BrookHemphill_A05_DataVisualization.Rmd

Brook Hemphill

Spring 2024

List of Figures

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.

```
install.packages('formatR')
```

```
## Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=50), tidy=TRUE)

#1
getwd()
```

```
## [1] "/home/guest/EDA_Spring2024_New"
```

```r
library(cowplot); library(tidyverse); library(lubridate); library(here)
```

```
## -- 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 1.9.2     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()
## x lubridate::stamp() masks cowplot::stamp()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## here() starts at /home/guest/EDA_Spring2024_New
```

```r
here()
```

```
## [1] "/home/guest/EDA_Spring2024_New"
```

```r
PeterPaul.chem.nutrients_processed <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = T)
Niwot_Ridge_Litter_processed <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = T)

#2

PeterPaul.chem.nutrients_processed$sampledate <- as.Date(PeterPaul.chem.nutrients_processed$sampledate)
#PeterPaul.chem.nutrients_processed$month <- as.Date(PeterPaul.chem.nutrients_processed$month)
Niwot_Ridge_Litter_processed$collectDate <- as.Date(Niwot_Ridge_Litter_processed$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

```r
# 3

mytheme <- theme_minimal(base_size = 14) + theme(axis.text = element_text(color = "black",
    size = 10), legend.position = "top", legend.text = element_text(size = 12),
    legend.title = element_text(size = 12), plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 10))

mytheme2 <- theme_minimal(base_size = 14) + theme(axis.text = element_text(color = "black",
    size = 10), legend.position = "top", plot.title = element_text(hjust = 0.5),
    axis.title.x = element_blank())
```

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

PvPO4 <- ggplot(PeterPaul.chem.nutrients_processed,
    aes(x = po4, y = tp_ug, color = lakename, position = "dodge",
        shape = lakename)) + geom_point(alpha = 0.8) +
    xlim(0, 35) + geom_smooth(method = "lm", se = FALSE,
    size = 1) + scale_color_manual(values = c("darkblue",
    "lightblue")) + labs(title = "Total Phosphorus vs Phosphate\n",
    y = "Total Phosphorus (ug)", x = "Phospate") +
    mytheme
```

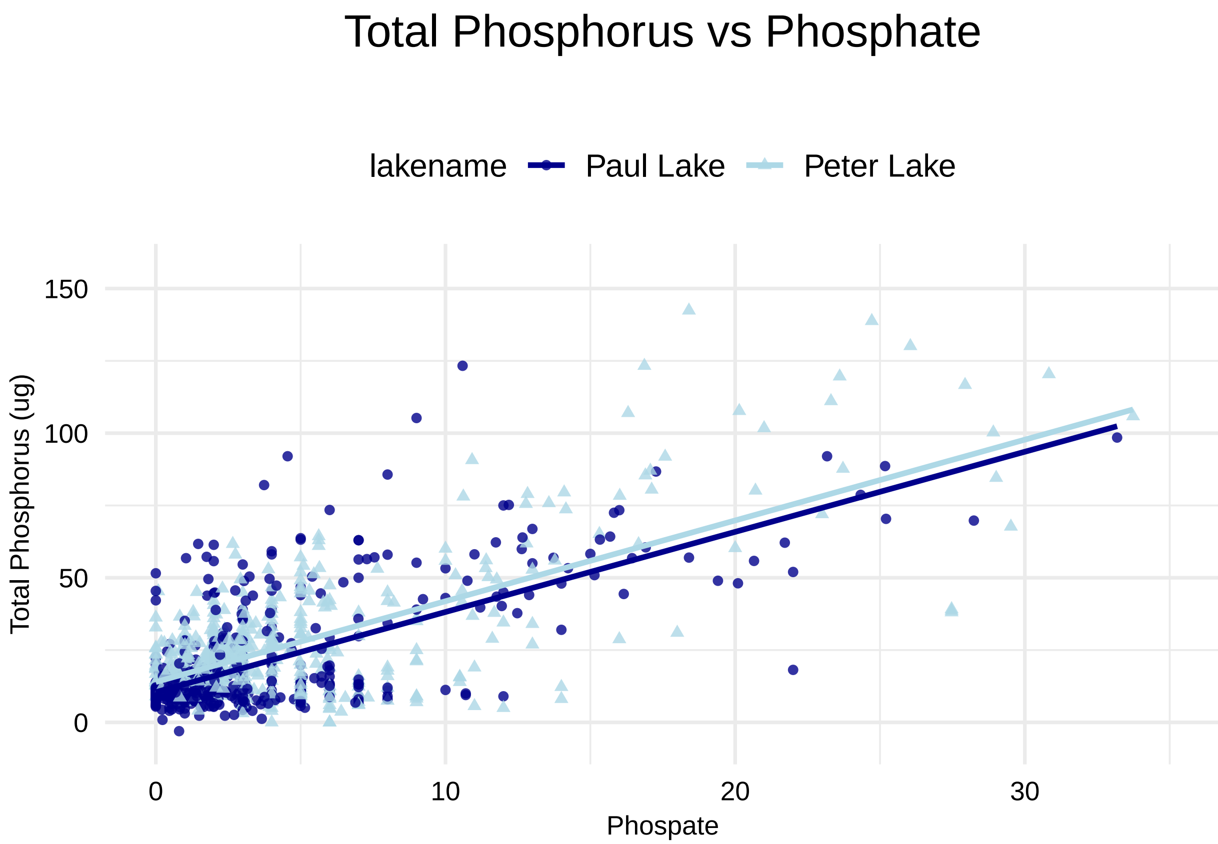
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
print(PvPO4)
```

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

```
## Warning: Removed 21951 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 21951 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.

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 months in integers to factors
PeterPaul.chem.nutrients_processed$month <- factor(PeterPaul.chem.nutrients_processed$month,
    levels = 1:12, labels = month.abb)

NTL_LTER_temperature <- ggplot(PeterPaul.chem.nutrients_processed,
    aes(x = month, y = temperature_C)) + geom_boxplot(aes(color = lakename),
    alpha = 0.8) + scale_color_manual(values = c("darkblue",
    "lightblue")) + labs(title = "Temperature each Month\n",
    y = "Temperature (C)", x = "Month", color = "Lake Name") +
    mytheme2

NTL_LTER_TP <- ggplot(PeterPaul.chem.nutrients_processed,
    aes(x = month, y = tp_ug)) + geom_boxplot(aes(color = lakename),
    alpha = 0.8, show.legend = FALSE) + scale_color_manual(values = c("darkblue",
    "lightblue")) + labs(title = "Total Phosphorus each Month\n",
    y = "Total Phosphorus", x = "Month", color = "Lake Name") +
    theme(legend.position = "none", axis.title.x = element_blank()) +
    mytheme

NTL_LTER_TN <- ggplot(PeterPaul.chem.nutrients_processed,
    aes(x = month, y = tn_ug)) + geom_boxplot(aes(color = lakename),
    alpha = 0.8, show.legend = FALSE) + scale_color_manual(values = c("darkblue",
    "lightblue")) + labs(title = "Total Nitrogen each Month\n",
    y = "Total Nitrogen (ug)", x = "Month", color = "Lake Name") +
    mytheme2

print(NTL_LTER_temperature)
```

```
## Warning: Removed 3566 rows containing non-finite values (‘stat_boxplot()‘).
```

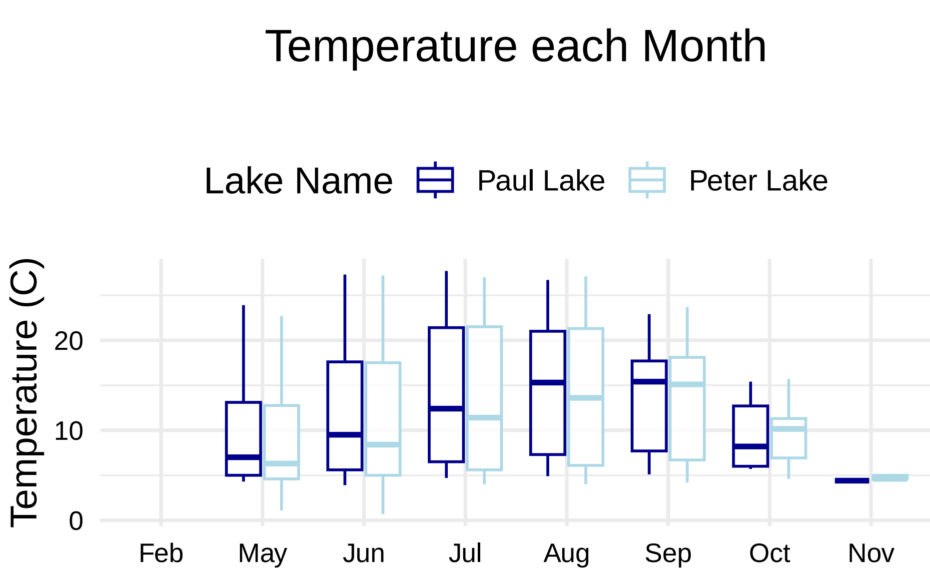


Figure 1: Figure 1

```r
print(NTL_LTER_TP)
```

```
## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).
```

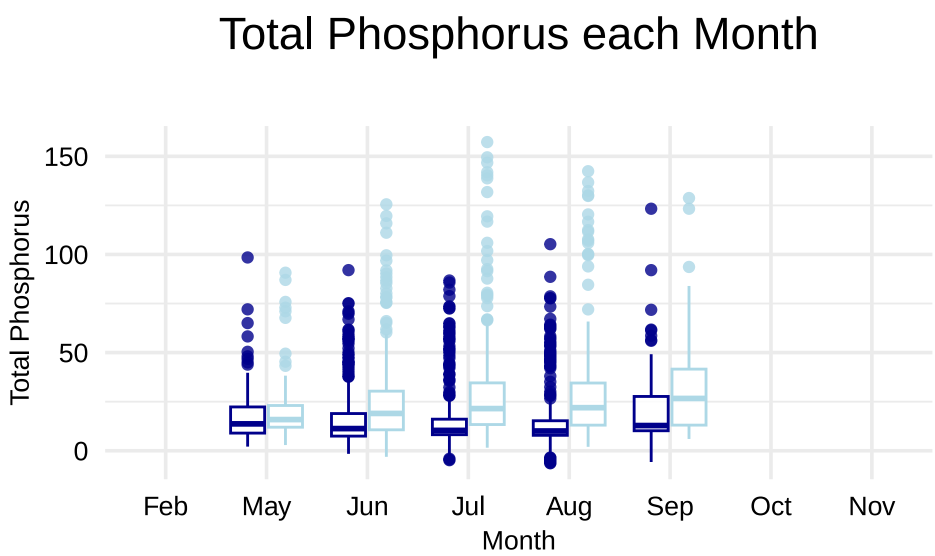


Figure 2: Figure 1

```r
print(NTL_LTER_TN)
```

```
## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).
```

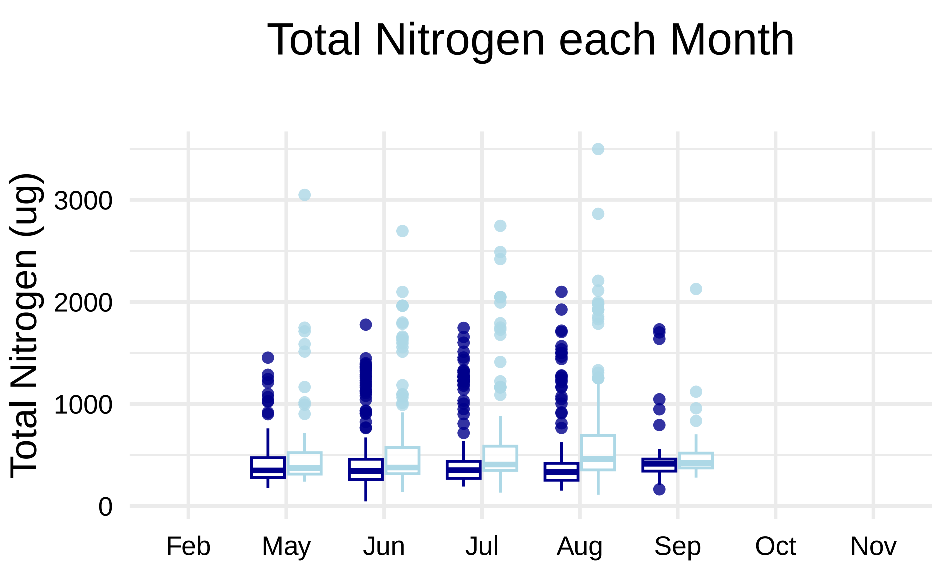


Figure 3: Figure 1

```r
plot_grid(NTL_LTER_temperature, NTL_LTER_TN, NTL_LTER_TP,
    nrow = 3, align = "h", rel_heights = c(1, 1, 1))
```

```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 21583 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 20729 rows containing non-finite values (`stat_boxplot()`).

## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

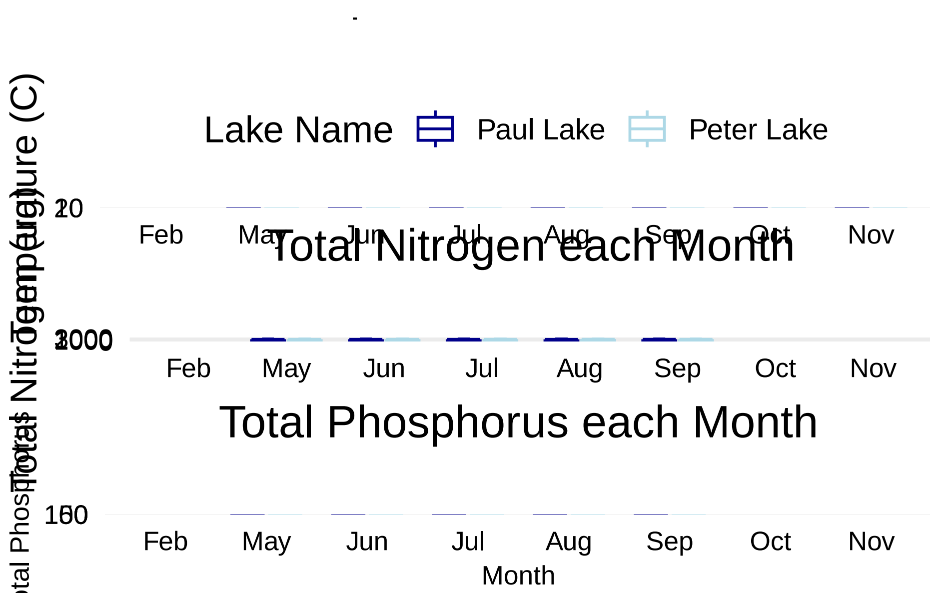


Figure 4: Figure 1

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

> Answer: Over seasons and between lakes, there is generally more variability in the total phorphorus and nitrogren by month in Peter Lake than in Paul lake. As temperature rises, total P and nitrogen also slightly increase and then decrease as the temperature decreases, however, that trend is slightly harder to see without data from October, Novemeber, and Decemeber.

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.

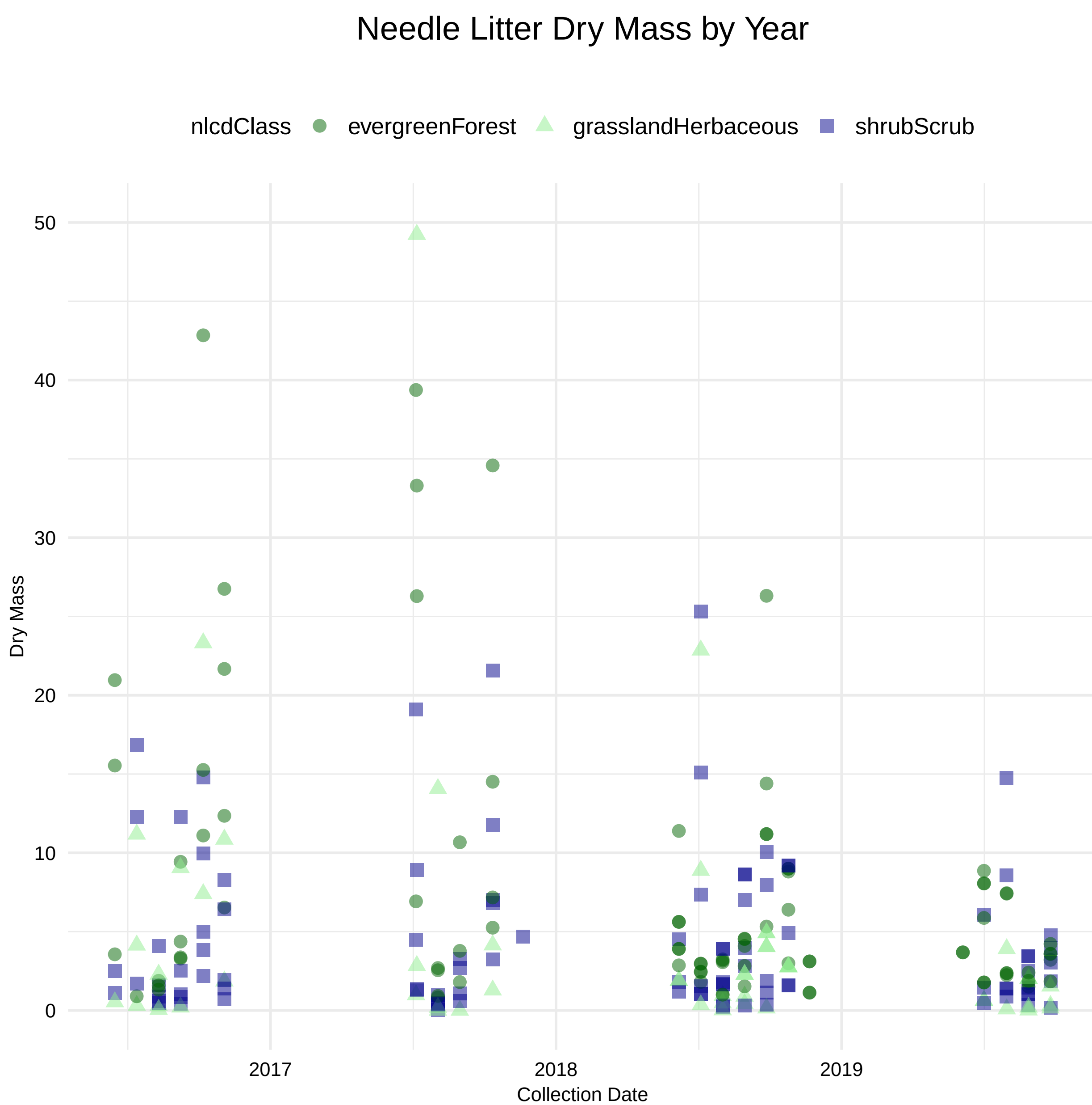
```r
# 6

needles_subset <- Niwot_Ridge_Litter_processed %>%
    filter(functionalGroup == "Needles")
```

```r
# Plot the dry mass of needle litter by date and
# separate by NLCD class with a color aesthetic
needles_plot <- ggplot(needles_subset, aes(x = collectDate,
    y = dryMass, color = nlcdClass, shape = nlcdClass)) +
    geom_point(alpha = 0.5, size = 3) + scale_color_manual(values = c("darkgreen",
    "lightgreen", "darkblue")) + labs(title = "Needle Litter Dry Mass by Year\n",
    y = "Dry Mass", x = "Collection Date", color = "nlcdClass") +
    ylim(0, 50) + mytheme

print(needles_plot)
```

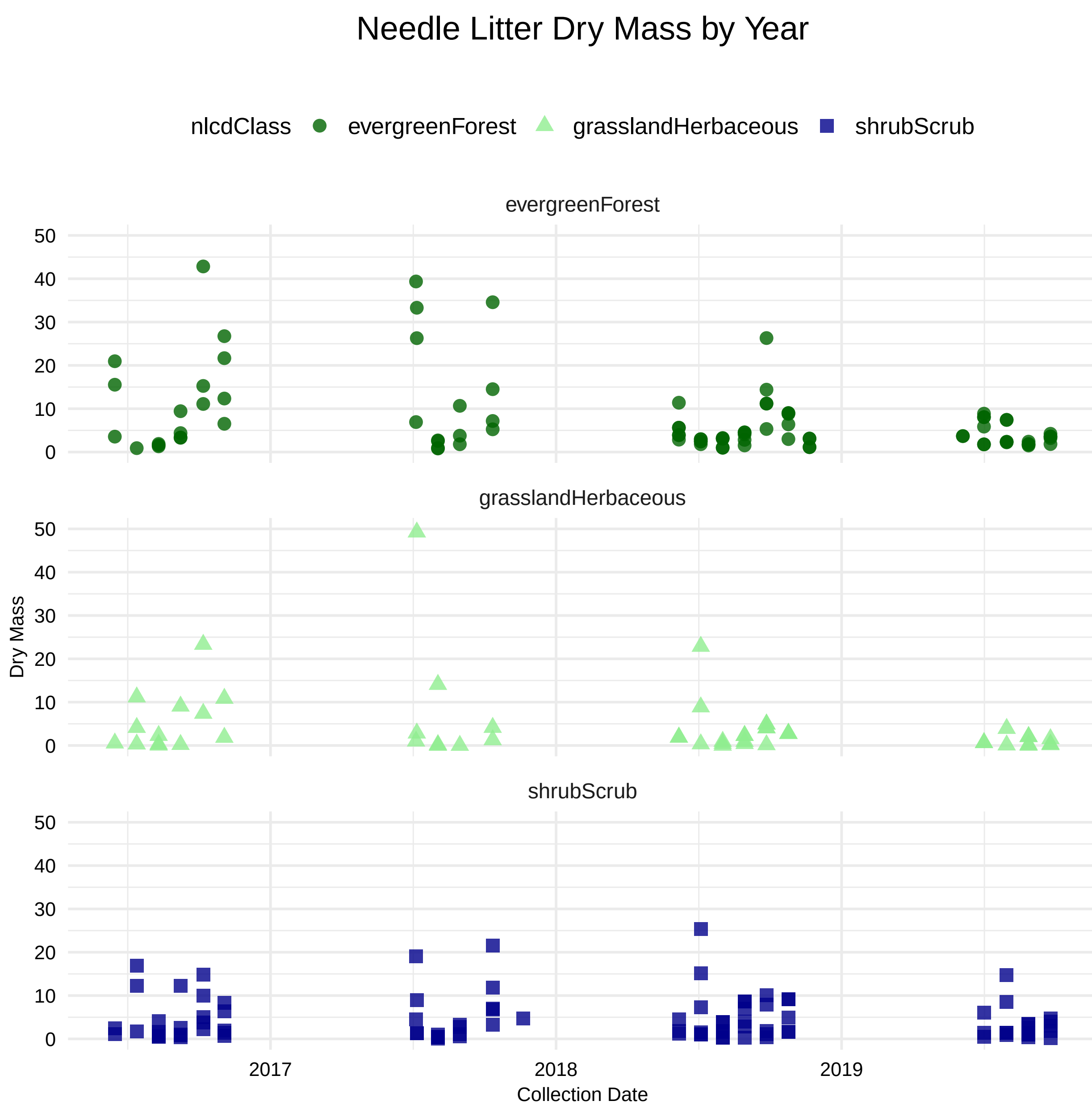
```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```



```r
# 7
needles_plot2 <- ggplot(needles_subset, aes(x = collectDate,
    y = dryMass, color = nlcdClass, shape = nlcdClass)) +
    geom_point(alpha = 0.8, size = 3) + scale_color_manual(values = c("darkgreen",
    "lightgreen", "darkblue")) + labs(title = "Needle Litter Dry Mass by Year\n",
    y = "Dry Mass", x = "Collection Date", color = "nlcdClass") +
    ylim(0, 50) + facet_wrap(vars(nlcdClass), nrow = 3) +
    mytheme

print(needles_plot2)
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```



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

Answer: I think it depends on what you are looking for, however, for general purposes 6 is more visually appealing, and the information is condensed/comparable, which makes observing trends easier.