Assignment 5: Data Visualization

Nagarajan Vaidya Subramanian

Spring 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. Upload 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) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version).

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

```
#1
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ ggplot2 3.4.0     ✓ purrr   1.0.1
## ✓ tibble  3.1.8     ✓ dplyr   1.1.0
## ✓ tidyr   1.3.0     ✓ stringr 1.5.0
## ✓ readr   2.1.3     ✓ forcats 1.0.0
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
```

```
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#install.packages("here")
library(here)
```

```
## here() starts at X:/ENV 872 Environmental Data Analytics/Git_codes/EDA-
Spring2023
```

```
#install.packages("cowplot")
library(cowplot)
```

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

```
getwd()
```

```
## [1] "X:/ENV 872 Environmental Data Analytics/Git_codes/EDA-Spring2023"
```

```
processedDir = "./Data/Processed"

ntl_litter <- read.csv(here(processedDir, "NTL-
LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"))
litter <- read.csv(here(processedDir,
"NEON_NIWO_Litter_mass_trap_Processed.csv"))

#2
#checking the format in which different fields are saved.
glimpse(ntl_litter)
```

```
## Rows: 23,008
## Columns: 15
## $ lakename        <chr> "Paul Lake", "Paul Lake", "Paul Lake", "Paul
Lake", "P…
## $ year4           <int> 1984, 1984, 1984, 1984, 1984, 1984, 1984, 1984,
1984, …
## $ daynum          <int> 148, 148, 148, 148, 148, 148, 148, 148, 148, 148,
148,…
## $ month           <int> 5, 5, 5, 5, 5, 5, 5, 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           <dbl> 0.00, 0.25, 0.50, 0.75, 1.00, 1.50, 2.00, 3.00,
4.00, …
## $ temperature_C   <dbl> 14.5, NA, NA, NA, 14.5, NA, 14.2, 11.0, 7.0, 6.1,
5.5,…
## $ dissolvedOxygen <dbl> 9.5, NA, NA, NA, 8.8, NA, 8.6, 11.5, 11.9, 2.5,
1.6, 0…
## $ irradianceWater <dbl> 1750.0, 1550.0, 1150.0, 975.0, 870.0, 610.0,
420.0, 22…
## $ irradianceDeck  <dbl> 1620, 1620, 1620, 1620, 1620, 1620, 1620, 1620,
1620, …
## $ tn_ug           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA…
## $ tp_ug           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA…
## $ nh34            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA…
## $ no23            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA…
## $ po4             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA…

print("**********")

## [1] "**********"

glimpse(litter)

## Rows: 1,692
## Columns: 13
## $ plotID           <chr> "NIWO_062", "NIWO_061", "NIWO_062", "NIWO_064",
"NIWO…
## $ trapID           <chr> "NIWO_062_050", "NIWO_061_169", "NIWO_062_050",
"NIWO…
## $ collectDate      <chr> "2016-06-16", "2016-06-16", "2016-06-16", "2016-
06-16…
## $ functionalGroup  <chr> "Seeds", "Other", "Woody material", "Seeds",
"Needles…
## $ dryMass          <dbl> 0.000, 0.270, 0.120, 0.000, 1.110, 0.000, 0.000,
0.00…
## $ qaDryMass        <chr> "N", "N", "N", "N", "Y", "N", "N", "N", "N", "N",
"N"…
## $ subplotID        <int> 31, 41, 31, 32, 32, 32, 40, 40, 40, 40, 40, 31,
31, 3…
## $ decimalLatitude  <dbl> 40.05114, 40.04762, 40.05114, 40.04737, 40.04872,
40.…
## $ decimalLongitude <dbl> -105.5858, -105.5861, -105.5858, -105.5840, -
105.5872…
## $ elevation        <dbl> 3477.0, 3413.4, 3477.0, 3373.2, 3446.4, 3446.4,
```

```
3509.…
## $ nlcdClass        <chr> "shrubScrub", "evergreenForest", "shrubScrub",
"everg…
## $ plotType         <chr> "tower", "tower", "tower", "tower", "tower",
"tower",…
## $ geodeticDatum    <chr> "WGS84", "WGS84", "WGS84", "WGS84", "WGS84",
"WGS84",…

#converting the date column of the litter dataframe to date because it has
been read as characters in the yyyy-mm-dd format
litter$collectDate <- ymd(litter$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

```
#3
my_theme <- theme_minimal() +
  theme(
    legend.position = "bottom",
    legend.title = element_text(colour = "#440000"),
    plot.title = element_text(colour = "blue",
                              family = "Times"),
    axis.title = element_text(colour = "#4169e1",
                              family = "Times")
  )
```

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
ntl_litter %>%
  ggplot(aes(x=po4,
                 y=tp_ug,
                 color=lakename)) +
  geom_point() +
  geom_smooth(method = "lm") +
```

```
  xlim(0, 50) +
  ylim(0, 150) +
  labs(x = "Phosphate",
       y = "Total Phosphorous (in ug)") +
  my_theme
```

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

```
## Warning: Removed 21948 rows containing non-finite values
(`stat_smooth()`).
```

```
## Warning: Removed 21948 rows containing missing values (`geom_point()`).
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
family
## not found in Windows font database
```

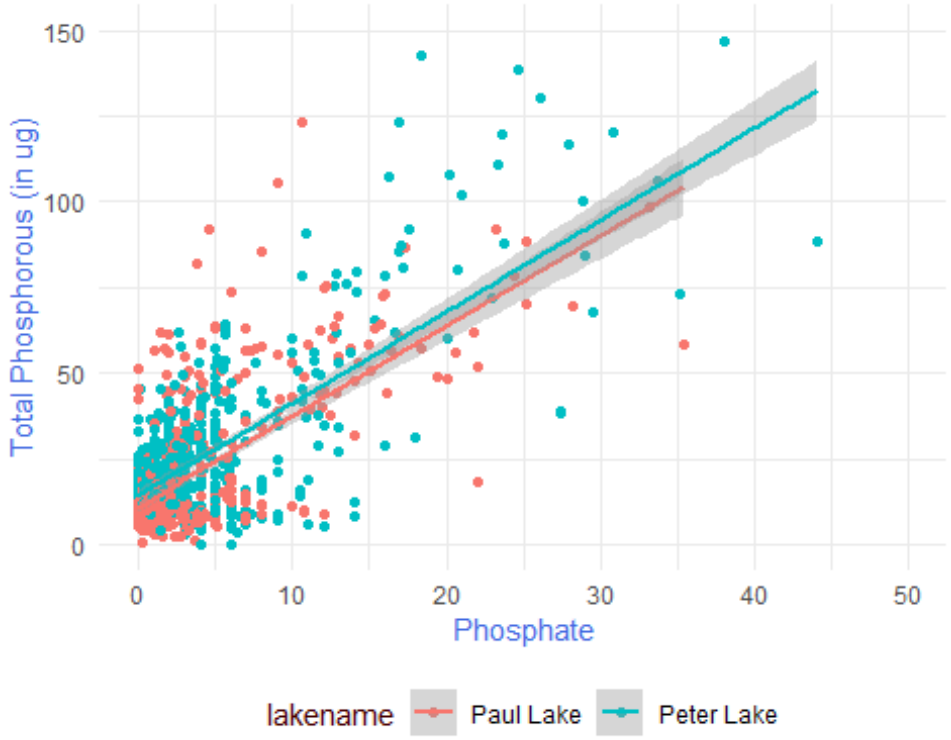
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database
```

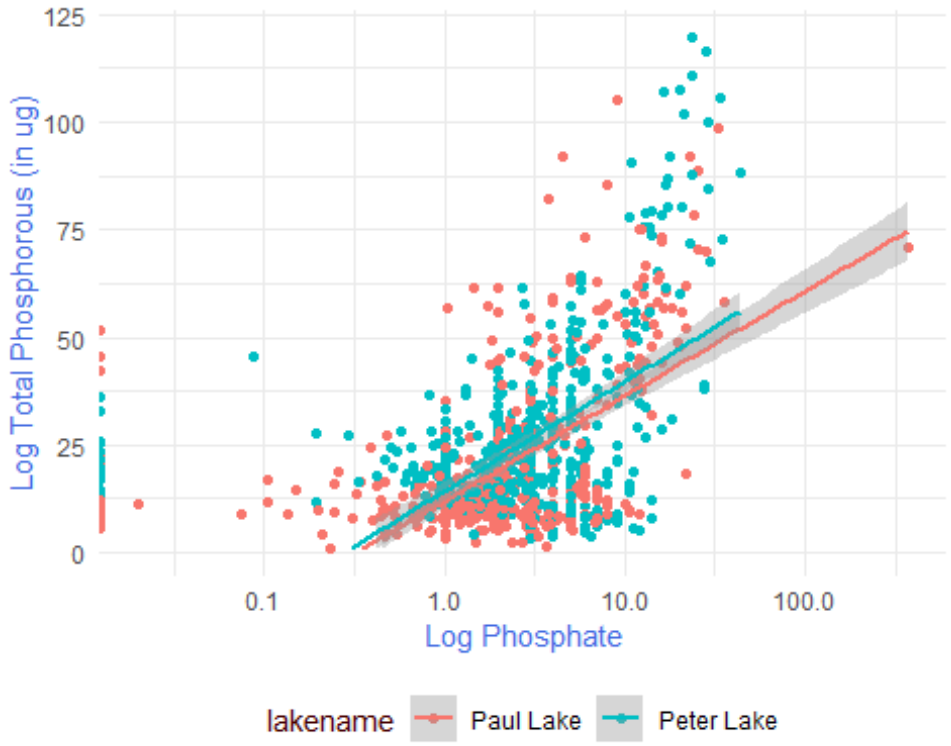
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database
```




```
#creating a log-log plot to visualise the data because it is very skewed
ntl_litter %>%
  ggplot(aes(x=po4,
                 y=tp_ug,
                 color=lakename)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlim(0, 50) +
  ylim(0, 150) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Log Phosphate",
       y = "Log Total Phosphorous (in ug)") +
  ylim(0.5,120) +
  my_theme
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous x-axis

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

## Warning: Removed 22019 rows containing non-finite values
(`stat_smooth()`).

## Warning: Removed 21957 rows containing missing values (`geom_point()`).

## Warning: Removed 39 rows containing missing values (`geom_smooth()`).

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database
```



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.

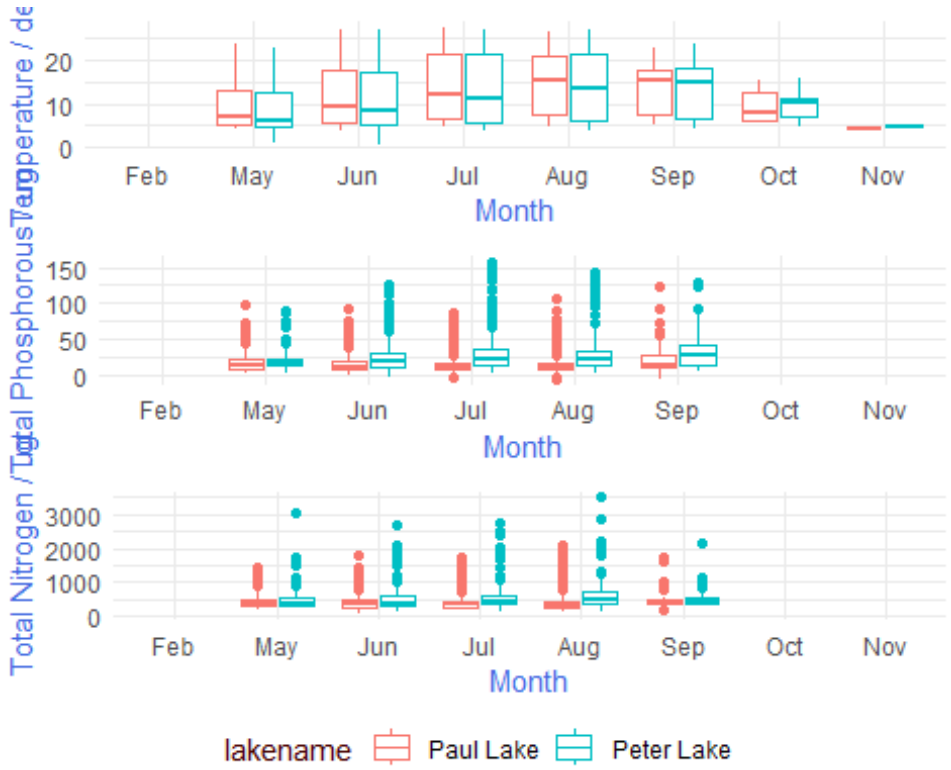
Tip: R has a build in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5
#Changing the format of the month column in the dataframe to Date so that it
acts as categories for later plots
ntl_litter$month <- month(ntl_litter$month, label=TRUE)
#ntl_litter$year4 <- year(ntl_litter$year4)

plot_temperature <-
  ntl_litter %>%
  group_by(month) %>%
  ggplot(mapping = aes(color=lakename)) +
    geom_boxplot(mapping=aes(x = month,
                             y = temperature_C)) +
    my_theme +
    labs(x = "Month",
         y = "Temperature / deg. C")

plot_tp <-
  ntl_litter %>%
```

```
  group_by(month) %>%
  ggplot(mapping = aes(color=lakename)) +
    geom_boxplot(mapping=aes(x = month,
                             y = tp_ug)) +
    my_theme +
    labs(x = "Month",
         y = "Total Phosphorous / ug")

plot_tn <-
  ntl_litter %>%
    group_by(month) %>%
    ggplot(mapping = aes(color=lakename)) +
    geom_boxplot(mapping=aes(x = month,
                             y = tn_ug)) +
    my_theme +
    labs(x = "Month",
         y = "Total Nitrogen / ug")

#first creating a combined plot of all three boxplots but without their
legends
combined_plot <- plot_grid(plot_temperature + theme(legend.position =
"none"),
                           plot_tp + theme(legend.position = "none"),
                           plot_tn + theme(legend.position = "none"),
                           axis = "b",
                           nrow = 3,
                           align = 'h')

## 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()`).

#creating a separate "plot" of the legend alone. Then combining it with our
combined plot of the three boxplots
legend <- get_legend(plot_temperature)

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

#creating a cowplot with the combined boxplots and a common legend. The
things we do for want of built-in functionality
my_cowplot <- plot_grid(combined_plot, legend,
                        ncol = 1,
                        rel_heights = c(1, 0.1))
my_cowplot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
```

```
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database
```



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

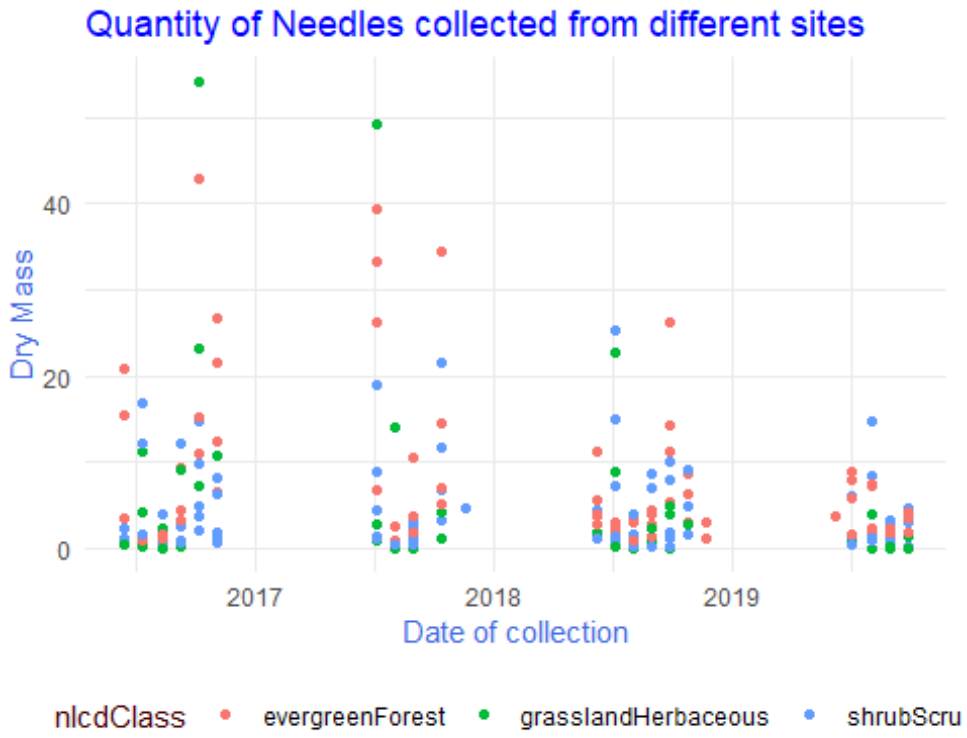
Answer: The variables of interest, which are temperature, total phosphorous and total nitrogen, increase in value in the warmer months of the year and decrease in the cooler months. Between the lakes, Peter Lake seems to be more sensitive to temperature as the range and value of the variables are higher for this lake compared to Paul Lake.

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)

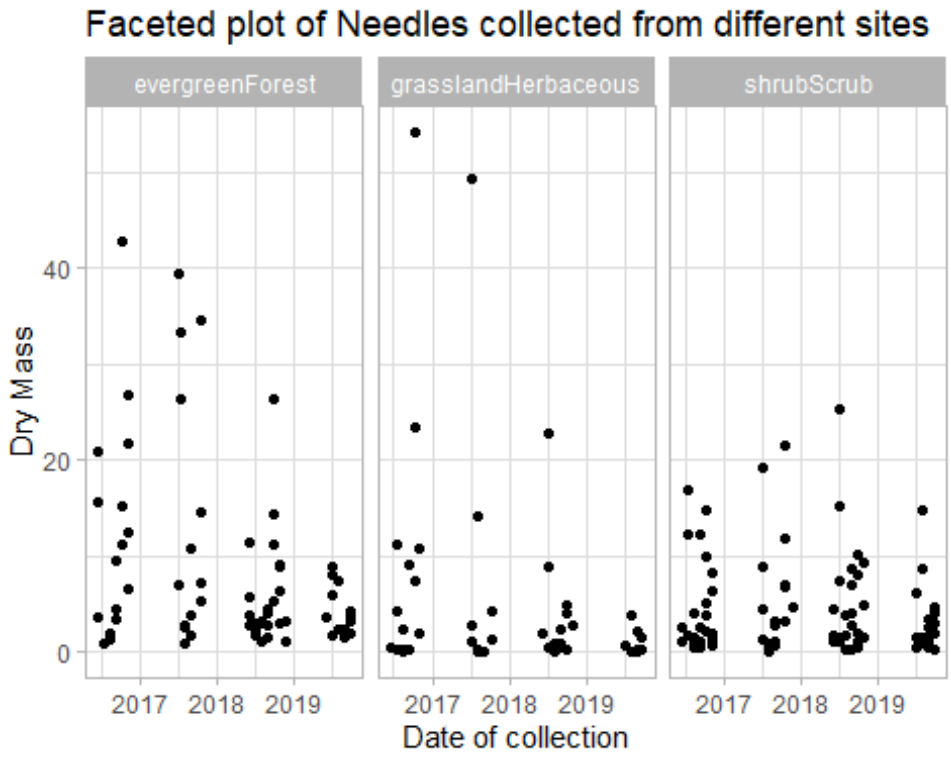
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(mapping = aes(x = collectDate,
                       y = dryMass,
                       colour = nlcdClass)) +
    geom_point() +
    labs(x = "Date of collection",
         y = "Dry Mass",
         title = "Quantity of Needles collected from different sites") +
    my_theme
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
family
## not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y,
:
## font family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
font
## family not found in Windows font database
```




```
#7
litter %>%
  filter(functionalGroup == "Needles") %>%
  ggplot(mapping = aes(x = collectDate,
                       y = dryMass)) +
    geom_point() +
    labs(x = "Date of collection",
         y = "Dry Mass",
         title = "Faceted plot of Needles collected from different sites") +
    facet_grid(. ~ nlcdClass) +
    theme_light()
```



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

Answer: I think the faceted plot is more effective at conveying the quantity of needles collected from different sites at different dates. This is because in the Cowplot, the points from the three locations overlap and it becomes hard to understand the distribution of mass across different NLCD classes. Whereas in the facet plot, the masses are plotted on separate windows which makes their relative distributions easier to visualise.