Assignment 5: Data Visualization

Ayden Schirmacher

Spring 2025

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.

```
remove(list=ls())
#1 loading packages to library and check workspace
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 1.9.4     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
```

```r
library(lubridate)
library(here)
```

```
## here() starts at /home/guest/EDA_Spring 2025
```

```r
library(cowplot)
```

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

```r
library(ggthemes)
```

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

```r
getwd()
```

```
## [1] "/home/guest/EDA_Spring 2025"
```

```r
#set variable name for workspace
processed = "./Data/Processed_KEY"

#read in csv files
lakes_data<-read_csv(
  here(processed,"NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
```

```
## Rows: 23008 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): lakename
## dbl  (13): year4, daynum, month, depth, temperature_C, dissolvedOxygen, irra...
## date  (1): sampledate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
niwot_ridge<-read_csv(
  here(processed,"NEON_NIWO_Litter_mass_trap_Processed.csv"))
```

```
## Rows: 1692 Columns: 13
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr  (7): plotID, trapID, functionalGroup, qaDryMass, nlcdClass, plotType, g...
## dbl  (5): dryMass, subplotID, decimalLatitude, decimalLongitude, elevation
## date (1): collectDate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#2 check that dates are read as dates
class(lakes_data$sampledate)
```

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

```
class(niwot_ridge$collectDate)
```

```
## [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

```
#3
custom_theme <- theme_base() +
  theme(
    text = element_text(color='black', size=10, face = 'italic'),
    panel.grid.minor = element_line(color="gray87"),
    panel.grid.major = element_line(color="gray87"),
    plot.background = element_rect(color='black', fill='snow2'),
    axis.ticks = element_line(size=0.25))
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
theme_set(custom_theme)
```

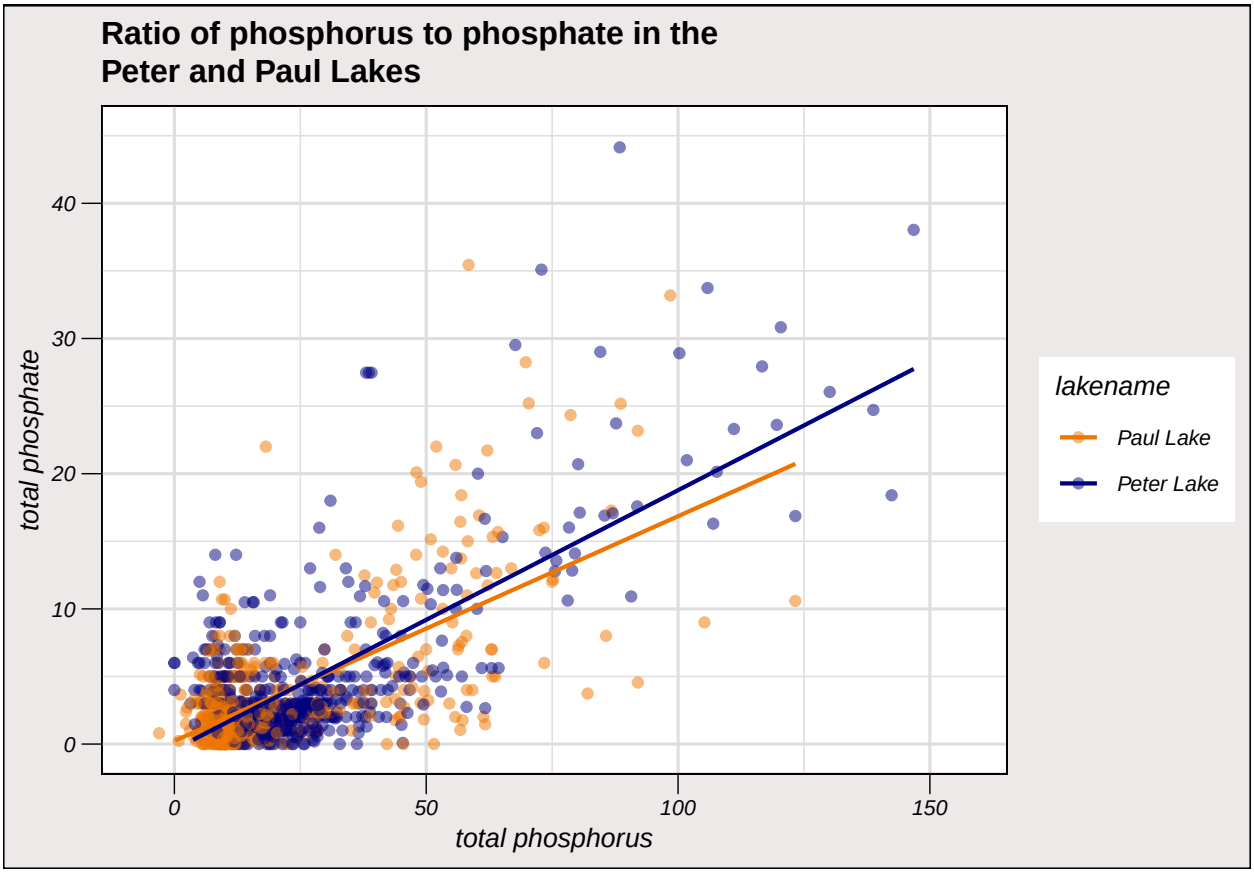
Create graphs

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
lakes_data%>%
ggplot(aes(x=tp_ug, y=po4, color=lakename))+
scale_color_manual(values = c("Peter Lake" = "navy",
                              "Paul Lake" = "darkorange2"))+
#^^ manual selecting of colors for each lake
  geom_point(alpha=0.5)+
  ylim(0,45)+
  geom_smooth(method = lm, se=FALSE, size=0.75)+
  labs(x="total phosphorus",
      y="total phosphate",
      title="Ratio of phosphorus to phosphate in the \nPeter and Paul Lakes")
```

```
## 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.

## ‘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()‘).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (‘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.

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`.

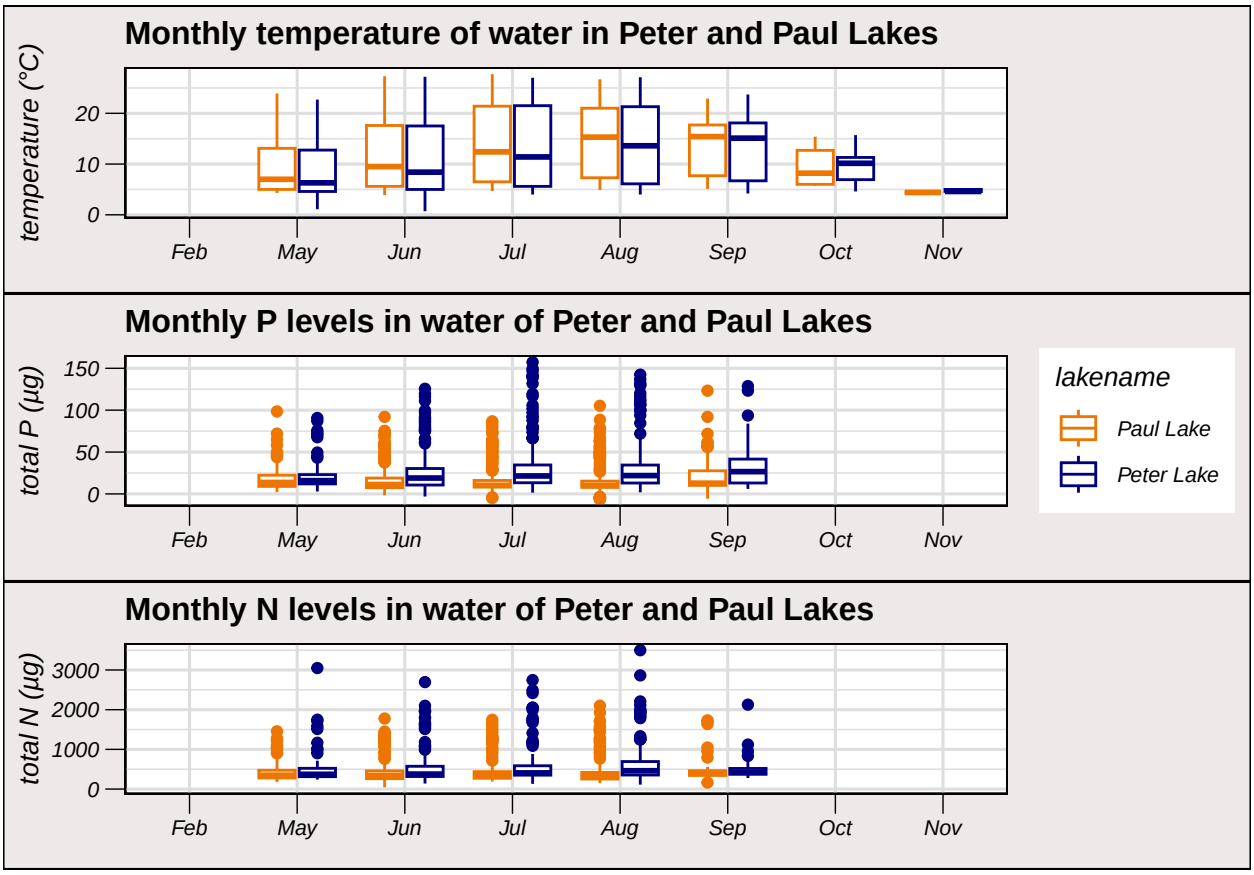
```
#5 creating three boxplots
temp_plot<-ggplot(lakes_data, aes(x=month, y=temperature_C, color=lakename))+
  #^^ selecting variables and coloring by lake name
scale_color_manual(values = c("Peter Lake" = "navy",
                              "Paul Lake" = "darkorange2"))+
  geom_boxplot(aes(x=factor(month, levels=1:12, labels=month.abb)))+
  labs(x=element_blank(), y="temperature (°C)", #eliminating x labels
       title="Monthly temperature of water in Peter and Paul Lakes")+
    theme(legend.position = "none") #eliminating legend

tp_plot<-ggplot(lakes_data, aes(x=month, y=tp_ug, color=lakename))+
scale_color_manual(values = c("Peter Lake" = "navy",
                              "Paul Lake" = "darkorange2"))+
  geom_boxplot(aes(x=factor(month, levels=1:12, labels=month.abb)))+
    #^^ making all months visible to appropriately visualize annual data
  labs(x=element_blank(), y="total P (µg)", #eliminating x labels
```

```
       title="Monthly P levels in water of Peter and Paul Lakes")

tn_plot<-ggplot(lakes_data, aes(x=month, y=tn_ug, color=lakename))+
scale_color_manual(values = c("Peter Lake" = "navy",
                              "Paul Lake" = "darkorange2"))+
  #^^ manual selecting of colors for each lake
  geom_boxplot(aes(x=factor(month, levels=1:12, labels=month.abb)))+
  labs(x=element_blank(), y="total N (µg)", #eliminating x labels
       title="Monthly N levels in water of Peter and Paul Lakes")+
    theme(legend.position = "none") #eliminating legend

plot_grid(temp_plot, tp_plot, tn_plot, ncol = 1, nrow = 3, align = 'v')
```

```
## 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()‘).
```



```
#^^cowplot for display
```

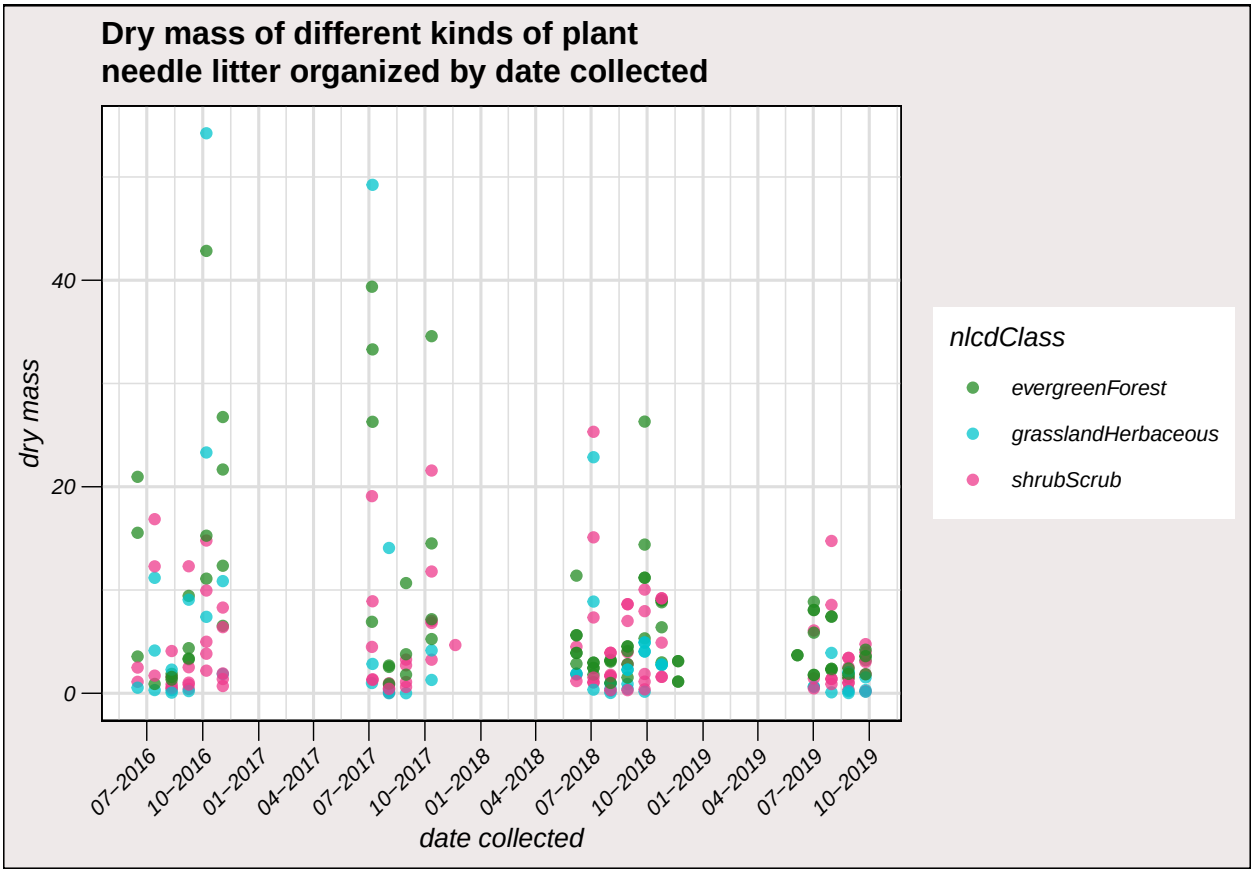
Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: The variables of interest all have a lack of data in the winter in both lakes. In each, dependent variable increases on average during the summer in each lake and in most months; while it is hard to compare 'amount' of temperature to total P and N, it is clear that there is a possible correlation between increased temperature and increased N and P. The total amount of N is on average a lot higher than total P. Peter Lake has greater amounts across the board. However, the P and N graphs stand apart for having a MUCH higher amount of outliers as compared to the temperature graph.

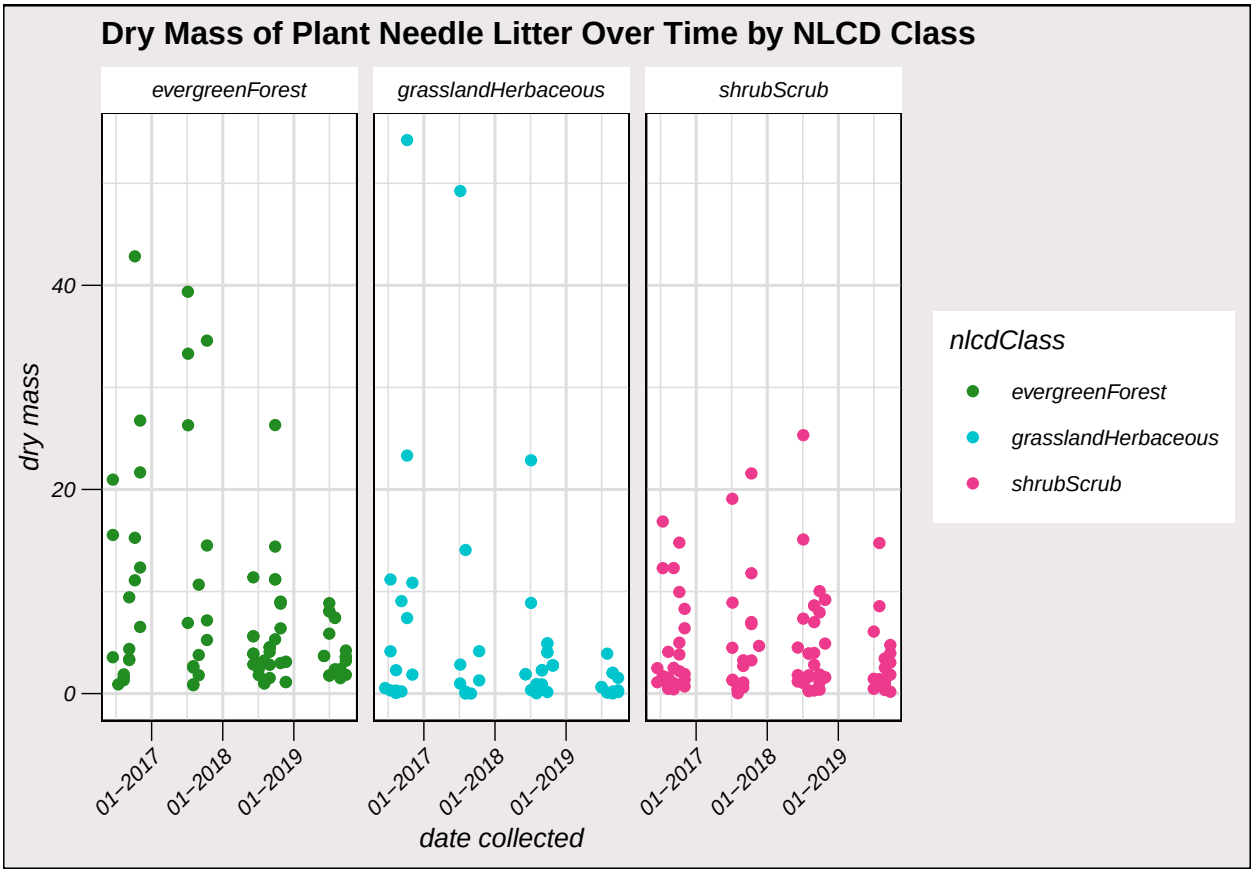
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
niwot_ridge%>%
  filter(functionalGroup == "Needles")%>%
  ggplot(aes(x=collectDate, y=dryMass, color=nlcdClass))+
  scale_color_manual(values=c("evergreenForest"="forestgreen",
                              "grasslandHerbaceous"="turquoise3",
                              "shrubScrub"="violetred2"))+
  #^^individual colors for each class
  geom_point(alpha=0.75)+ #transparency
  scale_x_date(date_breaks = "3 months", date_labels = "%m-%Y")+
  #^^adjusting x axis scale and date format
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  #^^adjust x axis labels
  labs(x="date collected", y="dry mass",
  title="Dry mass of different kinds of plant \nneedle litter organized by date collected")
```




```
#7
niwot_ridge %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(aes(x = collectDate, y = dryMass, color=nlcdClass)) +
  geom_point() +
  scale_x_date(date_labels = "%m-%Y") +
  #^^adjusting x axis scale and date format
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  #^^adjust x axis labels
  facet_wrap(vars(nlcdClass)) +  #facet by nlcd class
  scale_color_manual(values=c("evergreenForest"="forestgreen",
                              "grasslandHerbaceous"="turquoise3",
                              "shrubScrub"="violetred2"))+
  #^^individual colors for each class
  labs(x="date collected", y="dry mass",
       title = "Dry Mass of Plant Needle Litter Over Time by NLCD Class")
```



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 better shows the aligned seasonal gaps in the dry mass by time between all three of the categories. It also better shows the differences in the three classes. I think figure 7 is more difficult to immediately interpret, since it looks a little bit like a long continuous time scale.