Assignment 5: Data Visualization

Tani Valdez Rivas

Fall 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. 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

# Loading the necessary libraries
library(tidyverse)
```

```
## -- 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()
## 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/R/EDE_Fall2023
```

```
library(cowplot)
```

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

```
# Verifying home directory
getwd()
```

```
## [1] "/home/guest/R/EDE_Fall2023"
```

```
# Reading in the csv files
Chemistry_Nutrients_PeterPaul <-
  read.csv(here("Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv"),
           stringsAsFactors = TRUE)

Niwot_Litter_Mass <-
  read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv"),
           stringsAsFactors = TRUE)

#2

# Determining the class of sampledate
class(Chemistry_Nutrients_PeterPaul$sampledate) # It is a factor
```

```
## [1] "factor"
```

```
class(Niwot_Litter_Mass$collectDate) # It is a factor
```

```
## [1] "factor"
```

```
# Changing sampledate from factor to date
Chemistry_Nutrients_PeterPaul_Sampledate <- ymd(Chemistry_Nutrients_PeterPaul$sampledate)
Niwot_Litter_Mass_collectDate <- ymd(Niwot_Litter_Mass$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


# Building my theme
mytheme <- theme_classic() +
  theme(
    plot.background = element_rect(
      fill = "grey"
    ),
    plot.title = element_text(
      color = "black",
      size = 12
    ),
    axis.text = element_text(
      color = "black",
      size = 12
    ),
    axis.ticks = element_line(
      color = "black",
      linewidth = 2
    ),
    legend.title = element_text(
      color = "black",
      size = 12
    )

  )
```

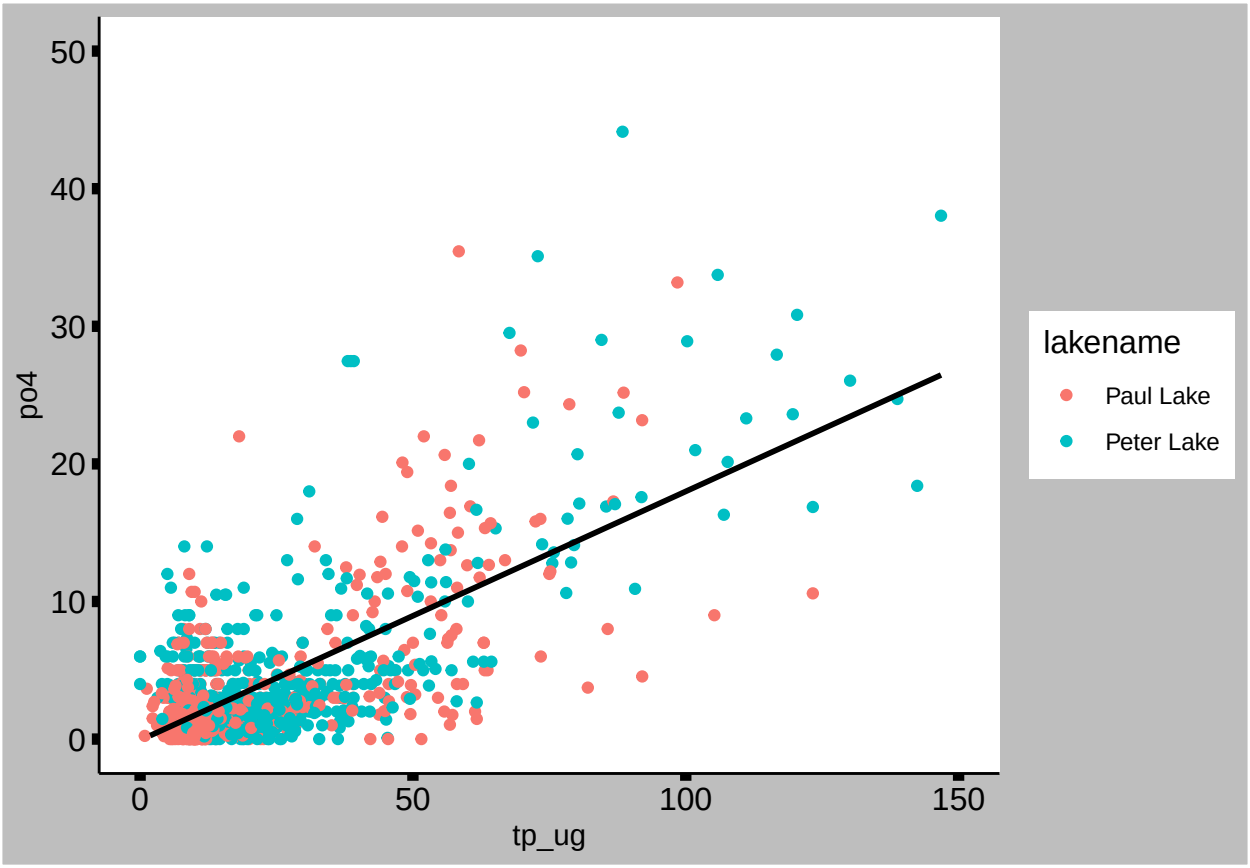
Create graphs

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

# Creating a ggplot of tp_ug vs po4
TP_by_PO4 <- ggplot(Chemistry_Nutrients_PeterPaul, (aes(x = tp_ug, y = po4, color = lakename))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  xlim(c(0, 150)) +
  ylim(c(0, 50)) +
  mytheme
TP_by_PO4
```

```
## `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: Removed 1 rows containing missing values (`geom_smooth()`).
```



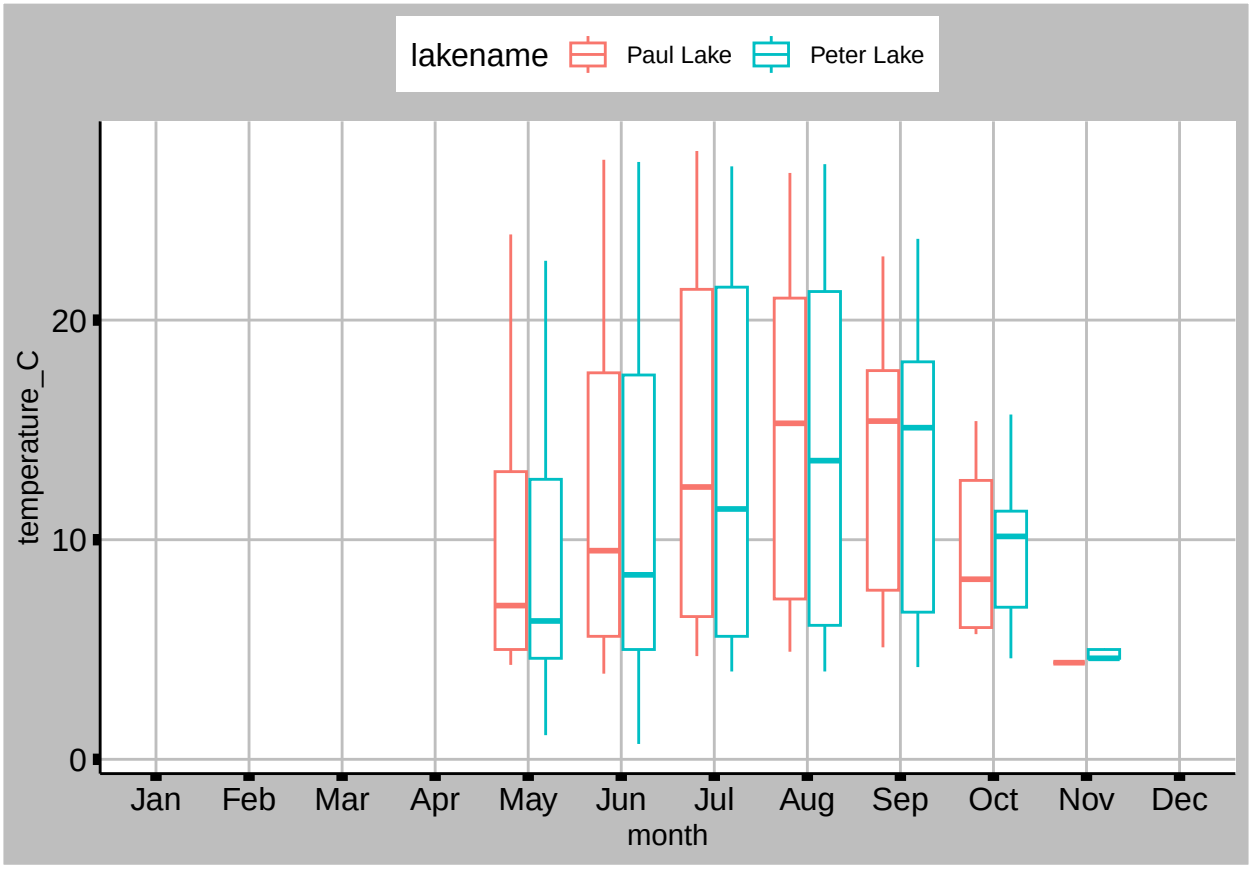
5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

Tip: * Recall the discussion on factors in the previous section as it may be helpful here. * R has a built-in variable called `month.abb` that returns a list of months;see https://r-lang.com/month-abb-in-r-with-example

```
#5

# Updating month from numeric value to character
Chemistry_Nutrients_PeterPaul$month <- factor(
  Chemistry_Nutrients_PeterPaul$month,
  levels = c("2", "5", "6", "7", "8", "9", "10", "11")
)

Chemistry_Nutrients_PeterPaul$month <- factor(
```

```
  Chemistry_Nutrients_PeterPaul$month,
  levels=c("1","2","3","4","5","6","7","8", "9", "10", "11", "12"),
  labels=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep",
           "Oct","Nov","Dec"))


# Creating three different box plots for Temp, TP, and TN
PeterPaul_Chem_Temp <-
  ggplot(Chemistry_Nutrients_PeterPaul,
         (aes(x = month, y = temperature_C, color = lakename))) +
  geom_boxplot() +
  scale_x_discrete(drop=FALSE) +
  mytheme +
  theme(legend.position ="top") +
  theme(panel.grid.major = element_line(color = "grey", linetype = "solid"))
  # Used ChatGPT for reference to add gridlines as other codes did not work
PeterPaul_Chem_Temp
```

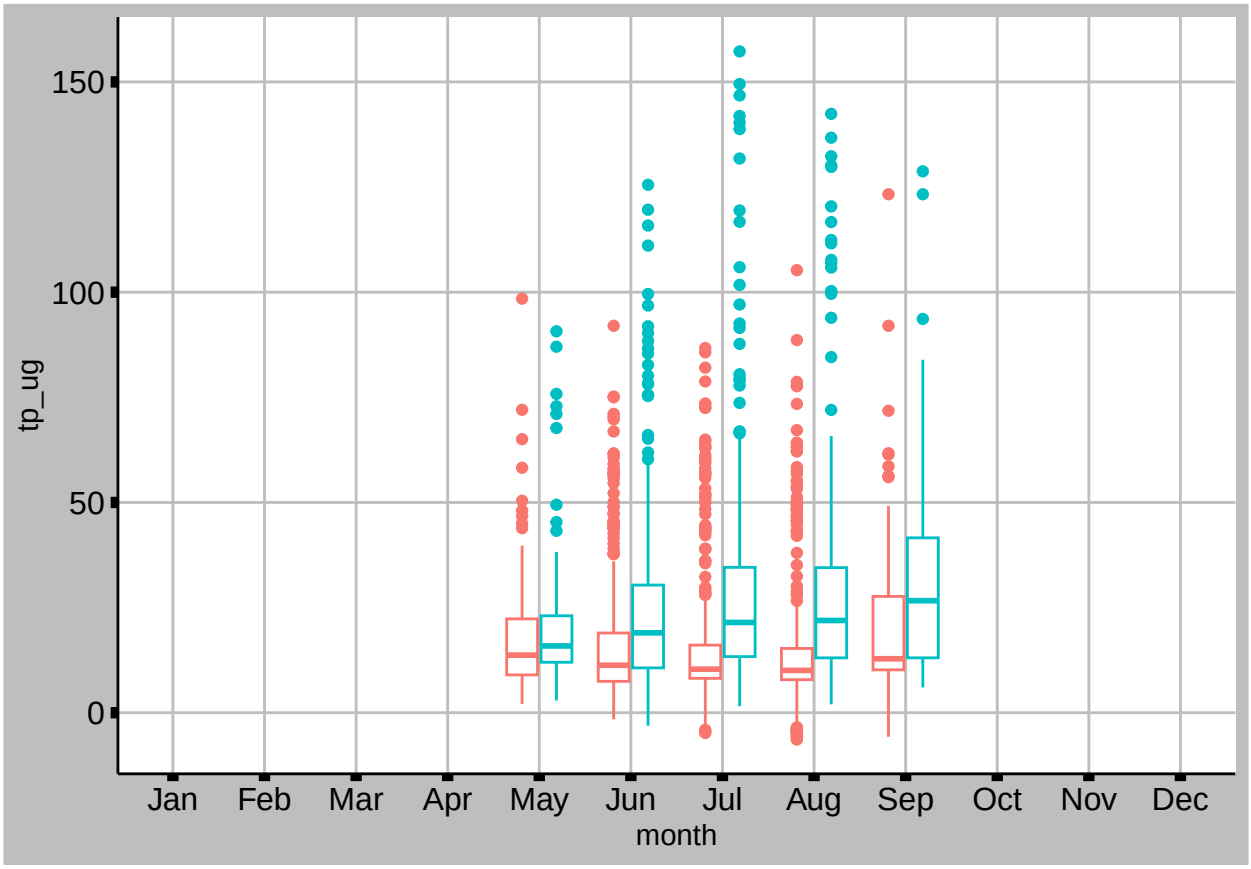
```
## Warning: Removed 3566 rows containing non-finite values (`stat_boxplot()`).
```




```
PeterPaul_Chem_TP <-
  ggplot(Chemistry_Nutrients_PeterPaul,
         (aes(x = month, y = tp_ug, color = lakename))) +
  geom_boxplot() +
```

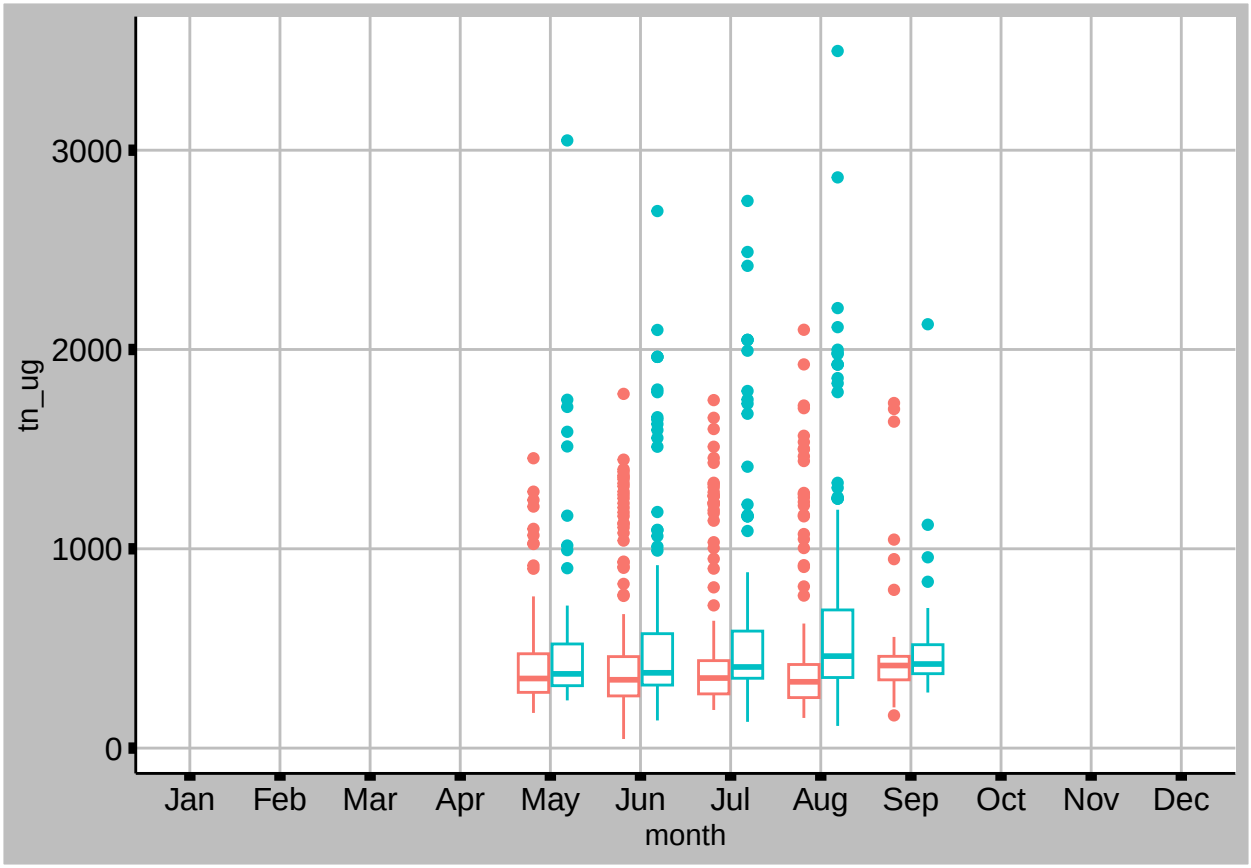
```
  scale_x_discrete(drop=FALSE) +
  mytheme +
  theme(legend.position ="none") +
  theme(panel.grid.major = element_line(color = "grey", linetype = "solid"))
PeterPaul_Chem_TP
```

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



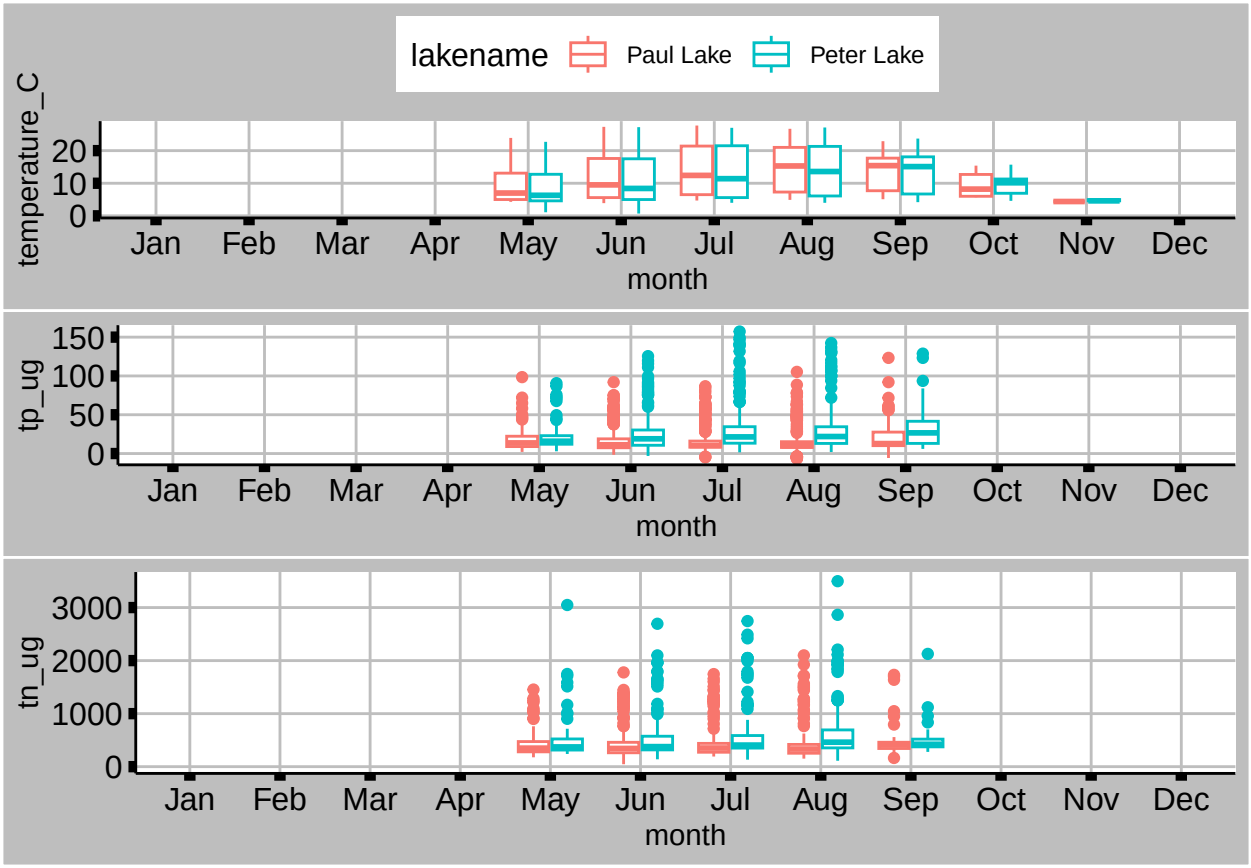

```
PeterPaul_Chem_TN <-
  ggplot(Chemistry_Nutrients_PeterPaul,
         (aes(x = month, y = tn_ug, color = lakename))) +
  geom_boxplot() +
  scale_x_discrete(drop=FALSE) +
  mytheme +
  theme(legend.position ="none") +
  theme(panel.grid.major = element_line(color = "grey", linetype = "solid"))
PeterPaul_Chem_TN
```

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




```
# Plotting all three boxplots
plot_grid(PeterPaul_Chem_Temp, PeterPaul_Chem_TP, PeterPaul_Chem_TN,
          nrow = 3, align = 'h', rel_heights = c(1.25, 1))
```

```
## 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()‘).

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



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

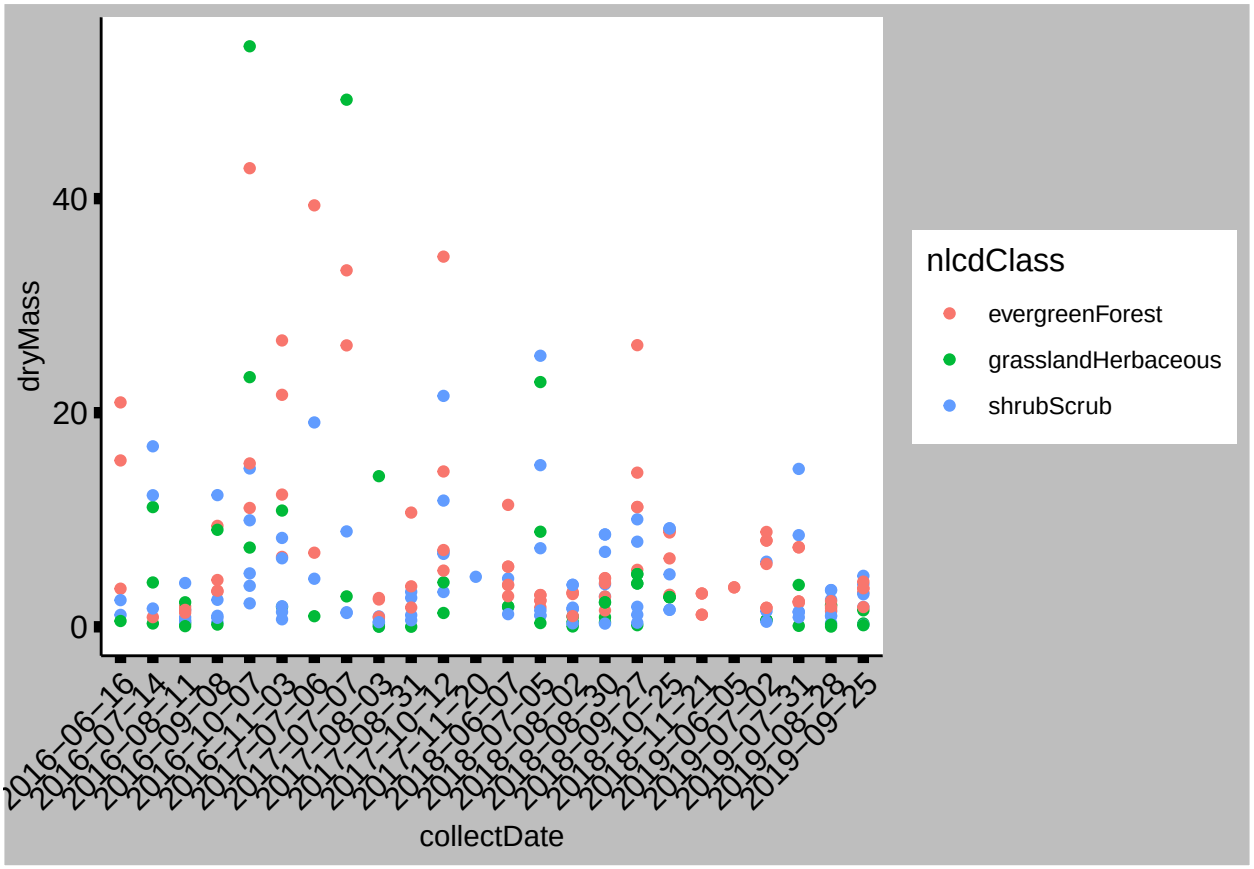
> Answer: Many of the variables that data that start in May. With the exception of temperature, tn and tp have data only going into September. There seems to be some variablity between Peter and Paul Lakes for tp and tn data while they have similiar temperatures. Peter Lakes seems to have more variability.

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)

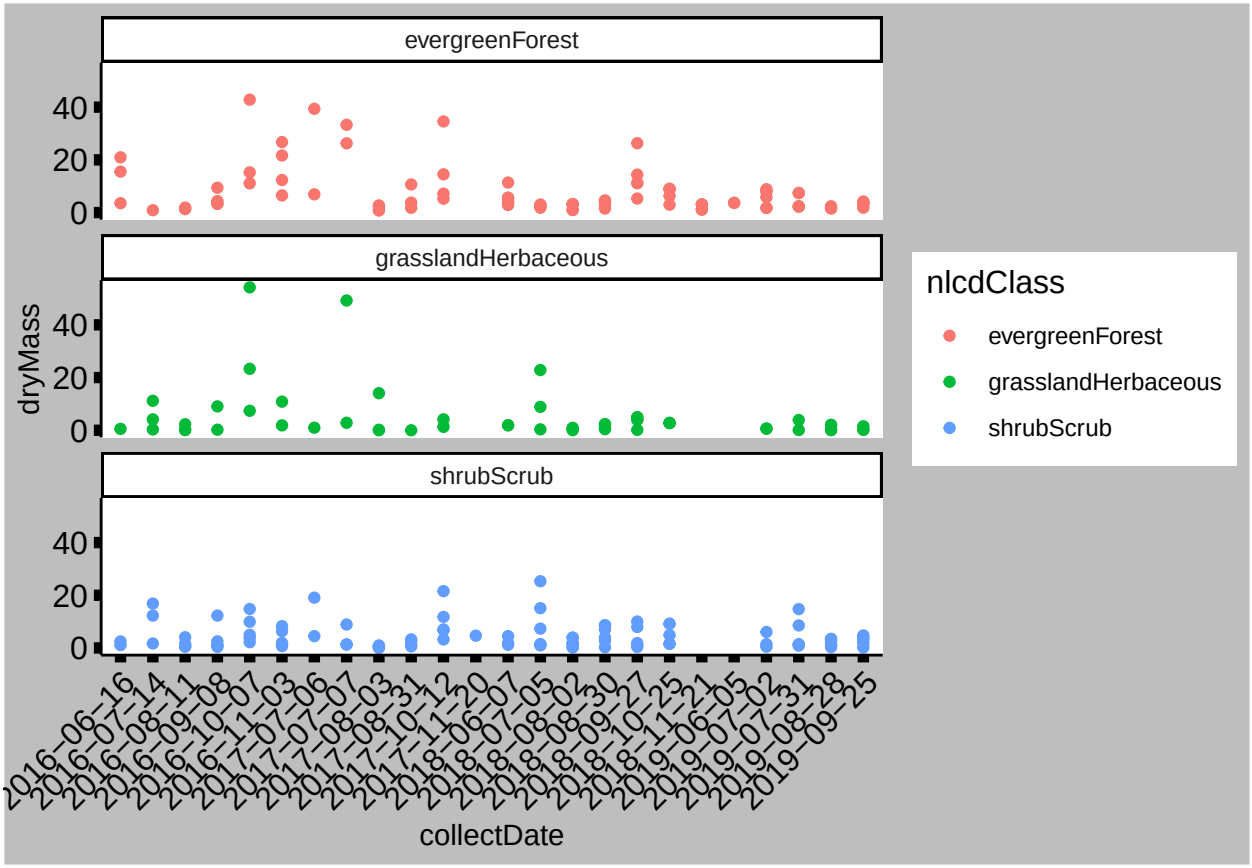
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6

# Filtering functionalGroup so Neddles is only displayed
Nitwot_Needles  <- filter(Niwot_Litter_Mass, functionalGroup == "Needles")

# Plotting dry mass by date and separating by NLCD class
Nitwot_Needles_DryMass_Date <-
  ggplot(Nitwot_Needles, (aes(x = collectDate, y = dryMass,
                              color = nlcdClass))) +
  geom_point() +
  mytheme +
  theme(axis.text.x = element_text(
```

```
    angle = 45, hjust = 1))
Nitwot_Needles_DryMass_Date
```




```
#7

# Plotting dry mass by date and separating by the various NLCD class
Nitwot_Needles_DryMass_Date_2 <-
  ggplot(Nitwot_Needles, (aes(x = collectDate, y = dryMass,
                              color = nlcdClass))) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow = 3) +
  mytheme +
  theme(axis.text.x = element_text(
    angle = 45, hjust = 1))
Nitwot_Needles_DryMass_Date_2
```



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

Answer: Plot 7 is more effective than plot 6. In plot 7, the range of the dry mass for each NLCD class is more noticeable and it is more legible. Plot 6 is more crowded and it is difficult to see the distribution of data.