Assignment 5: Data Visualization

Andi Mujollari

Fall 2023

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)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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(lubridate)
library(here)
```

```
## here() starts at /Users/andimujollari/Desktop/EDE-Fall2023
```

```
library(cowplot)
```

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

```
#install.packages('cowplot')
getwd()
```

```
## [1] "/Users/andimujollari/Desktop/EDE-Fall2023"
```

```
PP_Lake_Nutrients_data <- read.csv('./Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Pr
NN_Litter_data <- read.csv('./Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv')

#2
# Check the structure if dates are being read as date format
str(PP_Lake_Nutrients_data)
```

```
## 'data.frame':    23008 obs. of  15 variables:
##  $ lakename       : chr  "Paul Lake" "Paul Lake" "Paul Lake" "Paul Lake" ...
##  $ 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     : chr  "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-27" ...
##  $ 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 ...
```

```
str(NN_Litter_data)
```

```
## 'data.frame':    1692 obs. of  13 variables:
##  $ plotID          : chr  "NIWO_062" "NIWO_061" "NIWO_062" "NIWO_064" ...
##  $ trapID          : chr  "NIWO_062_050" "NIWO_061_169" "NIWO_062_050" "NIWO_064_103" ...
##  $ collectDate     : chr  "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-16" ...
##  $ functionalGroup : chr  "Seeds" "Other" "Woody material" "Seeds" ...
```

```
##  $ dryMass         : num  0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : chr  "N" "N" "N" "N" ...
##  $ 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       : chr  "shrubScrub" "evergreenForest" "shrubScrub" "evergreenForest" ...
##  $ plotType        : chr  "tower" "tower" "tower" "tower" ...
##  $ geodeticDatum   : chr  "WGS84" "WGS84" "WGS84" "WGS84" ...
```

```
#In both of the datasets the dates are not being reas as date format so we have to impose R to read the

PP_Lake_Nutrients_data$sampledate <- as.Date(PP_Lake_Nutrients_data$sampledate,
                                             format = "%Y-%m-%d")
NN_Litter_data$collectDate <- as.Date(NN_Litter_data$collectDate,
                                      format = "%Y-%m-%d")
```

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
library(ggplot2)
#mytheme <- theme_classic(base_size = 14) +
#  theme(axis.text = element_text(color = "black"),
#        legend.position = ("top"),
#Costumise background
#plot.background = element_rect(fill = "white"),

#Costumise axis title
#axis.title = element_text(color = "darkblue", size = 14, face = "bold"),

# Costumise axis ticks
#axis.text = element_text(color = "black", size = 12),
#axis.line = element_line(color = "green"),
#axis.ticks = element_line(color = "purple"),


#Costumise legend
#legend.title = element_text(color = "red", size = 12, face = "bold")

#theme_set(mytheme())
 #It was impossible to KNIT the file without commenting this section.
```

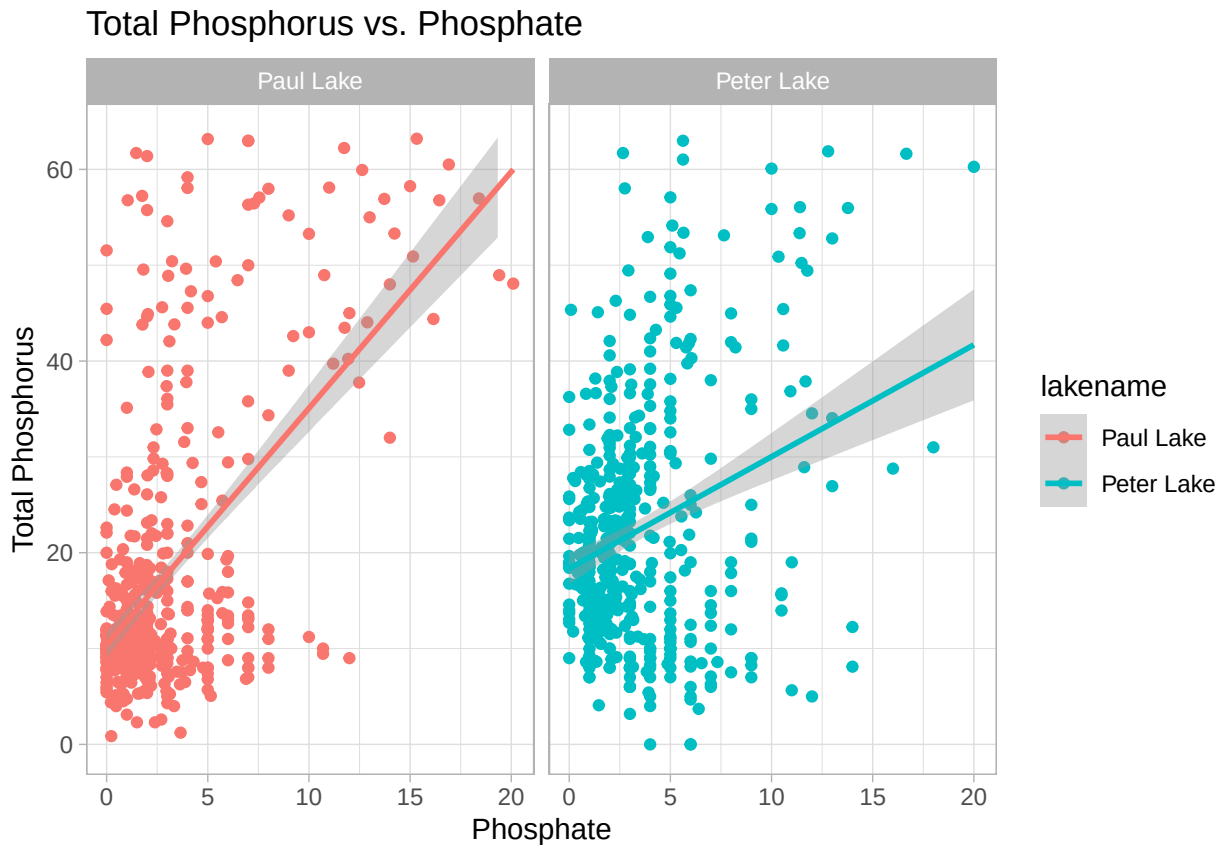
Create graphs

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 a line of best fit and color it black. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4

library(dplyr)

#Here i create the plots for each Lake

ggplot1 <- ggplot(PP_Lake_Nutrients_data,
       aes(x = po4,
           y = tp_ug,
           color = lakename)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(
    title = "Total Phosphorus vs. Phosphate",
    x = "Phosphate",
    y = "Total Phosphorus"
  ) +
  scale_y_continuous(limits = c(0, quantile(PP_Lake_Nutrients_data$tp_ug, 0.95, na.rm = TRUE))) +
  scale_x_continuous(limits = c(0, quantile(PP_Lake_Nutrients_data$po4, 0.95, na.rm = TRUE))) +
  theme_light () +
  facet_wrap(vars(lakename), ncol = 2)  # Separate plots for Peter and Paul)

print(ggplot1)
```

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

## Warning: Removed 22012 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 22012 rows containing missing values (`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.

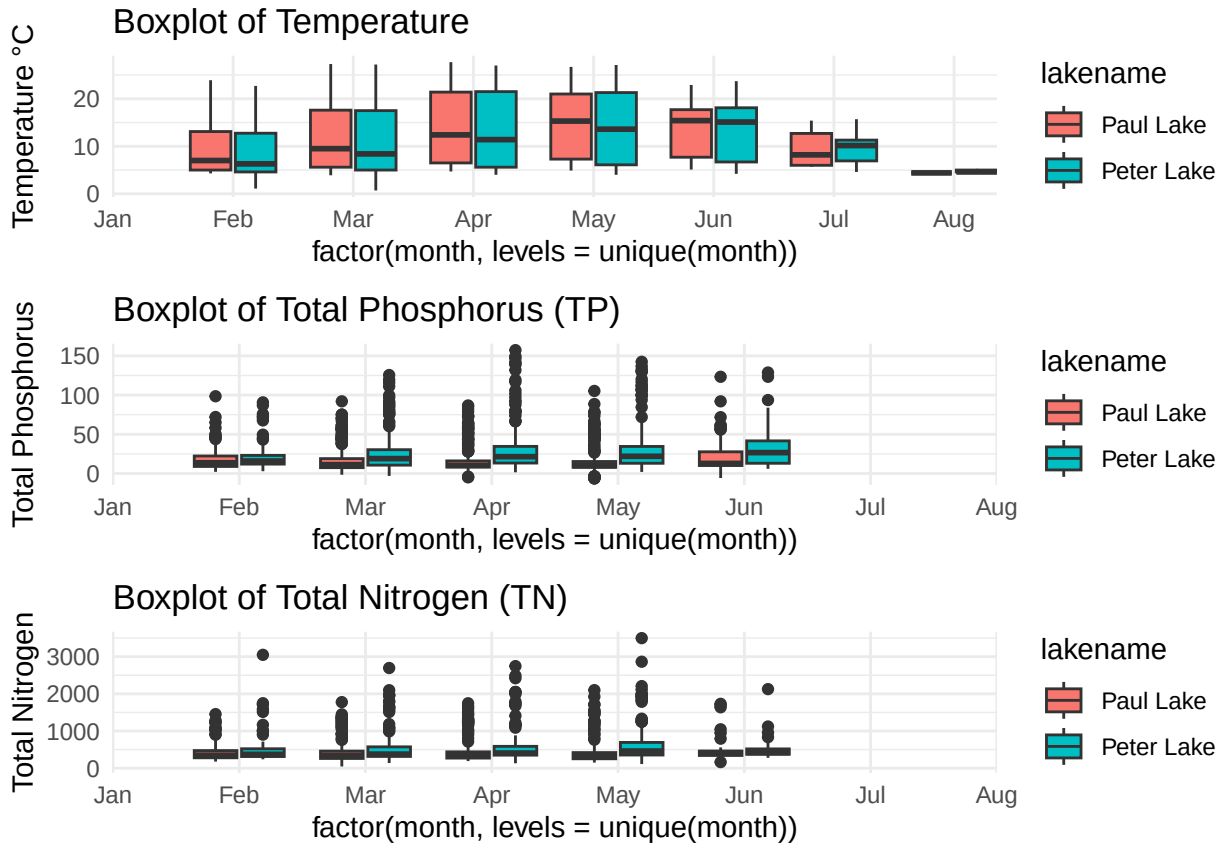
Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5



# Create a data frame with all months
all_months <- data.frame(month = unique(PP_Lake_Nutrients_data$month))

# Here i combine my original data with the data frame containing all months
PP_Lake_Nutrients_data <- merge(all_months, PP_Lake_Nutrients_data, by = "month", all.x = TRUE)

# Define a vector of abbreviated month labels
#month.abb

# Here i modify my boxplot code
boxplot_temp <- ggplot(PP_Lake_Nutrients_data, aes(x = factor(month, levels = unique(month)), y = tempe
  geom_boxplot() +
  labs(title = "Boxplot of Temperature", y = "Temperature °C") +
  theme_minimal() +
```

```
  scale_x_discrete(expand = c(0, 0), labels = month.abb, drop = FALSE)

boxplot_tp <- ggplot(PP_Lake_Nutrients_data, aes(x = factor(month, levels = unique(month)), y = tp_ug, f
  geom_boxplot() +
  labs(title = "Boxplot of Total Phosphorus (TP)", y = "Total Phosphorus") +
  theme_minimal() +
  scale_x_discrete(expand = c(0, 0), labels = month.abb, drop = FALSE)

boxplot_tn <- ggplot(PP_Lake_Nutrients_data, aes(x = factor(month, levels = unique(month)), y = tn_ug, f
  geom_boxplot() +
  labs(title = "Boxplot of Total Nitrogen (TN)", y = "Total Nitrogen") +
  theme_minimal() +
  scale_x_discrete(expand = c(0, 0), labels = month.abb, drop = FALSE)


# Combine the three boxplots into a single cowplot with one legend and aligned axes
combined_plot <- plot_grid(boxplot_temp, boxplot_tp,
                           boxplot_tn, ncol = 1, align = "v",
                           rel_heights = c(1, 1, 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(combined_plot)
```



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

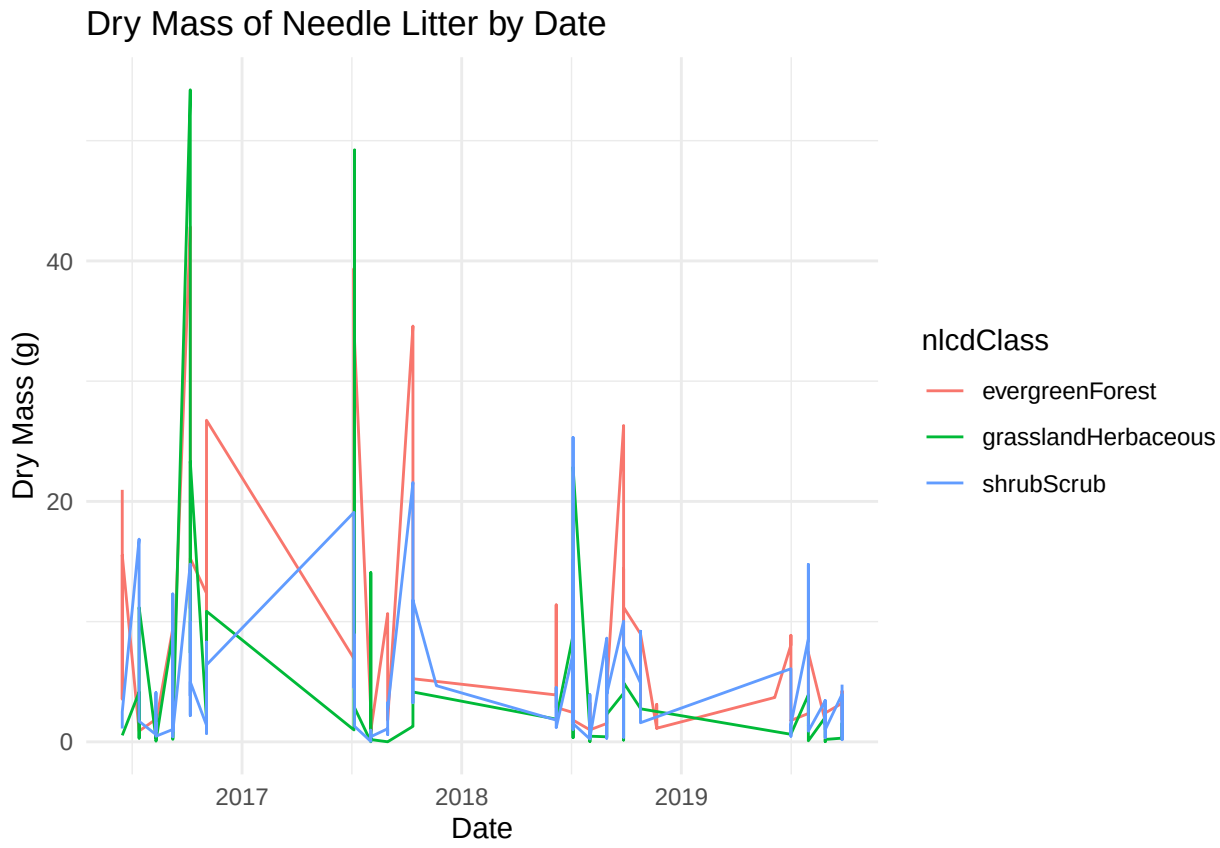
> Answer: The temperature is higher during the srping season for both of the lakes, compare to the other seasons. From the graph we can say that during February till June, lake Paul has a higher temperature than Peter. When it comes to Phosphorus indicator, lake Peter has a higher level than Paul almost the entire time. Finally, at the Total Nitrogen boxplot, we can observe that Peter lake has higher levels than Paul lake. Both of the indicators, Phosphorus and Nitrogen seem to have a possitive corrolation with temperature, because during the spring season when the temp is high, the level of the idicators are high too.

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)

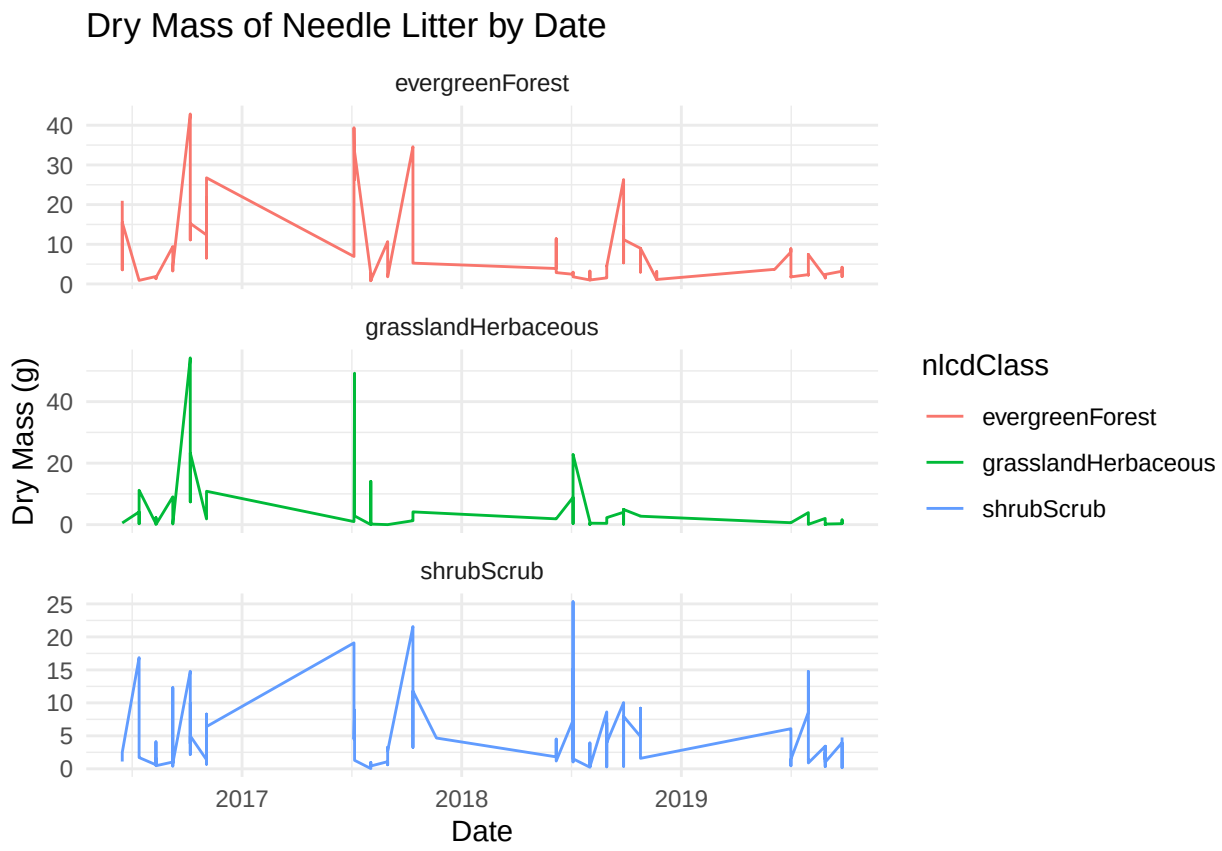
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

# Here i filter the dataset to include only the "Needles" group
needles_subset <- NN_Litter_data %>%
  filter(functionalGroup == "Needles")

#Now i create the plot
ggplot(needles_subset, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_line() +
```

```
  labs(
    title = "Dry Mass of Needle Litter by Date",
    x = "Date",
    y = "Dry Mass (g)"
  ) +
  theme_minimal()
```




```
#7

# Filter the dataset to include only the "Needles" functional group
needles_subset <- NN_Litter_data %>%
  filter(functionalGroup == "Needles")

# Here I create the plot with facets
ggplot(needles_subset, aes(x = collectDate, y = dryMass)) +
  geom_line(aes(color = nlcdClass)) +
  labs(
    title = "Dry Mass of Needle Litter by Date",
    x = "Date",
    y = "Dry Mass (g)"
  ) +
  facet_wrap(vars(nlcdClass), scales = "free_y", ncol = 1) +  # Separate by nlcdClass into facets
  theme_minimal()
```




```
#Finally i get the unique NLCD classes
unique_nlcd_classes <- unique(NN_Litter_data$nlcdClass)
```

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

> Answer: If we have to compare the levels betwen each other during the time, the plot 6 is using the same scale which might help us to compare. On the other hand, it is more appealing when we see the plot 7.