Assignment 5: Data Visualization

Halina Malinowski

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A05_DataVisualization.Rmd") prior to submission.

The completed exercise is due on Monday, February 14 at 7:00 pm.

Set up your session

1. Set up your session. Verify your working directory and load the tidyverse and cowplot packages. 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 Setting up session, working directory, load packages, get data
getwd()
```

```
## [1] "C:/Users/Dell Laptop/Documents/GitHub/EDA/Assignments"
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("cowplot")
Lake_Chemistry <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Processed/NTL-LTER_Lake_Chemis
Niwot_Ridge_Litter <- read.csv("/Users/Dell Laptop/Documents/GitHub/EDA/Data/Processed/NEON_NIWO_Litter

#2 Checking dates are dates
colnames(Lake_Chemistry)
```

```
##  [1] "lakename"        "year4"           "daynum"          "month"
##  [5] "sampledate"      "depth"           "temperature_C"   "dissolvedOxygen"
##  [9] "irradianceWater" "irradianceDeck"  "tn_ug"           "tp_ug"
## [13] "nh34"            "no23"            "po4"
```

```
(class(Lake_Chemistry$sampledate))
```

```
## [1] "character"
```

```
Lake_Chemistry$sampledate <- as.Date(
  Lake_Chemistry$sampledate, format = "%Y-%m-%d")
(class(Lake_Chemistry$sampledate))
```

```
## [1] "Date"
```

```
(class(Niwot_Ridge_Litter$collectDate))
```

```
## [1] "character"
```

```
Niwot_Ridge_Litter$collectDate <- as.Date(
  Niwot_Ridge_Litter$collectDate, format = "%Y-%m-%d")
(class(Niwot_Ridge_Litter$collectDate))
```

```
## [1] "Date"
```

Define your theme

3. Build a theme and set it as your default theme.

```
#3 Building and setting theme
mytheme <- theme_grey(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")

theme_set(mytheme)
```

Create graphs

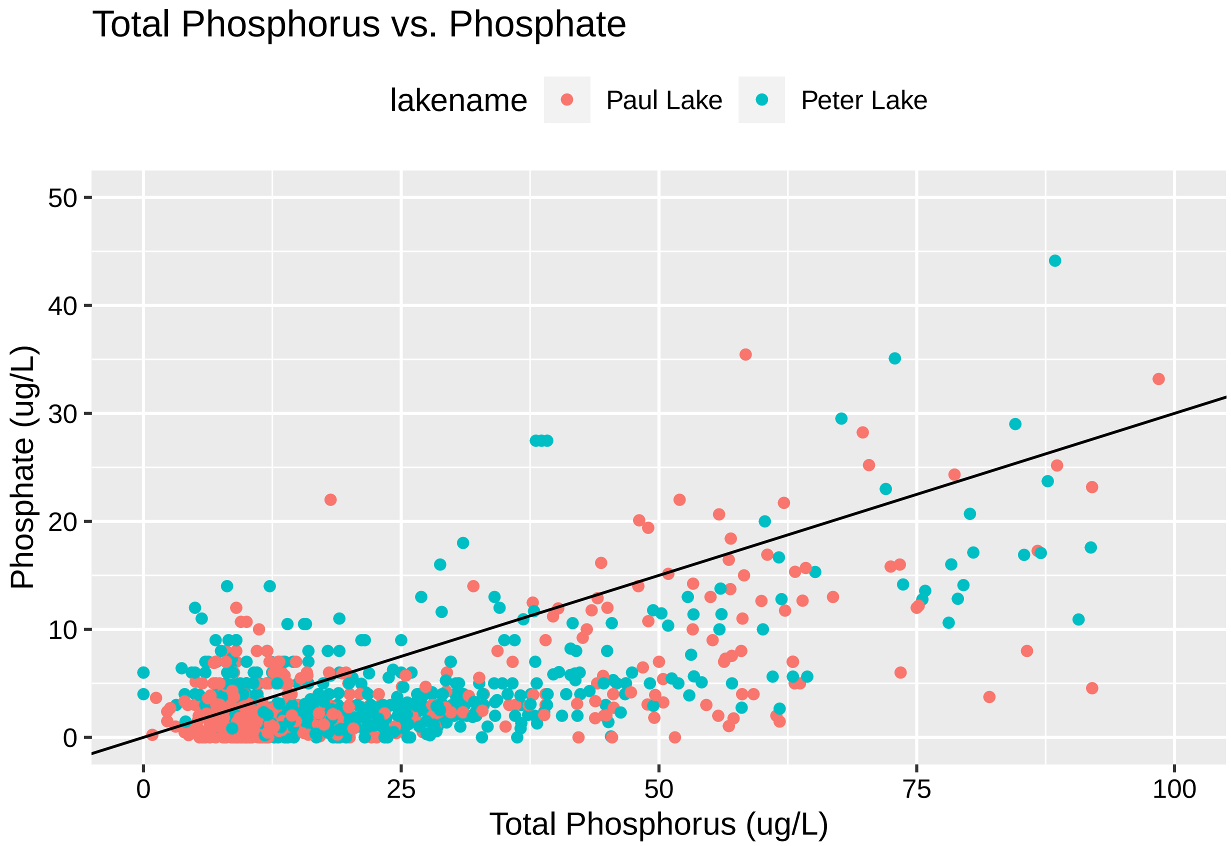
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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 `ylim()`).

```
#4 Scatter plot
phosphorus_phosphate <- ggplot(Lake_Chemistry, aes(x = tp_ug, y = po4,
                                                   color = lakename))+
  geom_point()+
  xlim(0,100) +
  ylim(0,50)+
  xlab("Total Phosphorus (ug/L)") + ylab("Phosphate (ug/L)")+
  ggtitle("Total Phosphorus vs. Phosphate") +
  geom_abline(aes(slope = 0.3, intercept = 0))

print(phosphorus_phosphate)
```

```
## Warning: Removed 21964 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.

```
#5 Boxplots
class(Lake_Chemistry$month)
```

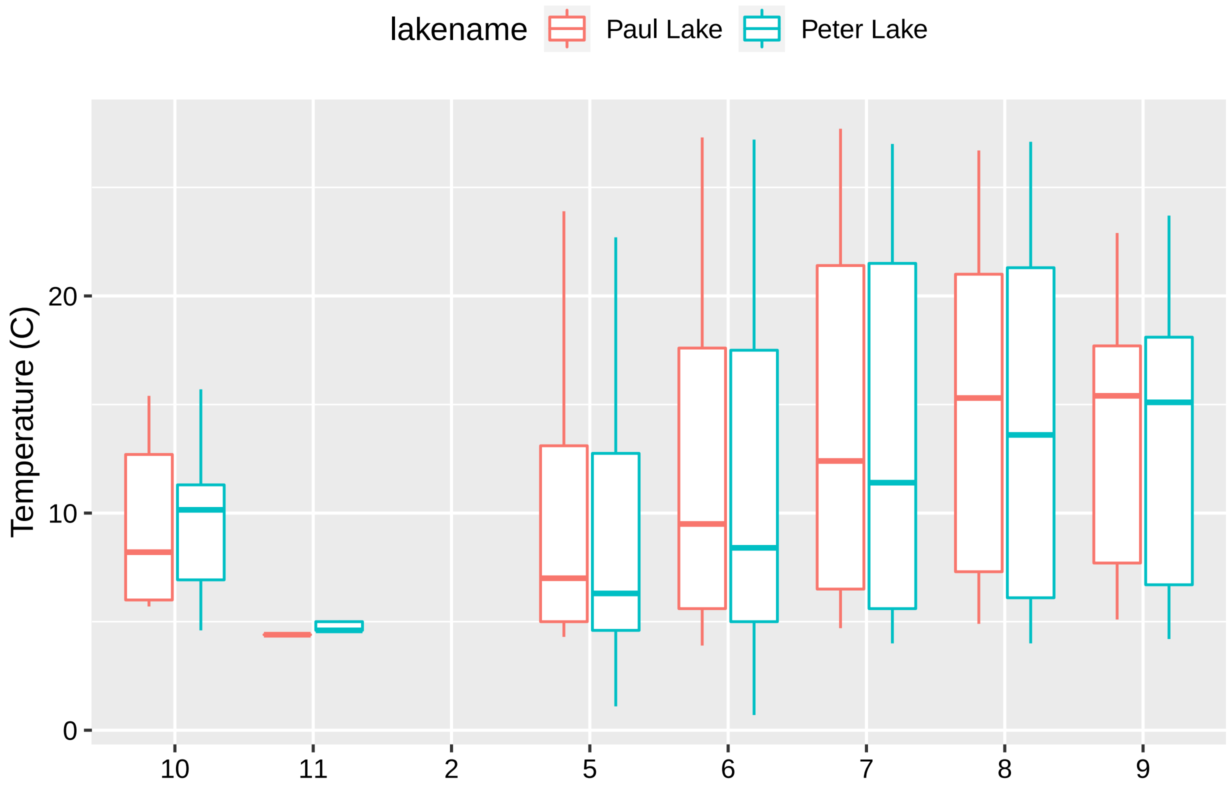
```
## [1] "integer"
```

```
Lake_Chemistry$month <- as.character(Lake_Chemistry$month)
class(Lake_Chemistry$month)
```

```
## [1] "character"
```

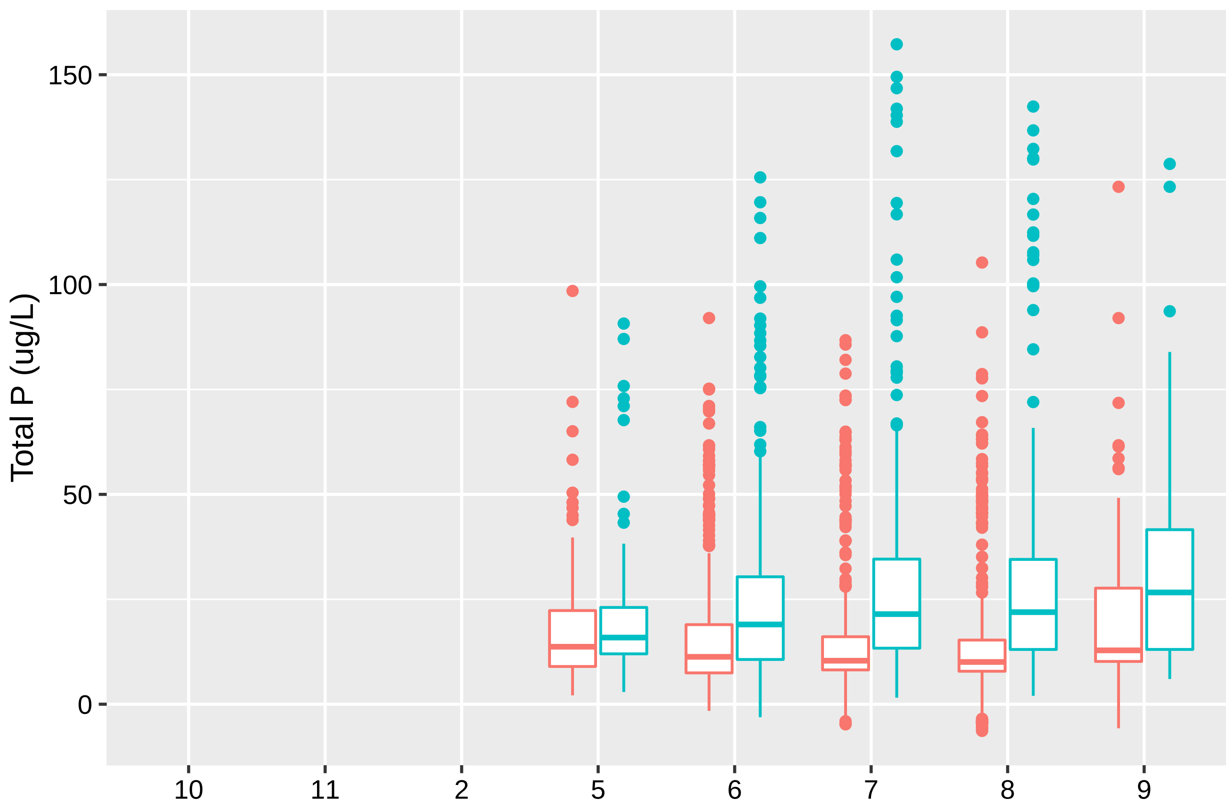
```
Temperature_boxplot <- ggplot(Lake_Chemistry, aes(x = month , y = temperature_C))+
  geom_boxplot(aes(color = lakename)) + xlab("")+ ylab("Temperature (C)")
print(Temperature_boxplot)
```

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



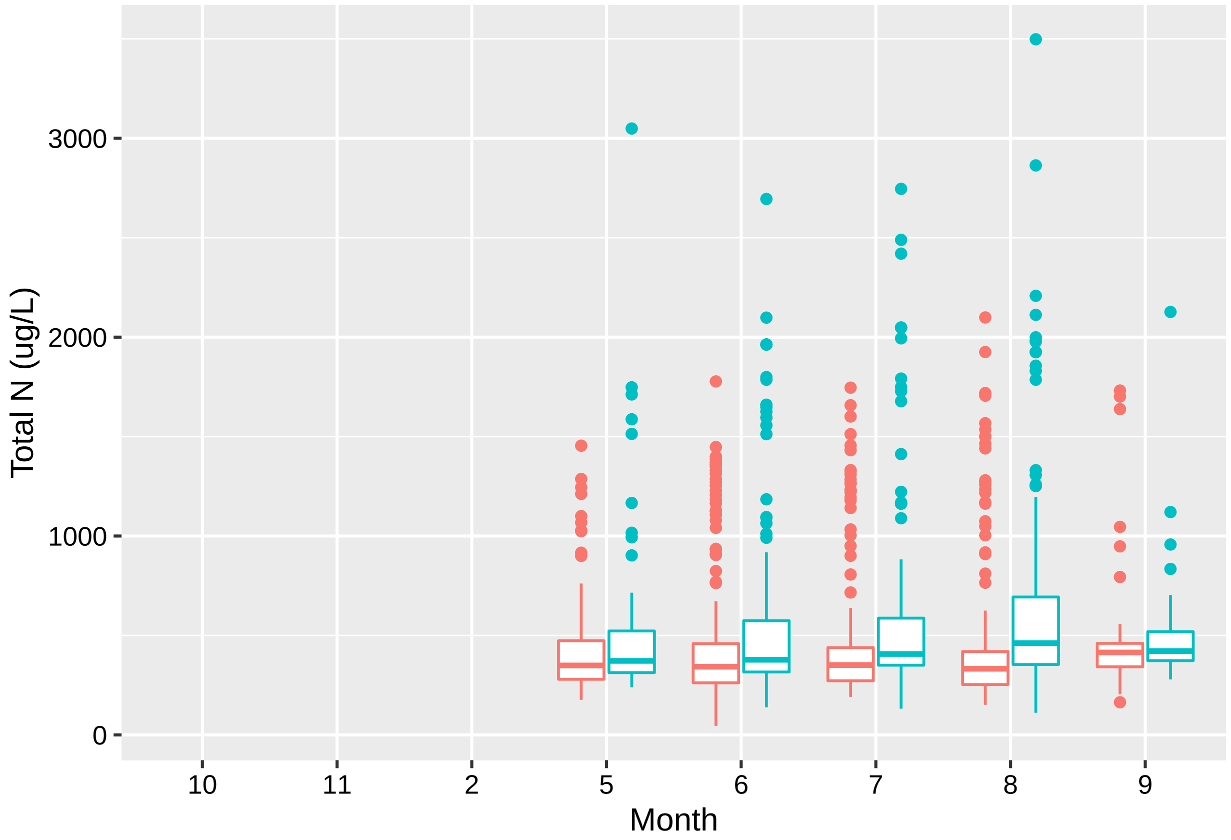
```
TP_boxplot <- ggplot(Lake_Chemistry, aes(x = month , y = tp_ug))+
  geom_boxplot(aes(color = lakename), show.legend = FALSE) + xlab("") +
  ylab("Total P (ug/L)")
print(TP_boxplot)
```

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



```
TN_boxplot <- ggplot(Lake_Chemistry, aes(x = month , y = tn_ug))+
  geom_boxplot(aes(color = lakename), show.legend = FALSE) + xlab("Month")+
  ylab("Total N (ug/L)")
print(TN_boxplot)
```

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



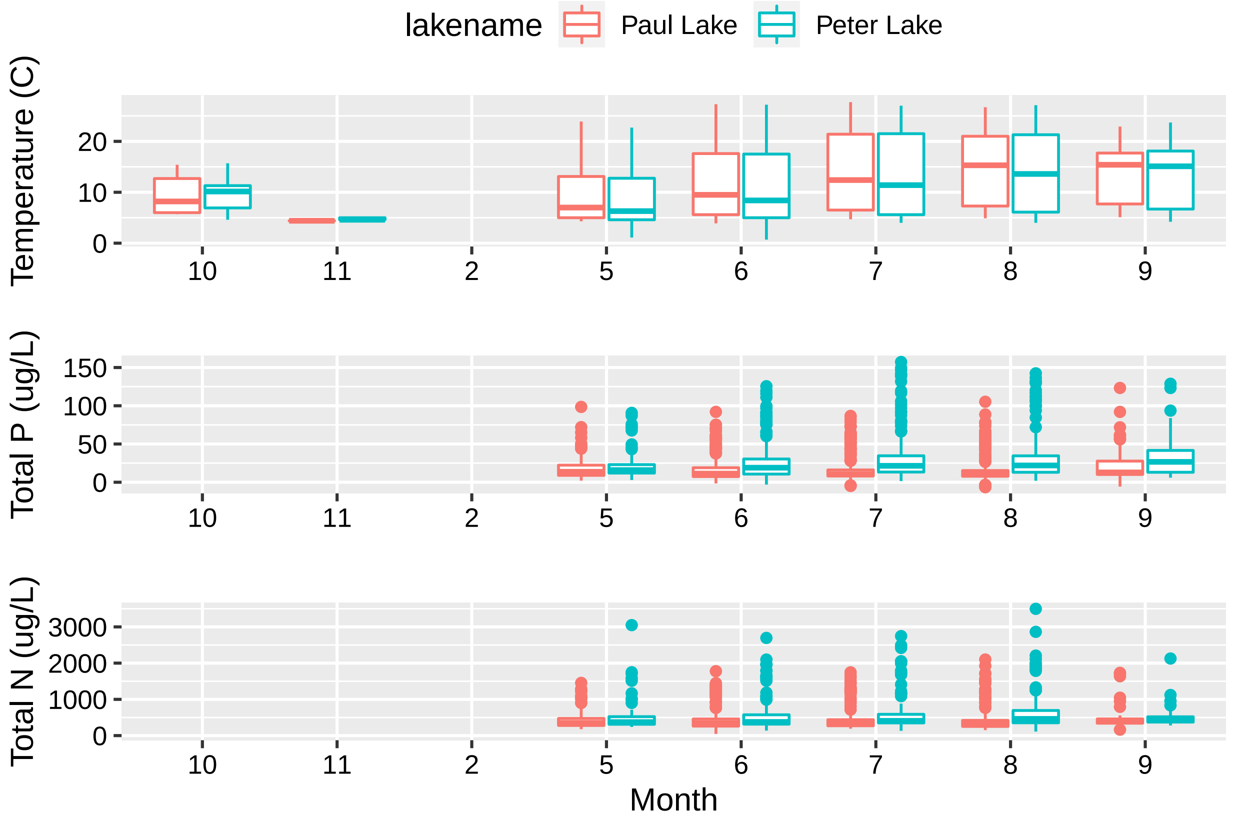
```
combined_boxplots <- plot_grid(Temperature_boxplot,
                               TP_boxplot + (aes(show.legend = FALSE)),
                               TN_boxplot, nrow = 3, align = 'v', rel_heights = c(1.5,1,1))
```

```
## 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).
```

```
print(combined_boxplots)
```



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

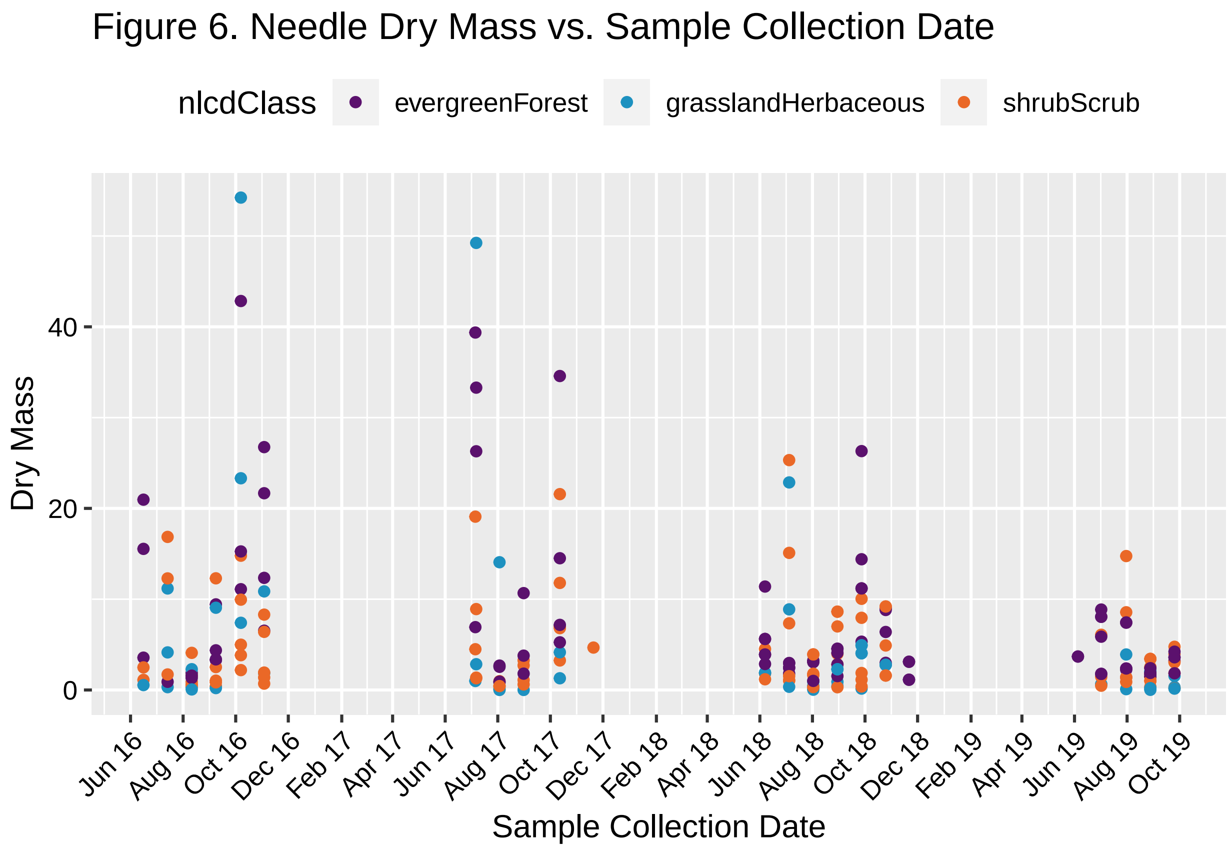
> Answer: There are several seasonal differences between the variables as well as between the two lakes. In the temperature boxplot, the trend is relatively similar for both lakes with highest mdeian temperatures recorded in August and throughout the summer months and lowest temperatures recorded in November. There is no temperature data for December - April for both lakes. Generally, Paul lake had higher temperatures than Peter Lake except for October and November.The temperature appears to steadily increase from May to August and then decrease thereafter for both lakes, which would be expected following seasonal trends in North America. For the total phorphorus boxplot there is an increase in phosphorus around the summer with the highest median values occuring in July and August. Peter Lake had higher total phosphorus for all of the months where sampels were taken. Again, there is a similar trend for total nitrogen. Highest vaules are seen in the summer for August and then followed closely by July. Peter Lake again has higher values than Paul Lake. No data was collected for phosphorus and nitrogen from October through Apri. From the observations in the boxplots there may be a positive relationship between temperature and lake nutrient concentration.

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)

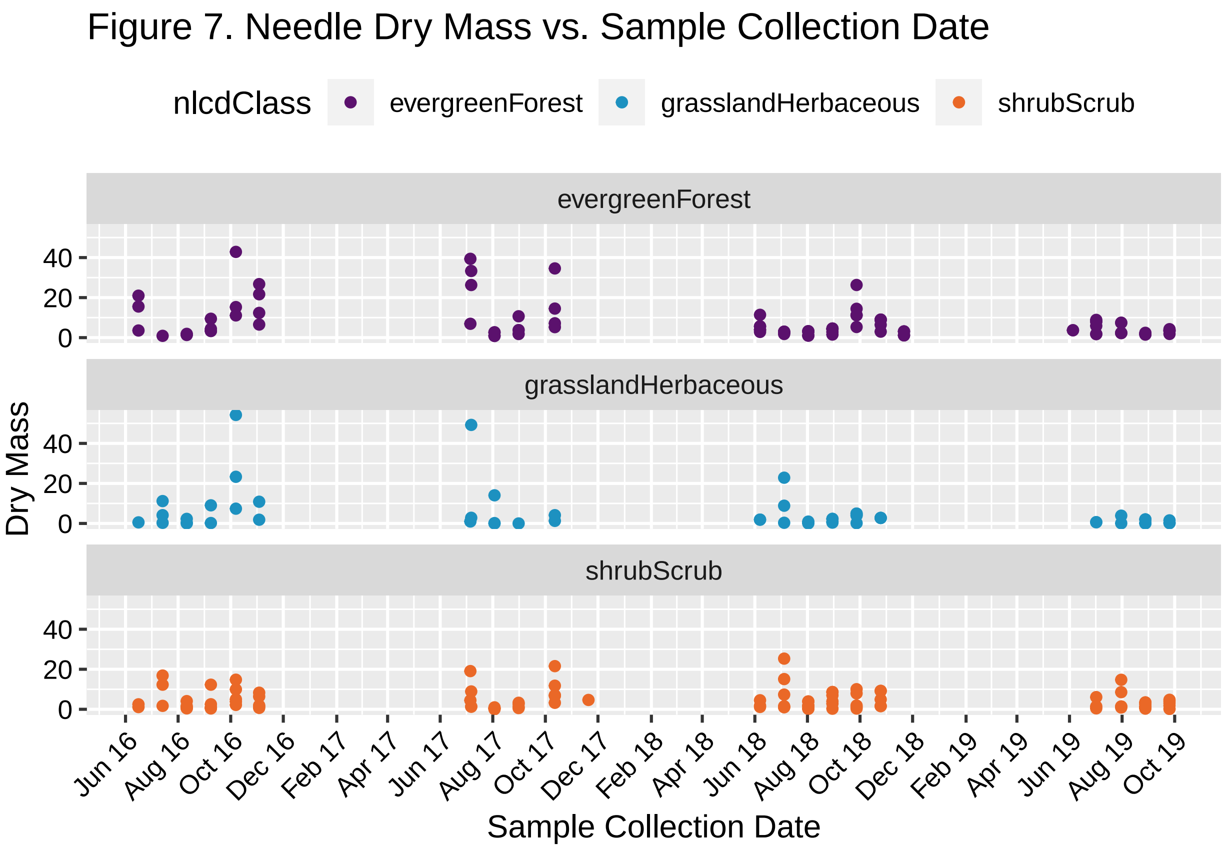
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```
#6
Niwot_Ridge_Litter_subset <- Niwot_Ridge_Litter %>%
  filter(functionalGroup %in% c("Needles"))

Needles_Mass_Date <- ggplot(Niwot_Ridge_Litter_subset, aes(x = collectDate,
  y = dryMass, color = nlcdClass))+
  geom_point()+
  scale_x_date(limits = as.Date(c("2016-06-16", "2019-09-25")),
    date_breaks = "2 months", date_labels = "%b %y")+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))+
  xlab("Sample Collection Date") +
  ylab("Dry Mass")+
  ggtitle("Figure 6. Needle Dry Mass vs. Sample Collection Date")+
  scale_color_manual(values = c("#5b116dff", "#1d91c0", "#ea6827ff"))
print(Needles_Mass_Date)
```



```
#7
Needles_Mass_Date_Facets <- ggplot(Niwot_Ridge_Litter_subset, aes(x = collectDate,
  y = dryMass, color = nlcdClass))+
  geom_point()+
  scale_x_date(limits = as.Date(c("2016-06-16", "2019-09-25")),
    date_breaks = "2 months", date_labels = "%b %y")+
  theme(axis.text.x = element_text(angle = 45,  hjust = 1))+
  xlab("Sample Collection Date") +
  ylab("Dry Mass")+
  ggtitle("Figure 7. Needle Dry Mass vs. Sample Collection Date")+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  scale_color_manual(values = c("#5b116dff", "#1d91c0", "#ea6827ff"))
print(Needles_Mass_Date_Facets)
```



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

Answer:I believe figure 7 is more effective because figure 6 appears to be over crowded. Although each graph has their pros and cons. For example, I believe it is easier to see which landcover type had the highest dry mass values per year in Figure 6. However, overall figure 7 is more esthetically pleasing and easier to read. You can still make comparisons easily between the land cover types, years, and amount of dry mass. It is also easier to investigate the values for lower dry mass in figure 7 because points are not lying on top of each other. The equal scales on the y-axis and x-axis make it so it is easy to compare the variables and values over time.