Assignment 5: Data Visualization

Chrissie Pantoja

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).

```
# Load necessary packages
library(tidyverse)
library(lubridate)
library(here)
library(cowplot)

# Verify your home directory
home_directory <- here::here()
print(home_directory)
```

```
## [1] "/Users/chrissiepantoja/Library/CloudStorage/OneDrive-DukeUniversity/PHD DUKE/1 COURSES/3 FALL SE
```

```r
# Define paths to the processed data files
peter_paul_file <- here("Data/Processed_KEY", "NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv
niwo_litter_file <- here("Data/Processed_KEY", "NEON_NIWO_Litter_mass_trap_Processed.csv")

# Read in the NTL-LTER data for Peter and Paul Lakes
peter_paul_data <- read_csv(peter_paul_file)

# Read in the Niwot Ridge litter dataset
niwo_litter_data <- read_csv(niwo_litter_file)

# Display the first few rows of each dataset to verify successful loading
head(peter_paul_data)
```

```
## # A tibble: 6 x 15
##   lakename  year4 daynum month sampledate depth temperature_C dissolvedOxygen
##   <chr>     <dbl>  <dbl> <dbl> <date>     <dbl>         <dbl>           <dbl>
## 1 Paul Lake  1984    148     5 1984-05-27  0             14.5             9.5
## 2 Paul Lake  1984    148     5 1984-05-27  0.25          NA              NA
## 3 Paul Lake  1984    148     5 1984-05-27  0.5           NA              NA
## 4 Paul Lake  1984    148     5 1984-05-27  0.75          NA              NA
## 5 Paul Lake  1984    148     5 1984-05-27  1             14.5             8.8
## 6 Paul Lake  1984    148     5 1984-05-27  1.5           NA              NA
## # i 7 more variables: irradianceWater <dbl>, irradianceDeck <dbl>, tn_ug <dbl>,
## #   tp_ug <dbl>, nh34 <dbl>, no23 <dbl>, po4 <dbl>
```

```r
head(niwo_litter_data)
```

```
## # A tibble: 6 x 13
##   plotID   trapID       collectDate functionalGroup dryMass qaDryMass subplotID
##   <chr>    <chr>        <date>      <chr>             <dbl> <chr>         <dbl>
## 1 NIWO_062 NIWO_062_050 2016-06-16  Seeds              0    N                31
## 2 NIWO_061 NIWO_061_169 2016-06-16  Other              0.27 N                41
## 3 NIWO_062 NIWO_062_050 2016-06-16  Woody material     0.12 N                31
## 4 NIWO_064 NIWO_064_103 2016-06-16  Seeds              0    N                32
## 5 NIWO_058 NIWO_058_101 2016-06-16  Needles            1.11 Y                32
## 6 NIWO_058 NIWO_058_101 2016-06-16  Leaves             0    N                32
## # i 6 more variables: decimalLatitude <dbl>, decimalLongitude <dbl>,
## #   elevation <dbl>, nlcdClass <chr>, plotType <chr>, geodeticDatum <chr>
```

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

```r
# Check the structure of the data to see the format of the date columns
str(peter_paul_data$sampledate)
```

```
##  Date[1:23008], format: "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-27" "1984-05-27" ...
```

```r
str(niwo_litter_data$collectDate)
```

```
##  Date[1:1692], format: "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-16" "2016-06-16" ...
```

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

```
# Custom theme
my_custom_theme <- theme(

  # Customize plot background
  plot.background = element_rect(fill = "lightgray", color = "black"),  # light gray background with bl

   # Customize plot title
  plot.title = element_text(face = "bold", size = 14, hjust = 0.5, color = "darkblue"),  # centered, bo

  # Customize axis labels
  axis.title.x = element_text(face = "italic", size = 12, color = "darkblue"),  # italic and colored x-
  axis.title.y = element_text(face = "italic", size = 12, color = "darkblue"),  # italic and colored y-

  # Customize axis ticks/gridlines
  axis.ticks = element_line(color = "blue"),  # blue axis ticks
  panel.grid.major = element_line(color = "gray80", size = 0.5),  # light gray major gridlines
  panel.grid.minor = element_blank(),  # remove minor gridlines
)
```

```
## 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.
```

```
# Set the custom theme as the default
theme_set(my_custom_theme)
```

Create graphs

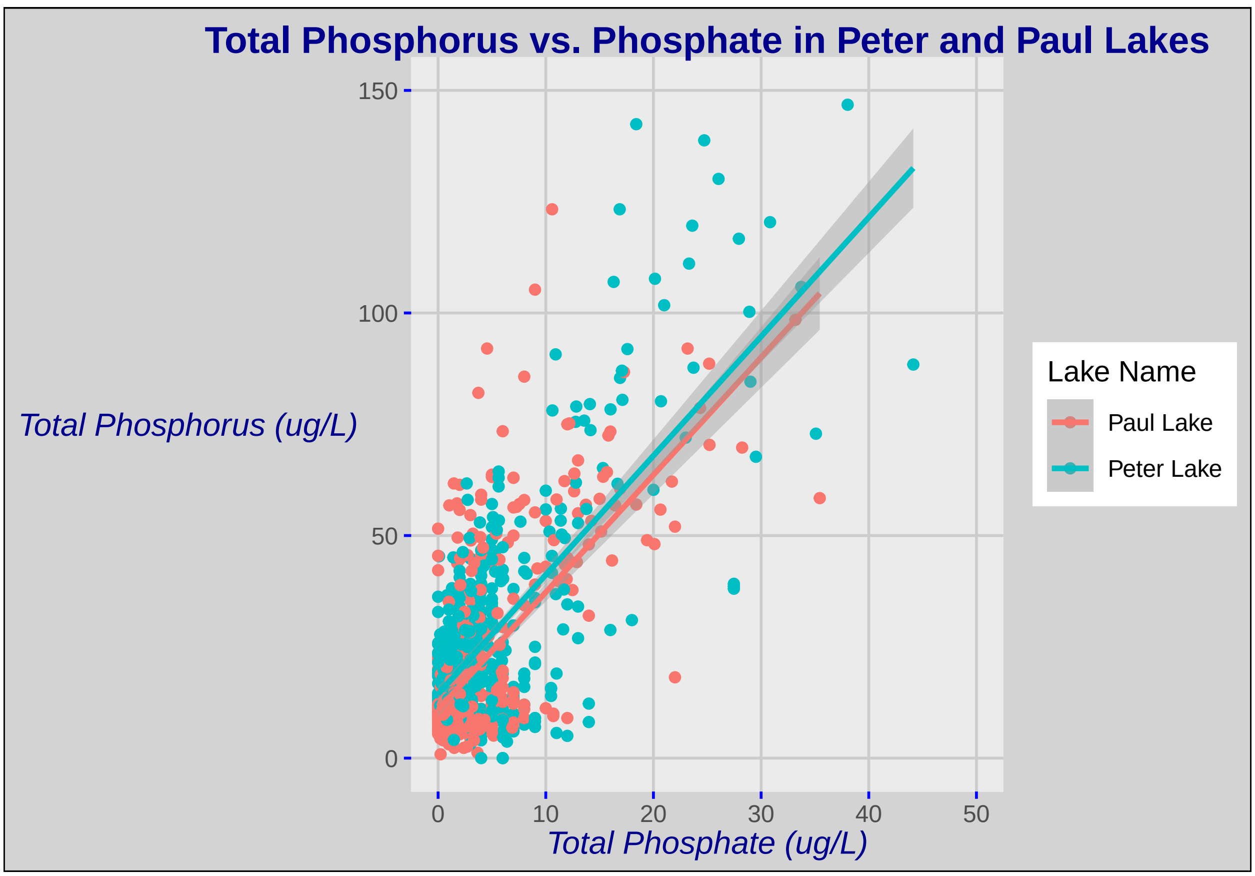
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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()`).

```
# Example plot to demonstrate the theme

example_plot <- ggplot(peter_paul_data, aes(x = po4,
           y = tp_ug,
```

```
            color = lakename)) +
  geom_point()+
  geom_smooth(method = "lm") +  # Add a line of best fit using the linear model (lm) method
  labs(
    title = "Total Phosphorus vs. Phosphate in Peter and Paul Lakes",
    x = "Total Phosphate (ug/L)",
    y = "Total Phosphorus (ug/L)",
    color = "Lake Name"
  ) +
  # Adjust axes to hide extreme values (use xlim() and ylim() based on data distribution)
  xlim(0, 50) +  # Set the limits for the x-axis based on reasonable phosphate values
  ylim(0, 150)   # Set the limits for the y-axis based on reasonable total phosphorus values

example_plot
```

```
## `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.

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`.

```
#Ensure 'month' is treated as a factor
peter_paul_data$month <- factor(peter_paul_data$month,
                                levels = 1:12,
                                labels = month.abb)

# (a) Boxplot for Temperature by Month and Lake
plot_temp <- ggplot(peter_paul_data, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  labs(title = "Temperature by Month", y = "Temperature (°C)", x = "Month") +
  theme(
    legend.position = "top",  # Remove legend for this plot
    legend.background = element_blank(),  # Remove legend background
    legend.text = element_text(size = 8),  # Smaller legend text size
    legend.title = element_text(size = 8),  # Smaller legend title size
    axis.title.x = element_blank(),  # Remove x-axis title for alignment
    axis.title.y = element_text(angle = 90, size = 8, hjust = 0.5)
  )+
  scale_color_discrete(name = "Lake Name") +  # Set legend title to "Lake Name"
  ylim(0,30)

# (b) Boxplot for Total Phosphorus (TP) by Month and Lake
plot_tp <- ggplot(peter_paul_data, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  labs(title = "Total Phosphorus by Month", y = "TP (µg/L)", x = "Month") +
  theme(
    legend.position = "none",  # Remove legend for this plot
    axis.title.x = element_blank(),  # Remove x-axis title for alignment
    axis.title.y = element_text(angle = 90, size = 8, hjust = 0.5)
  )+
  ylim(0,160)

# (c) Boxplot for Total Nitrogen (TN) by Month and Lake
plot_tn <- ggplot(peter_paul_data, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  labs(title = "Total Nitrogen by Month", y = "TN (µg/L)", x = "Month") +
  theme(
    legend.position = "none",  # Remove legend for this plot
    axis.title.x = element_blank(),  # Remove x-axis title for alignment
    axis.title.y = element_text(angle = 90, size = 8, hjust = 0.5)
  )+
  ylim(0,3500)
```

```
# Combine the plots into one grid while aligning axes
combined_plot <- plot_grid(
  plot_temp,
  plot_tp,
  plot_tn,
  nrow = 3,                       # Arrange plots in 3 rows
  align = "h"                    # Align plots horizontally
```

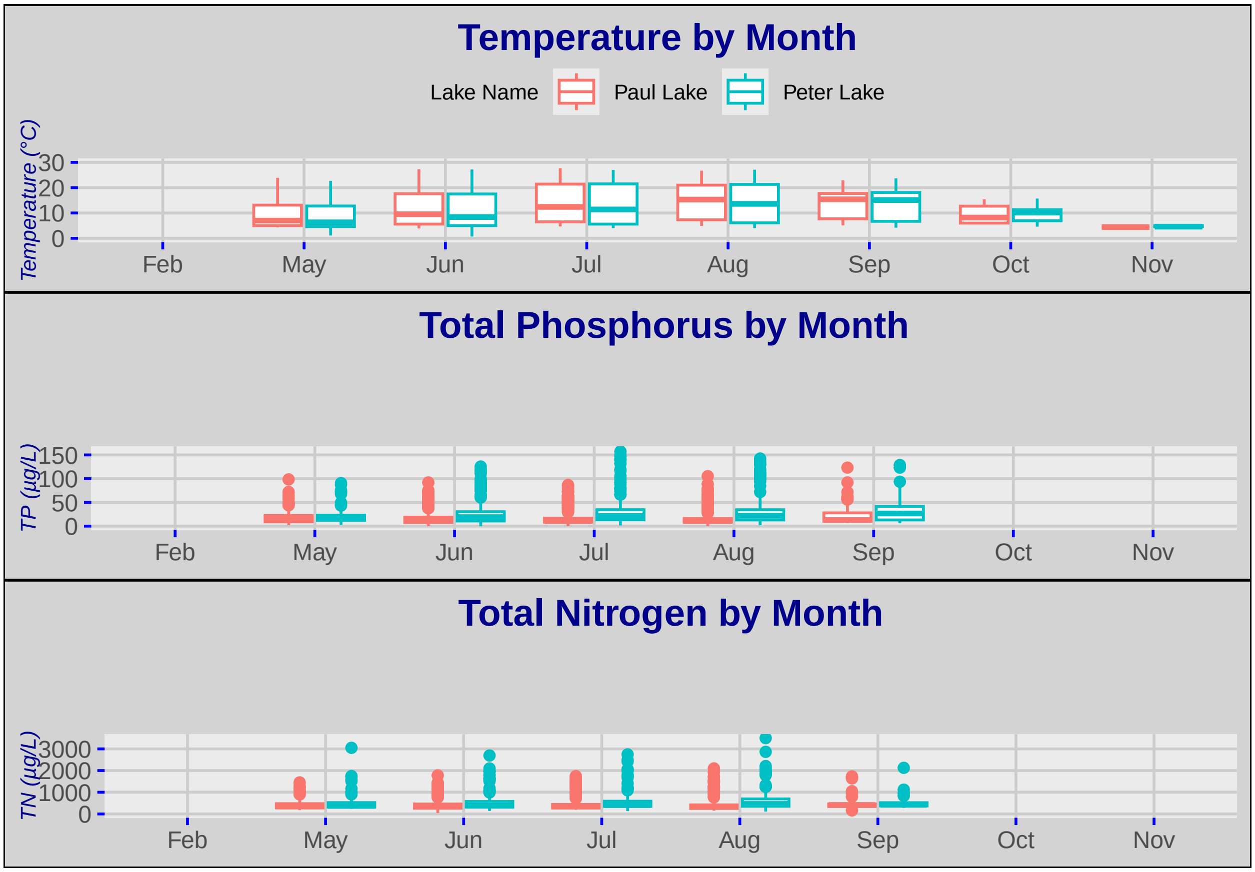
```
)
```

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

## Warning: Removed 20766 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

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

```
# Add a shared legend from one of the plots (e.g., from plot_temp)
legend <- get_legend(ggplot(peter_paul_data, aes(x = month, color = lakename)) +
                        geom_boxplot() +
                        theme(legend.position = "bottom") +
                        labs(title = "Lake Name") +
                        theme_minimal() +
                        theme(
                           axis.title.x = element_blank(),  # Remove x-axis title for alignment
                        )+ labs(color = "Lake Name"))
```

```
## Warning in get_plot_component(plot, "guide-box"): Multiple components found;
## returning the first one. To return all, use ‘return_all = TRUE‘.
```

```
# Combine the plots and the legend into the final layout
final_plot <- plot_grid(combined_plot, align='h', rel_heights = c(1.3, 1,1)) # Adjust legend size if ne

# Display the final combined plot
print(final_plot)
```

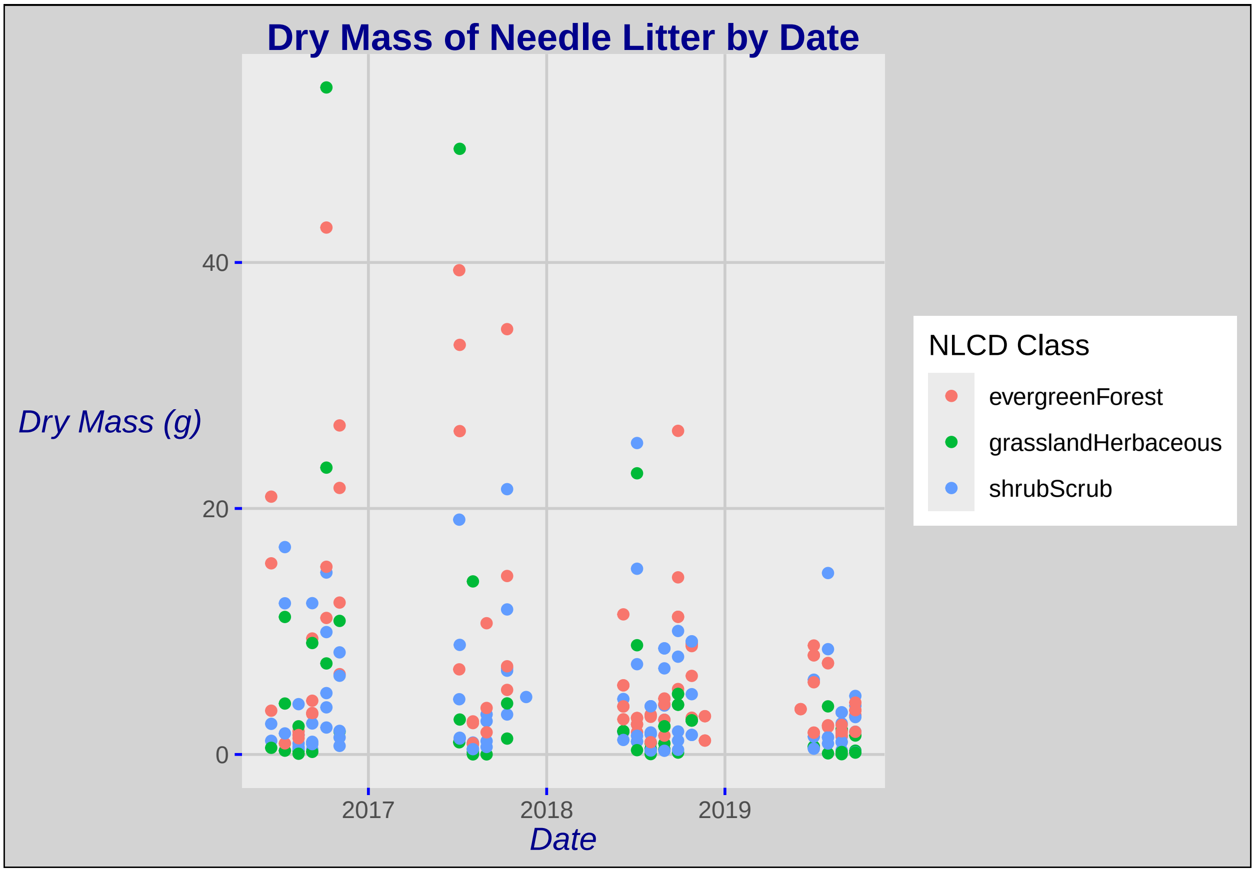


Question: What do you observe about the variables of interest over seasons and between lakes?

> Answer: Temperature: Peaks in summer (July/August) with higher variability. Paul Lake generally shows slightly higher temperatures than Peter Lake. Total Phosphorus & Nitrogen: Consistent medians, more variability in warmer months. Paul Lake tends to have higher Total Phosphorus levels, while Peter Lake shows higher Total Nitrogen levels during certain months. Lakes Comparison: Similar seasonal trends, but Paul Lake has higher temperatures and Total Phosphorus, whereas Peter Lake has higher Total Nitrogen in some months.

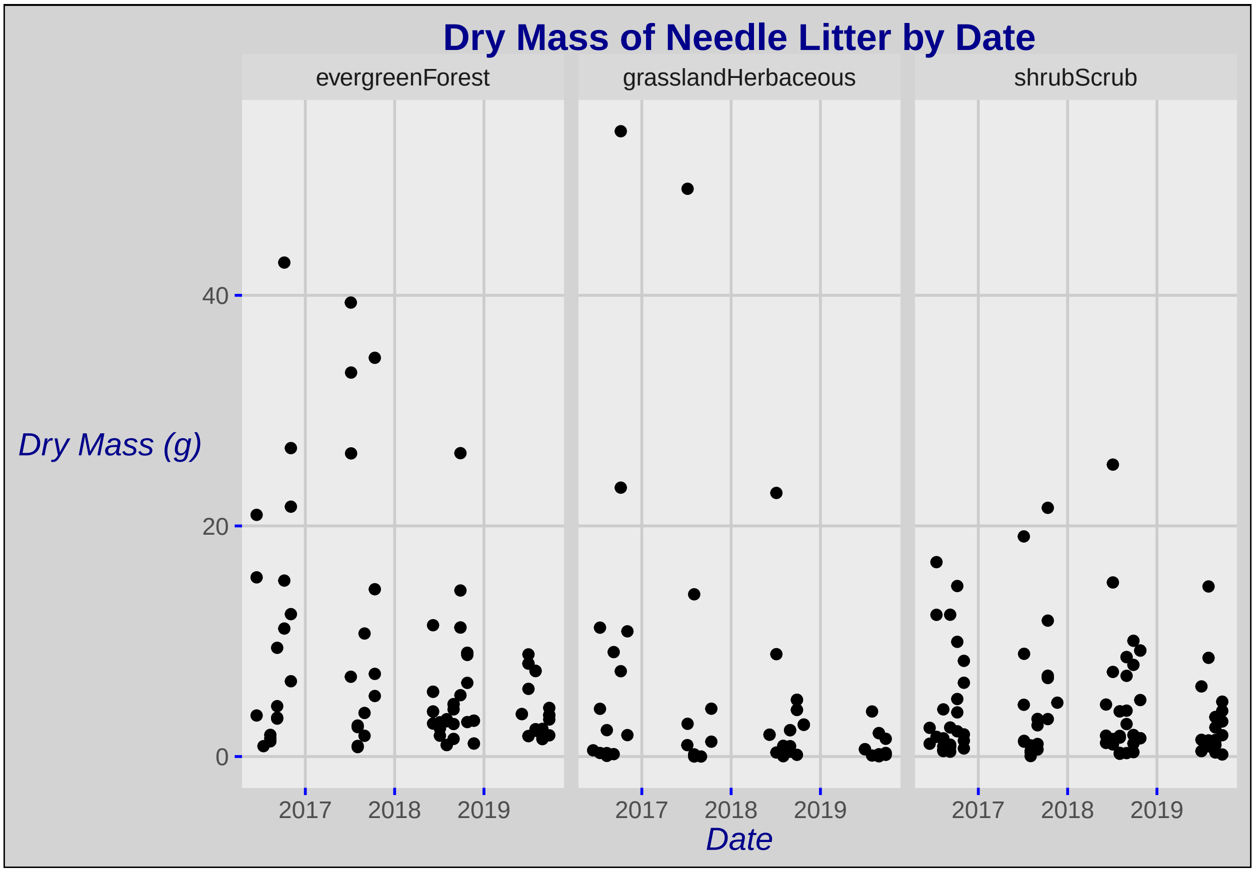
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

```r
# Filter the dataset for the "Needles" functional group
needles_data <- niwo_litter_data %>% filter(functionalGroup == "Needles")

# Plot the data
ggplot(needles_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
  labs(title = "Dry Mass of Needle Litter by Date",
       x = "Date",
       y = "Dry Mass (g)",
       color = "NLCD Class")
```



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

```
# Plot the data with facets

ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  facet_wrap(~nlcdClass)+
  labs(title = "Dry Mass of Needle Litter by Date",
       x = "Date",
       y = "Dry Mass (g)")
```



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

Answer: I would lean towards Plot 6 for its potential to clearly show seasonal dry mass trends and differences between NLCD classes, which can be very insightful for environmental analysis.