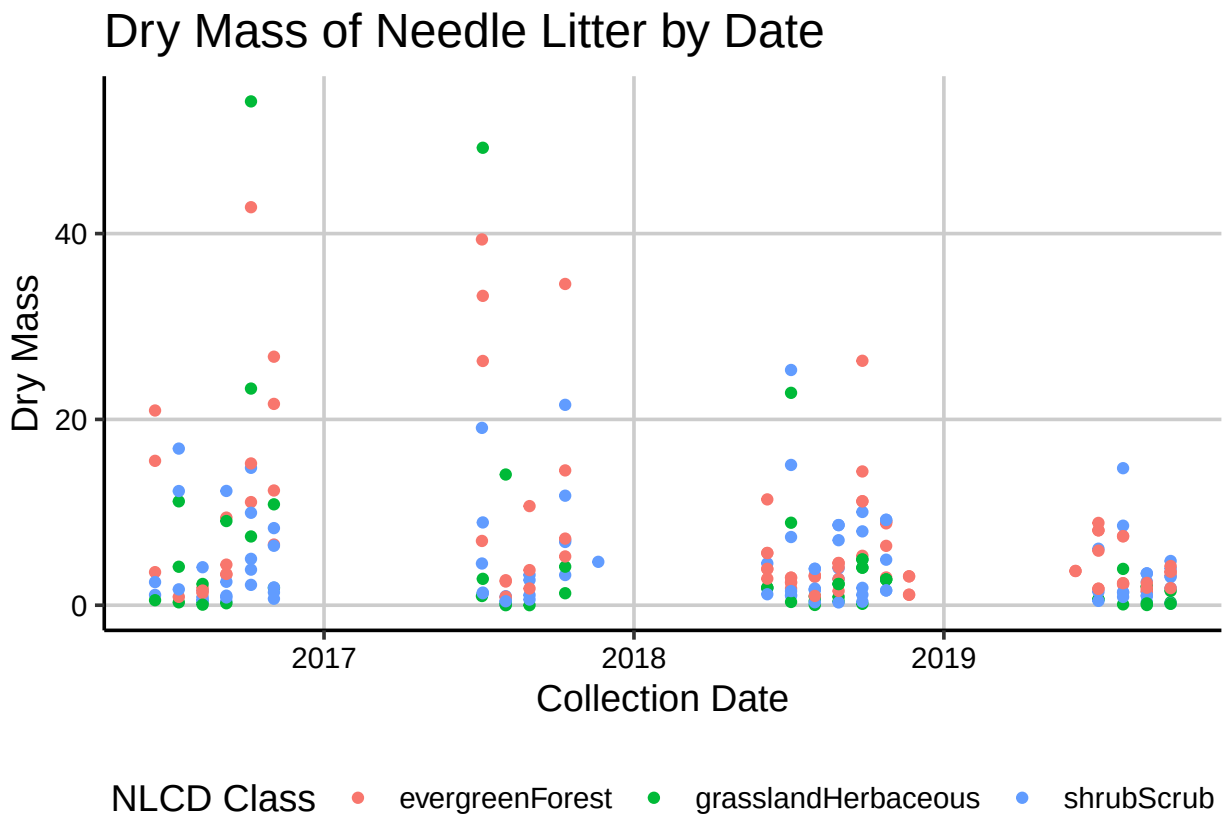
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: I observe that the temperatures and temperature variation is similar between the two lakes but Paul lake has a slightly higher median temperature than Peter lake during all months except October and November. The total phosphorus boxplots are more tightly clustered around the median, with both lakes having several high outlier values. between Paul lake and Peter lake the boxplots are all shifted up slightly, so though the distribution of values is similar, the total phosphorus values are a few degrees higher on average at Peter lake than Paul lake. I had the same observations for the total nitrogen boxplots as I did for the total phosphorus boxplots.

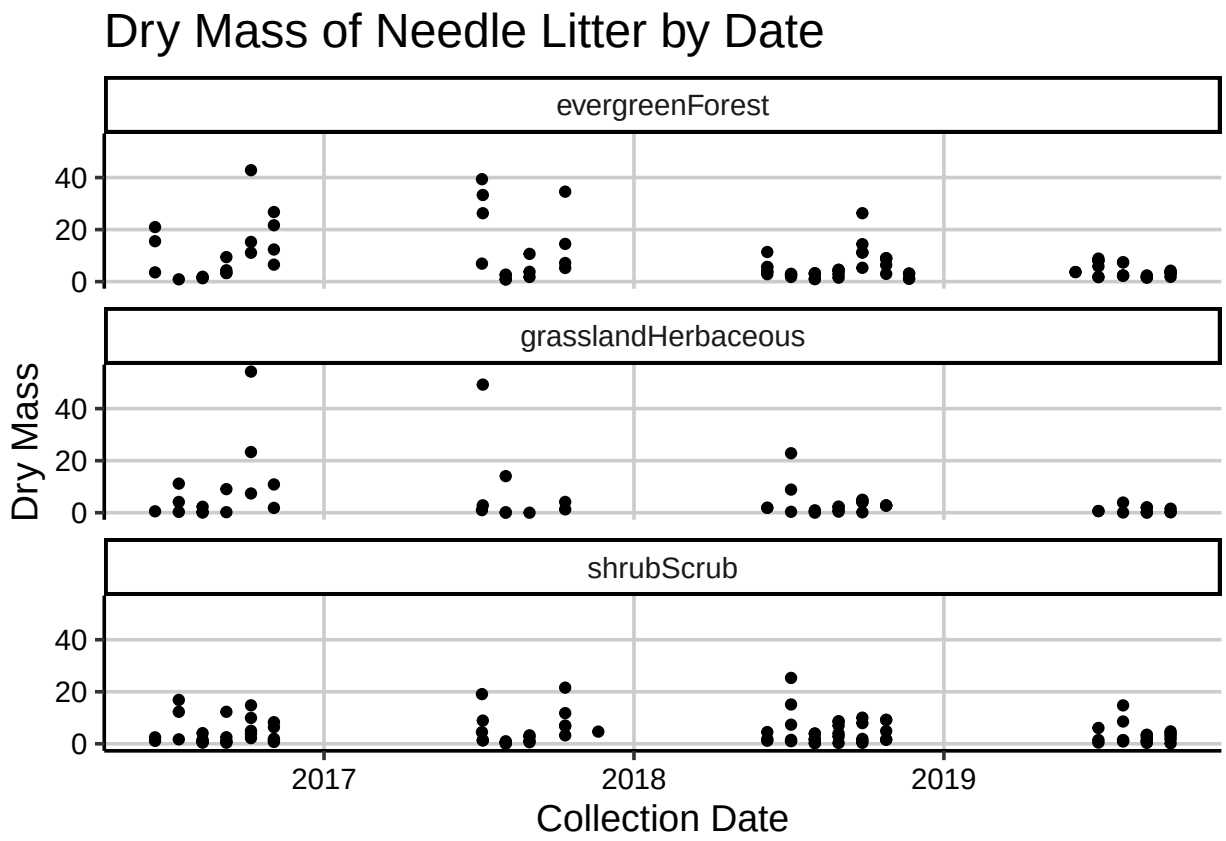
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
#plot dry mass of needle litter by date
litter_mass_plot <- Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass,
                      color = nlcdClass)) +
  geom_point() +
  labs(
    title = "Dry Mass of Needle Litter by Date",
    x = "Collection Date",
    y = "Dry Mass",
    color = "NLCD Class"
  )
print(litter_mass_plot)
```




```
#7
#same plot with NLCD classes separated into 3 facets
litter_mass_plot_faceted <- Litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass)) +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  geom_point() +
  labs(
    title = "Dry Mass of Needle Litter by Date",
    x = "Collection Date",
    y = "Dry Mass")
print(litter_mass_plot_faceted)
```



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

Answer:I think that plot #6 is more effective because it is easier to compare how close together the dry mass values for different NLCD classes are when they are on the same plot. With the current y scale, this plot makes it easy to see which NLCD class the high outlier points belong to.