Assignment 5: Data Visualization

Hiranmayi Nataraj

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.

```
#1
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
```

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

```
## here() starts at /home/guest/EDA_Spring2025
```

```
getwd()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
NTL <- read.csv(
file = here('Data','Processed','NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
stringsAsFactors = T
)

NEON <- read.csv(
file = here('Data','Processed','NEON_NIWO_Litter_mass_trap_Processed.csv'),
stringsAsFactors = T
)


#2

NTL$sampledate <- as.Date(NTL$sampledate)
NEON$collectDate <- as.Date(NEON$collectDate)

NTL$sampledate <- ymd(NTL$sampledate)
NEON$collectDate <- ymd(NEON$collectDate)
```

Define your theme

3.

```
#3

mytheme <- theme_classic(base_size = 10)+
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
```

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()`).

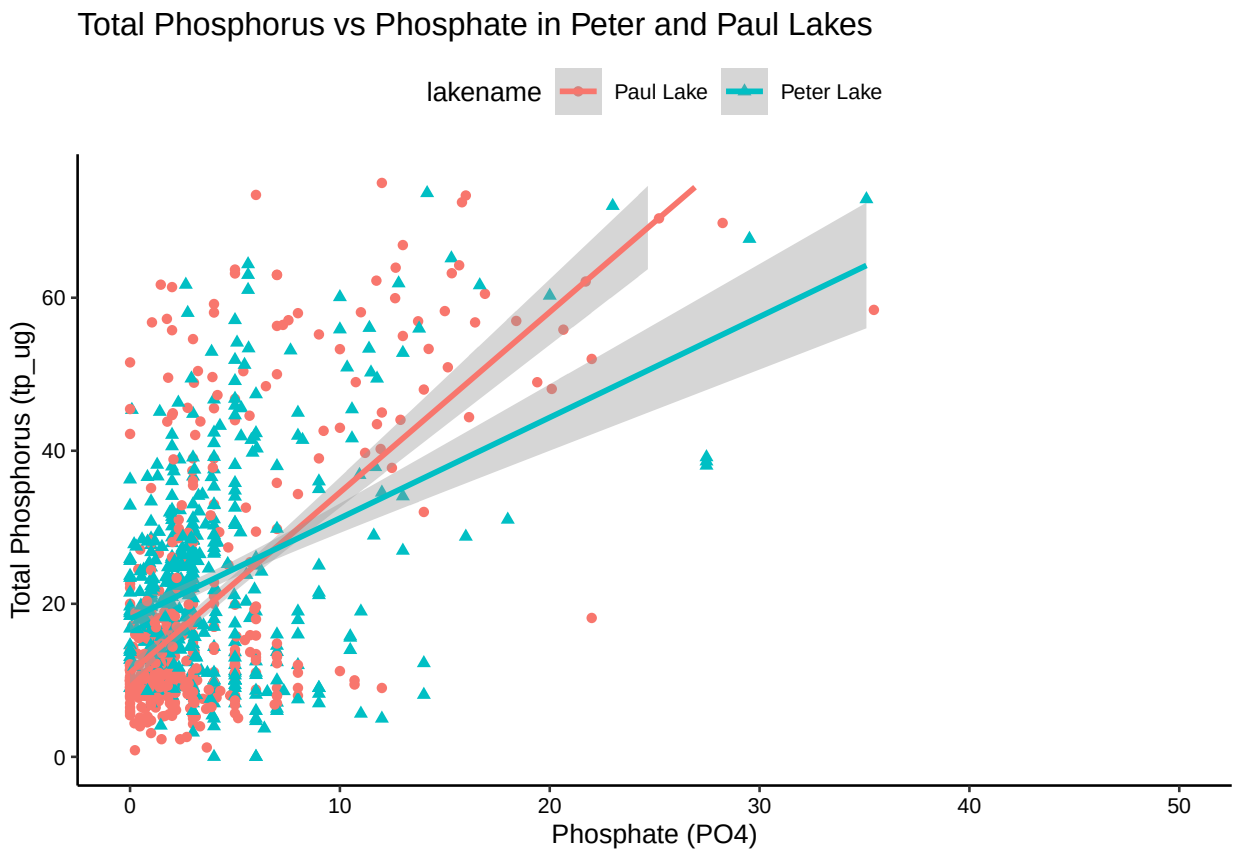
```r
#4
NTLp1 <-
  ggplot(NTL, aes(x = po4, y = tp_ug, , color = lakename, shape = lakename)) +
  geom_point() +
  geom_smooth(method = lm) +
labs(x = "Phosphate (PO4)", y = "Total Phosphorus (tp_ug)",
       title = "Total Phosphorus vs Phosphate in Peter and Paul Lakes") +
  mytheme +
  xlim(0, 50) +
  ylim(0, 75)
print(NTLp1)
```

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

```
## Warning: Removed 21988 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 21988 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 19 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`.

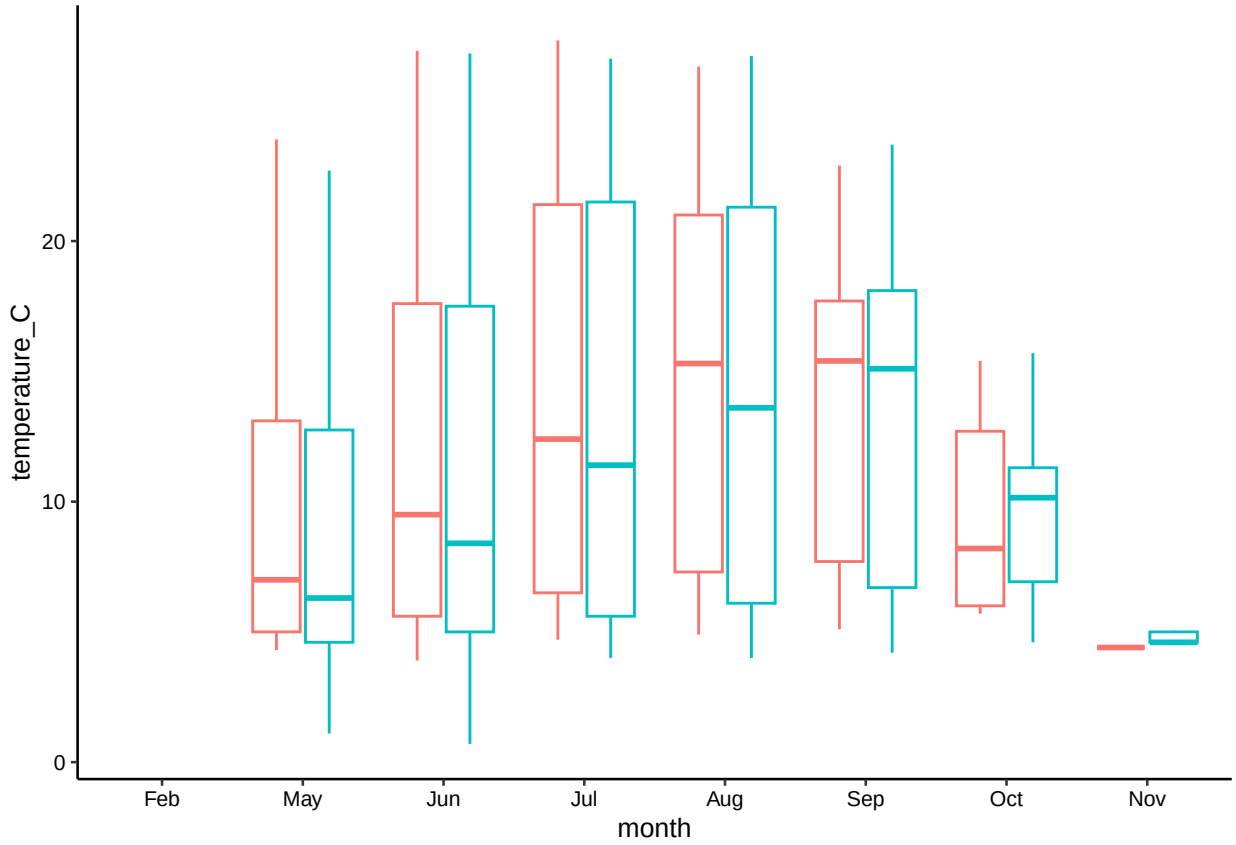
```
#5
library(ggplot2)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

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

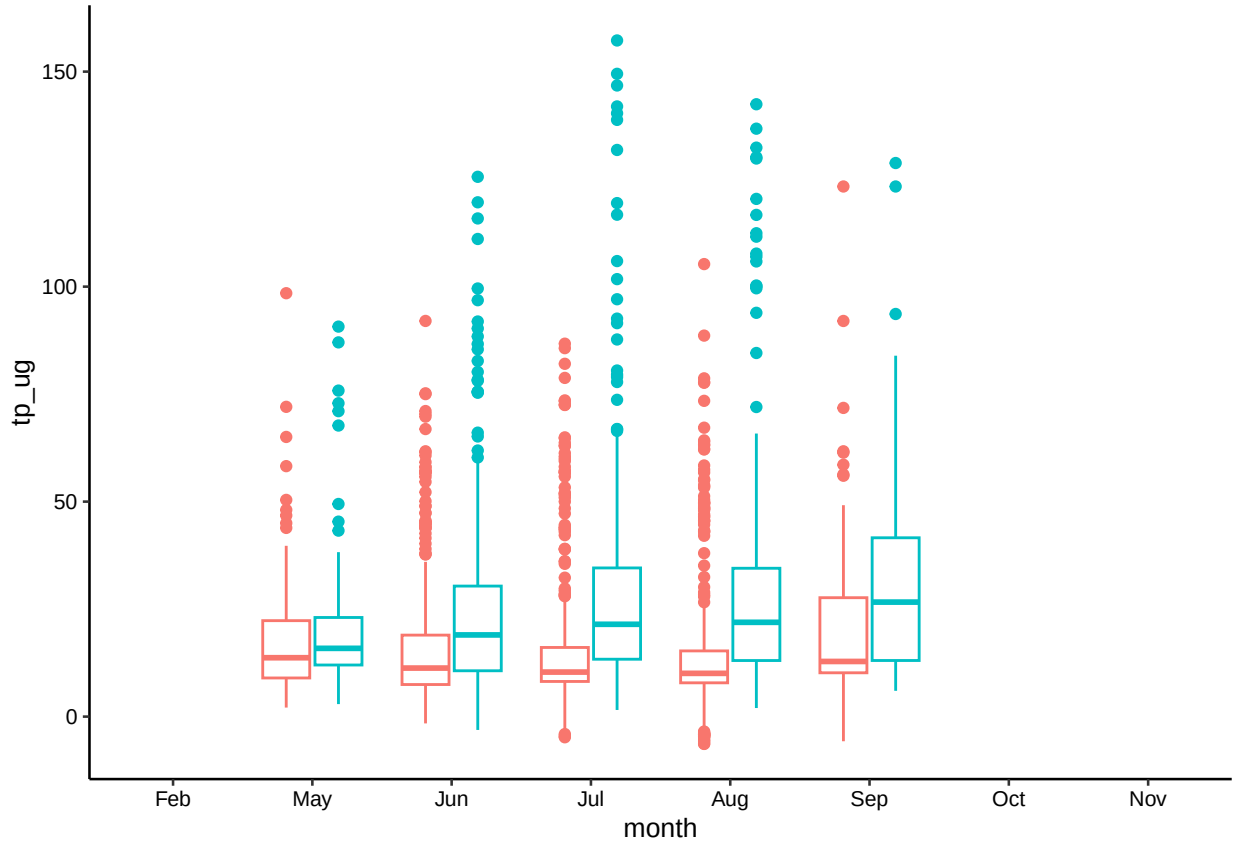
```
library(lubridate)

NTL$month <- factor(month(NTL$sampledate, label = TRUE), levels = month.abb)

temp_plot <- ggplot(NTL, aes(x = month, y = temperature_C, color = lakename)) +
  geom_boxplot() +
  mytheme +
  theme(legend.position = "none")
print(temp_plot)
```

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



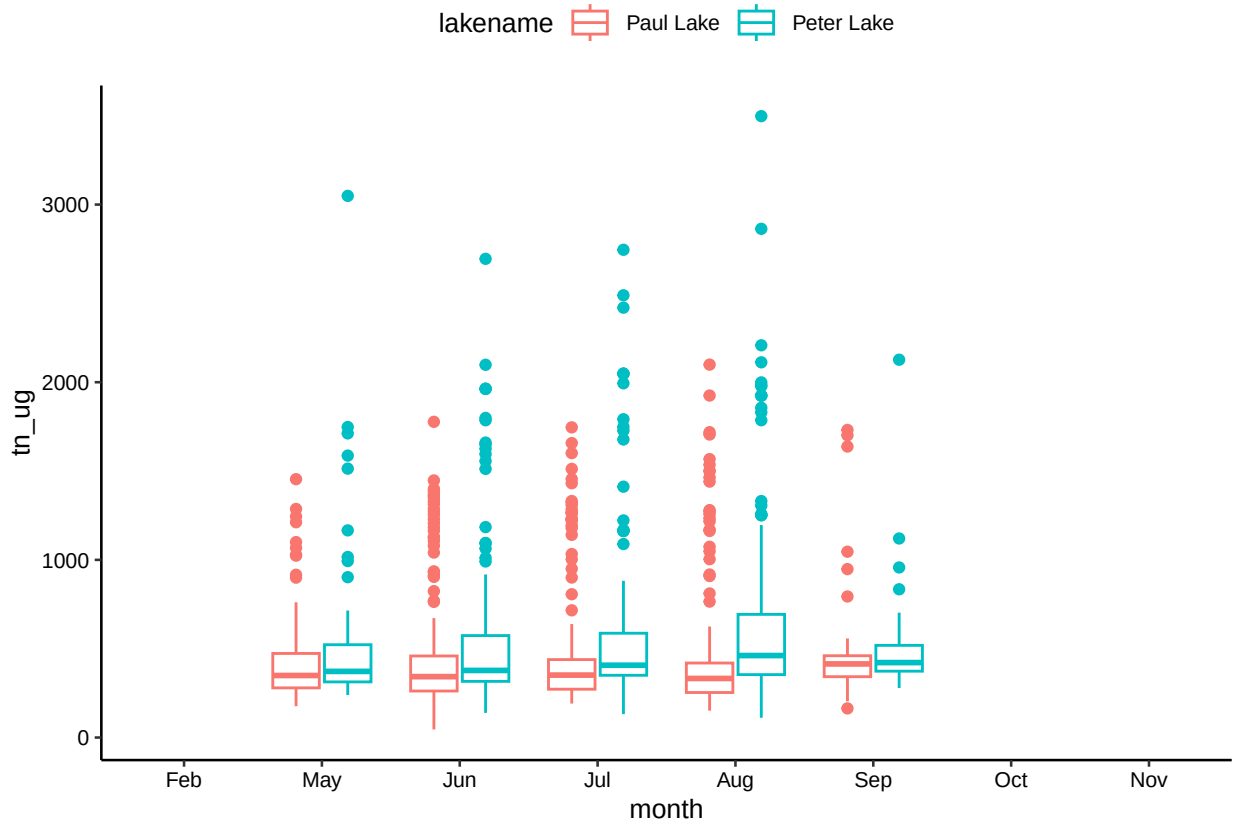

```
tp_plot <- ggplot(NTL, aes(x = month, y = tp_ug, color = lakename)) +
  geom_boxplot() +
  mytheme +
  theme(legend.position = "none")
print(tp_plot)
```

```
## Warning: Removed 20729 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
tn_plot <- ggplot(NTL, aes(x = month, y = tn_ug, color = lakename)) +
  geom_boxplot() +
  mytheme
print(tn_plot)
```

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




```
legend <- get_legend(tn_plot)
```

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

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

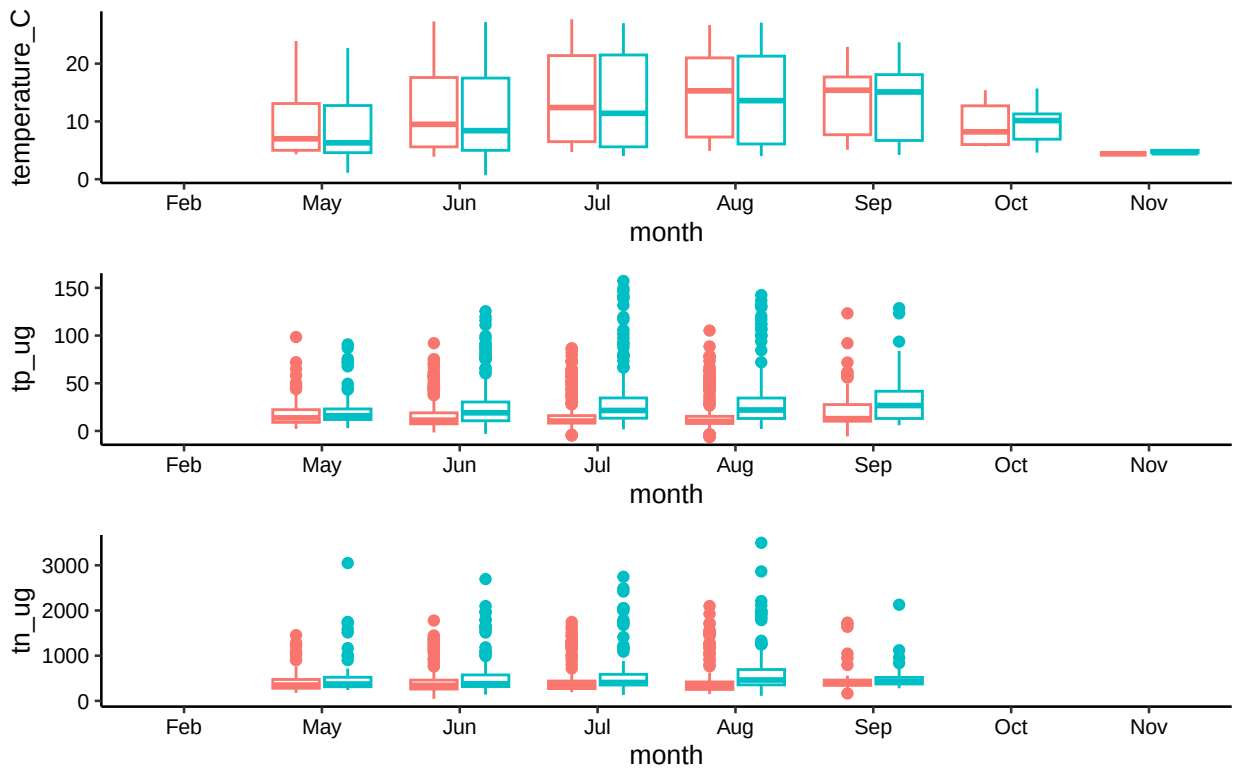
```
tn_plot <- tn_plot + theme(legend.position = "none")

final_plot <- plot_grid(
  temp_plot, tp_plot, tn_plot,
  ncol = 1, 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()`).
```

```
final_plot <- plot_grid(final_plot, legend, ncol = 1, rel_heights = c(3, 0.3))
print(final_plot)
```



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

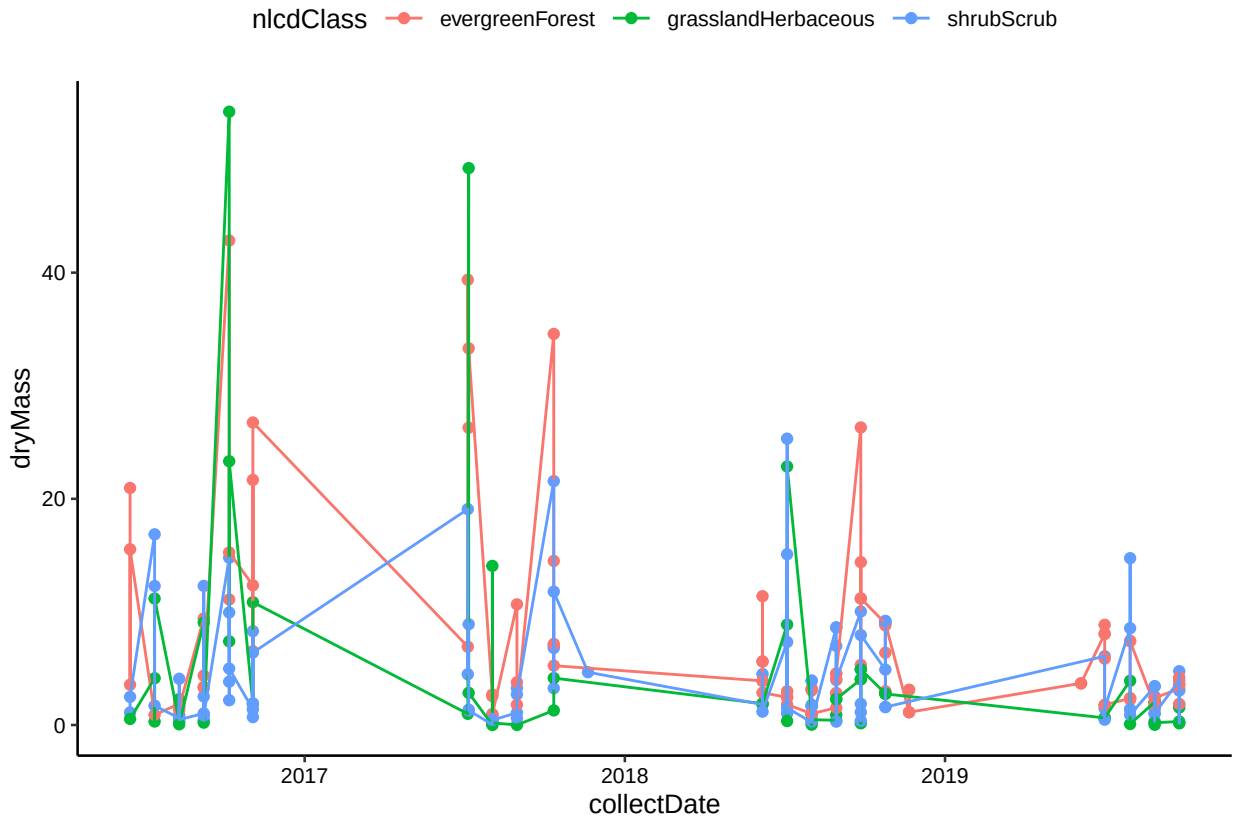
> Answer: a) Temperatures in the lakes as expected are higher in summer months. b) tp_ug levels seem to be higher in Peter Lake,with its median being higher than Paul Lake too. During summer months Paul lake has lower levels with its box plot lenghth also being lower, while Peter has higher levels and bigger box plots. c) tn_ug levels also seem to be higher in Peter lake than Paul lake, increasing as summer months progress. Paul lake levels are lower and decrease with summer 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)

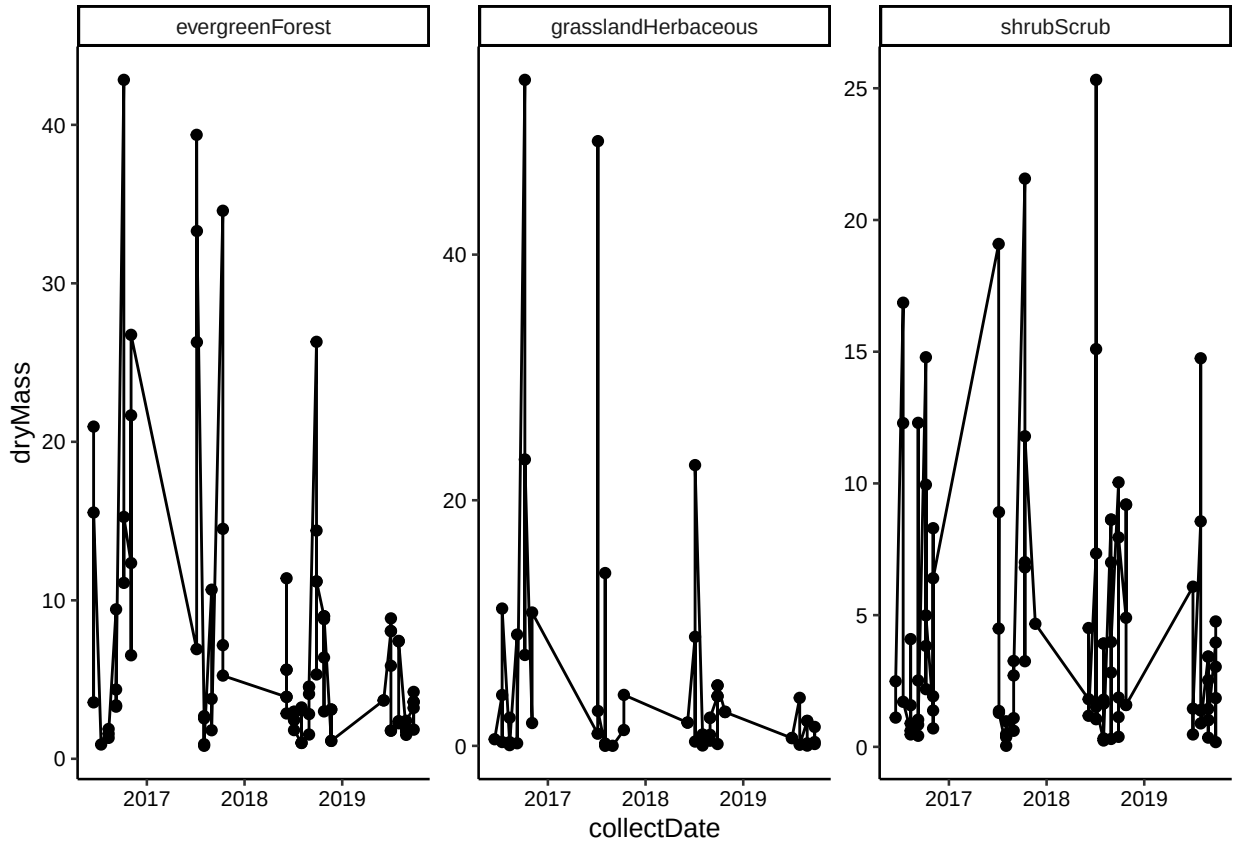
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

needles_data <- subset(NEON, functionalGroup == "Needles")

needles_plot <- ggplot(needles_data, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_point() +
```

```
  geom_line() +
  mytheme
print(needles_plot)
```




```
#7

needles_plot2 <- ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  geom_line() +
  facet_wrap(~ nlcdClass, scales = "free_y") +
  mytheme
print(needles_plot2)
```



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

Answer:7 is more effective cause it lets us view the three different types on their own across the years and lets us make comparisons.