WeilinWang_A05_Data_Visualization

Weilin Wang

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).

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

```r
#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 1.9.3     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/EDE_Fall2024
```

```
library(cowplot)
```

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

```
here::here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
chem_nutrients_file <- here("~/EDE_Fall2024/Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPa
litter_mass_file <- here("~/EDE_Fall2024/Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv")

lake_data <- read_csv(chem_nutrients_file)
```

```
## 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.
```

```
litter_data <- read_csv(litter_mass_file)
```

```
## 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.
```

```
head(lake_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>
```

```
head(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
lake_data <- lake_data %>%
  mutate(sampledate = as_date(as.numeric(sampledate), origin = "1970-01-01"))

litter_data <- litter_data %>%
  mutate(collectDate = as_date(as.numeric(collectDate), origin = "1970-01-01"))

str(lake_data)
```

```
## tibble [23,008 x 15] (S3: tbl_df/tbl/data.frame)
##  $ lakename       : chr [1:23008] "Paul Lake" "Paul Lake" "Paul Lake" "Paul Lake" ...
##  $ year4          : num [1:23008] 1984 1984 1984 1984 1984 ...
##  $ daynum         : num [1:23008] 148 148 148 148 148 148 148 148 148 148 ...
##  $ month          : num [1:23008] 5 5 5 5 5 5 5 5 5 5 ...
##  $ sampledate     : Date[1:23008], format: "1984-05-27" "1984-05-27" ...
##  $ depth          : num [1:23008] 0 0.25 0.5 0.75 1 1.5 2 3 4 5 ...
##  $ temperature_C  : num [1:23008] 14.5 NA NA NA 14.5 NA 14.2 11 7 6.1 ...
##  $ dissolvedOxygen: num [1:23008] 9.5 NA NA NA 8.8 NA 8.6 11.5 11.9 2.5 ...
##  $ irradianceWater: num [1:23008] 1750 1550 1150 975 870 610 420 220 100 34 ...
##  $ irradianceDeck : num [1:23008] 1620 1620 1620 1620 1620 1620 1620 1620 1620 1620 ...
##  $ tn_ug          : num [1:23008] NA NA NA NA NA NA NA NA NA NA ...
##  $ tp_ug          : num [1:23008] NA NA NA NA NA NA NA NA NA NA ...
##  $ nh34           : num [1:23008] NA NA NA NA NA NA NA NA NA NA ...
##  $ no23           : num [1:23008] NA NA NA NA NA NA NA NA NA NA ...
##  $ po4            : num [1:23008] NA NA NA NA NA NA NA NA NA NA ...
```

```
str(litter_data)
```

```
## tibble [1,692 x 13] (S3: tbl_df/tbl/data.frame)
##  $ plotID          : chr [1:1692] "NIWO_062" "NIWO_061" "NIWO_062" "NIWO_064" ...
##  $ trapID          : chr [1:1692] "NIWO_062_050" "NIWO_061_169" "NIWO_062_050" "NIWO_064_103" ...
##  $ collectDate     : Date[1:1692], format: "2016-06-16" "2016-06-16" ...
##  $ functionalGroup : chr [1:1692] "Seeds" "Other" "Woody material" "Seeds" ...
##  $ dryMass         : num [1:1692] 0 0.27 0.12 0 1.11 0 0 0 0.07 0.02 ...
##  $ qaDryMass       : chr [1:1692] "N" "N" "N" "N" ...
##  $ subplotID       : num [1:1692] 31 41 31 32 32 32 40 40 40 40 ...
##  $ decimalLatitude : num [1:1692] 40.1 40 40.1 40 40 ...
##  $ decimalLongitude: num [1:1692] -106 -106 -106 -106 -106 ...
##  $ elevation       : num [1:1692] 3477 3413 3477 3373 3446 ...
##  $ nlcdClass       : chr [1:1692] "shrubScrub" "evergreenForest" "shrubScrub" "evergreenForest" ...
##  $ plotType        : chr [1:1692] "tower" "tower" "tower" "tower" ...
##  $ geodeticDatum   : chr [1:1692] "WGS84" "WGS84" "WGS84" "WGS84" ...
```

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
my_theme <- theme(
  plot.background = element_rect(fill = "lightblue", color = NA),
  plot.title = element_text(size = 16, face = "bold", color = "darkblue"),
  axis.title.x = element_text(size = 14, color = "black"),
  axis.title.y = element_text(size = 14, color = "black"),
  axis.ticks = element_line(size = 1, color = "gray"),
  legend.position = "bottom",
  legend.background = element_rect(fill = "white", color = "black")
)
```

```
## 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(my_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()`).

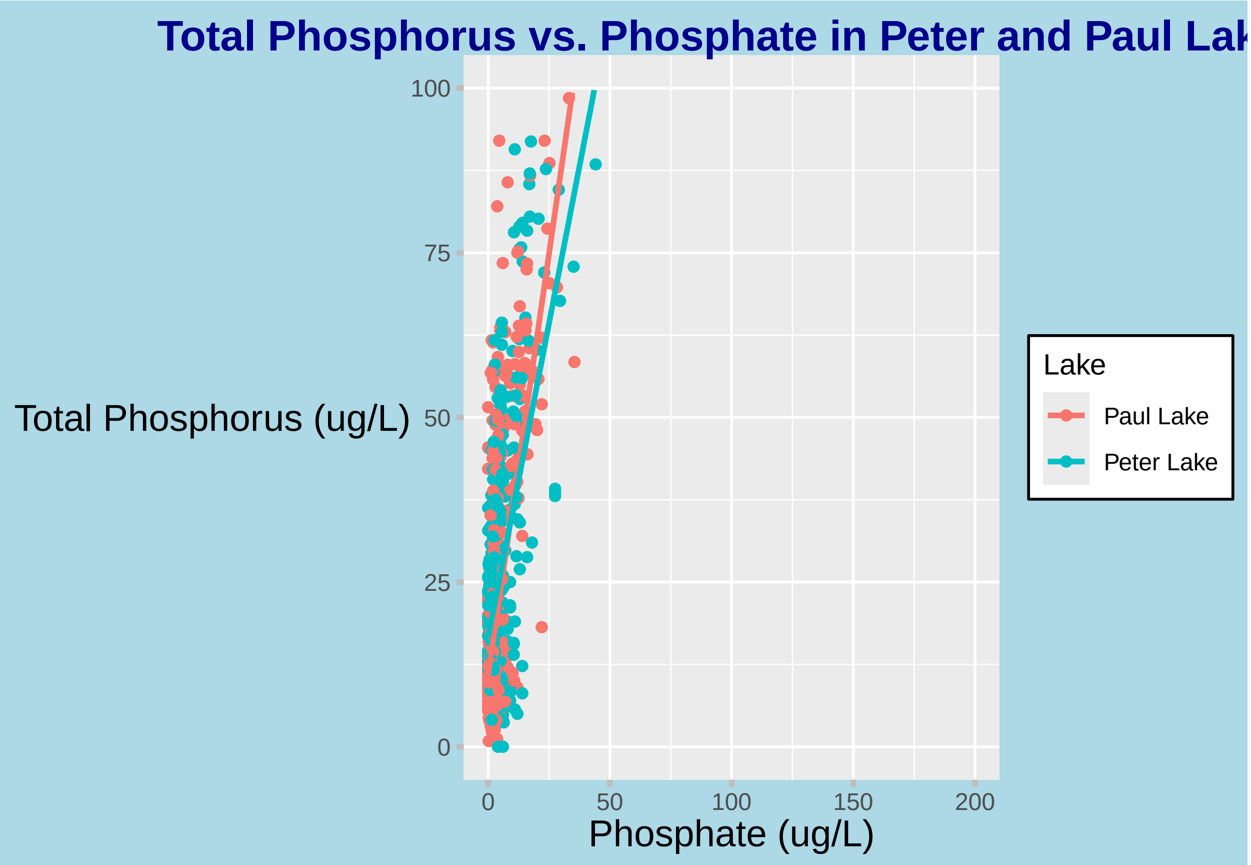
```r
#4
ggplot(lake_data, aes(x = po4, y = tp_ug, color = lakename)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +  # Line of best fit without confidence intervals
  xlim(0, 200) +  # Adjust x-axis limits (change as needed based on the actual data range)
  ylim(0, 100) +  # Adjust y-axis limits (change as needed based on the actual data range)
  labs(
    title = "Total Phosphorus vs. Phosphate in Peter and Paul Lakes",
    x = "Phosphate (ug/L)",
    y = "Total Phosphorus (ug/L)",
    color = "Lake"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    legend.position = "right"
  )
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 21964 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```

```
## Warning: Removed 21964 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

```
## Warning: Removed 3 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
lake_data <- lake_data %>%
  mutate(month = factor(month, levels = 1:12, labels = month.abb))
# Boxplot for Temperature
temp_plot <- ggplot(lake_data, aes(x = month, y = temperature_C, fill = lakename)) +
  geom_boxplot() +
  labs(title = "Temperature by Month", x = "Month", y = "Temperature (°C)", fill = "Lake") +
  theme(legend.position = "none")  # Remove legend for this plot

# Boxplot for Total Phosphorus (TP)
tp_plot <- ggplot(lake_data, aes(x = month, y = tp_ug, fill = lakename)) +
  geom_boxplot() +
  labs(title = "Total Phosphorus (TP) by Month", x = "Month", y = "TP (ug/L)") +
  theme(legend.position = "none")  # Remove legend for this plot

# Boxplot for Total Nitrogen (TN)
```

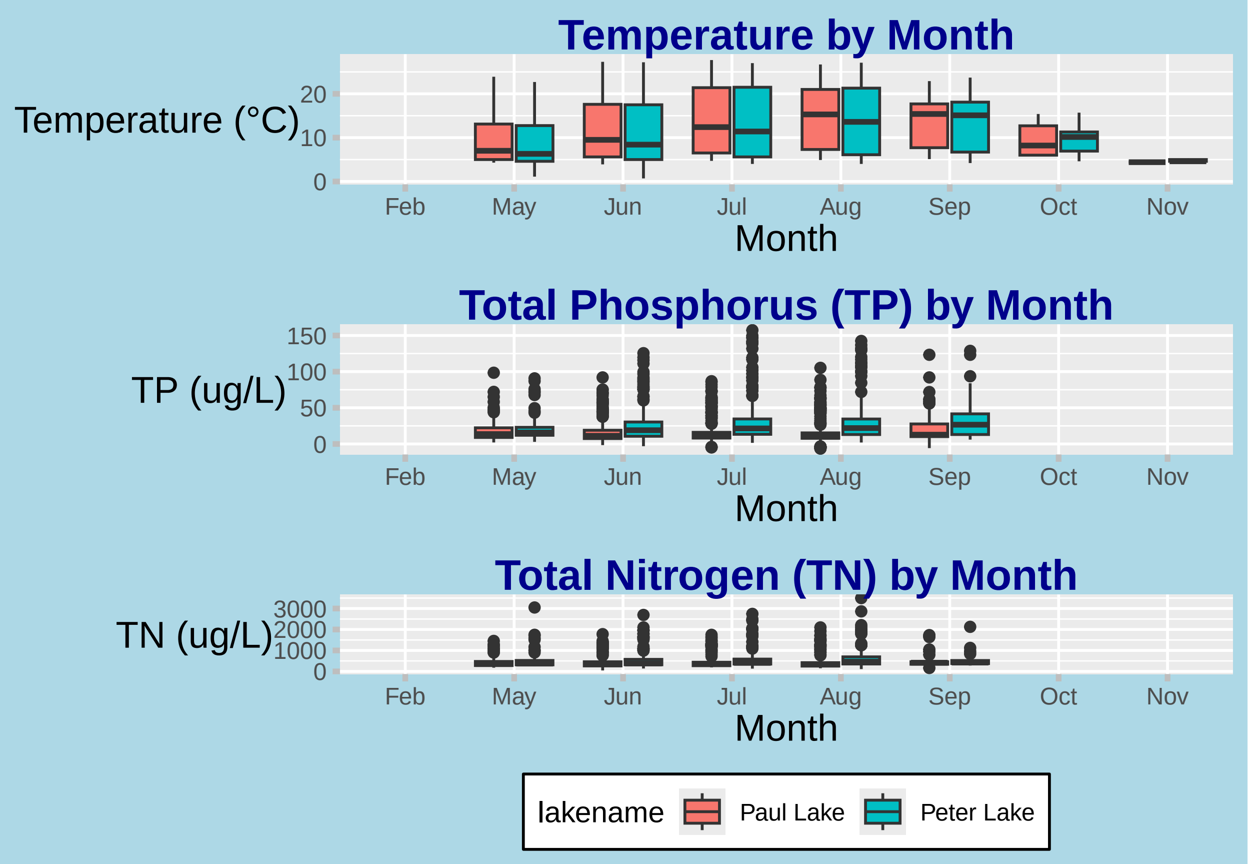
```
tn_plot <- ggplot(lake_data, aes(x = month, y = tn_ug, fill = lakename)) +
  geom_boxplot() +
  labs(title = "Total Nitrogen (TN) by Month", x = "Month", y = "TN (ug/L)") +
  theme(legend.position = "bottom")  # Keep legend only for this plot

library(cowplot)

combined_plot <- plot_grid(
  temp_plot,
  tp_plot,
  tn_plot,
  ncol = 1,  # Arrange in one column
  align = 'v',  # Align vertically
  axis = 'lr',  # Align left and right axes
  rel_heights = c(1, 1, 1.2)  # Make the last plot slightly bigger for the legend
)
```

```
## 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()`).
```

```
combined_plot
```

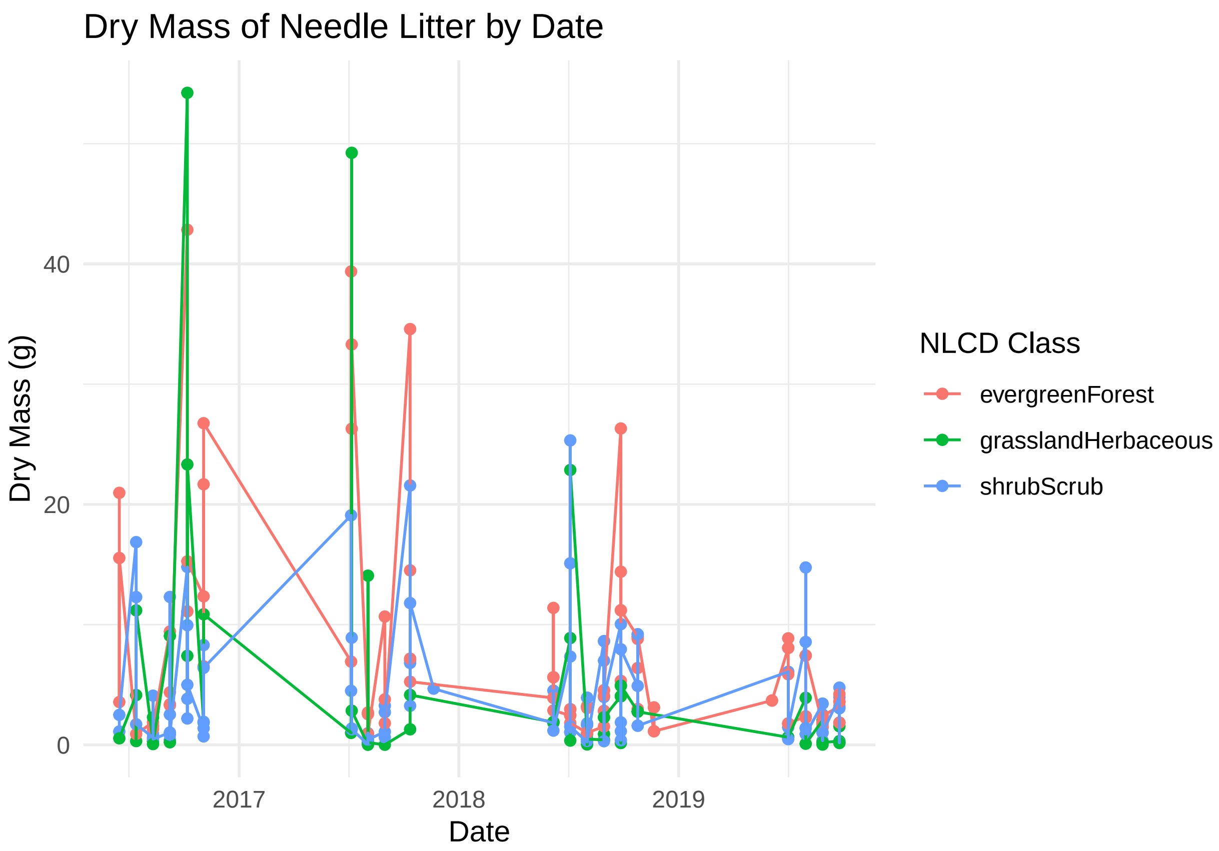


Question: What do you observe about the variables of interest over seasons and between lakes?

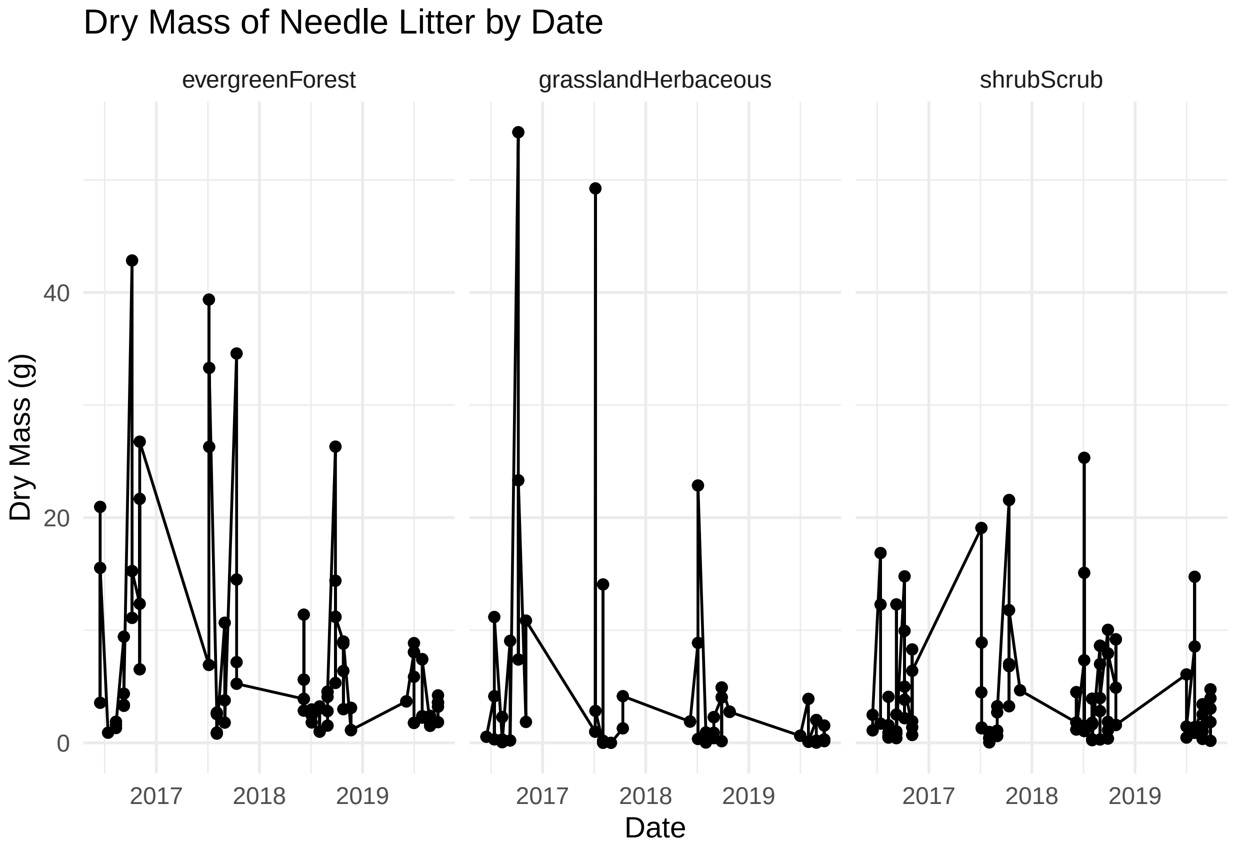
> Answer: the temperature in both Peter and Paul Lakes follows a consistent trend with some seasonal variability, showing higher values in the warmer months. For both TP and TN, there seems to be more variability in nutrient levels, with Peter Lake generally showing lower levels than Paul Lake, and extreme values present in certain 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.

```r
#6
needles_data <- litter_data %>%
  filter(functionalGroup == "Needles")

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



```
#7
# Plot dry mass of needle litter by date, with NLCD class separated into facets
ggplot(needles_data, aes(x = collectDate, y = dryMass)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Dry Mass of Needle Litter by Date",
    x = "Date",
    y = "Dry Mass (g)"
  ) +
  facet_wrap(~ nlcdClass) +
  theme_minimal()
```



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

Answer:The faceted plot (7) is more effective because it clearly separates trends by NLCD class, reducing visual clutter and making comparisons easier.