Assignment 5: Data Visualization

Clara Fast

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
#Set up session and install or load necessary packages
#getwd()
require("tidyverse")
```

```
## Loading required package: tidyverse

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

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

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

```
## Loading required package: cowplot
```

```
#Upload data files
peterpaul<-
  read.csv("./Data/Processed/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv",
           stringsAsFactors = TRUE)
niwotlitter<-read.csv("./Data/Processed/NEON_NIWO_Litter_mass_trap_Processed.csv",
                      stringsAsFactors = TRUE)

#2
#Check for date class
class(peterpaul$sampledate)
```

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

```
class(niwotlitter$collectDate)
```

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

```
#Change classes of dates to date format
niwotlitter$collectDate <- as.Date(niwotlitter$collectDate, format = "%Y-%m-%d")
peterpaul$sampledate <- as.Date(peterpaul$sampledate, format = "%Y-%m-%d")

#Check success of previous code
class(peterpaul$sampledate)
```

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

```
class(niwotlitter$collectDate)
```

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

Define your theme

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

```
#3
#Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
#Set theme as default theme
theme_set(mytheme)
```

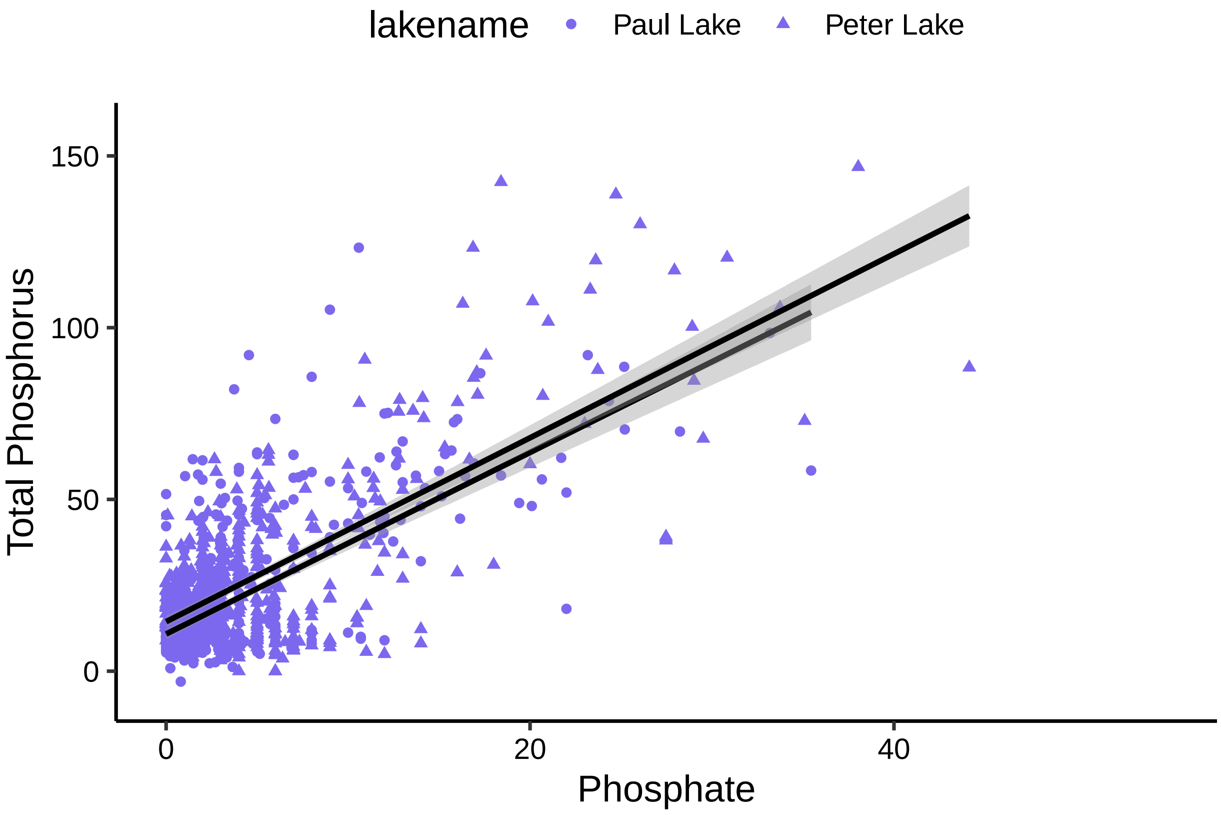
Create graphs

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
#Generate graph plotting total phosphorus by phosphate
phosphorus_phosphate1 <-
  ggplot(peterpaul, aes(x = po4, y = tp_ug, shape = lakename)) +
  geom_point(color = "mediumslateblue") + xlim(0,55) +
  geom_smooth(method = lm, color = "black") + ylab("Total Phosphorus") +
  xlab("Phosphate")

print(phosphorus_phosphate1)
```

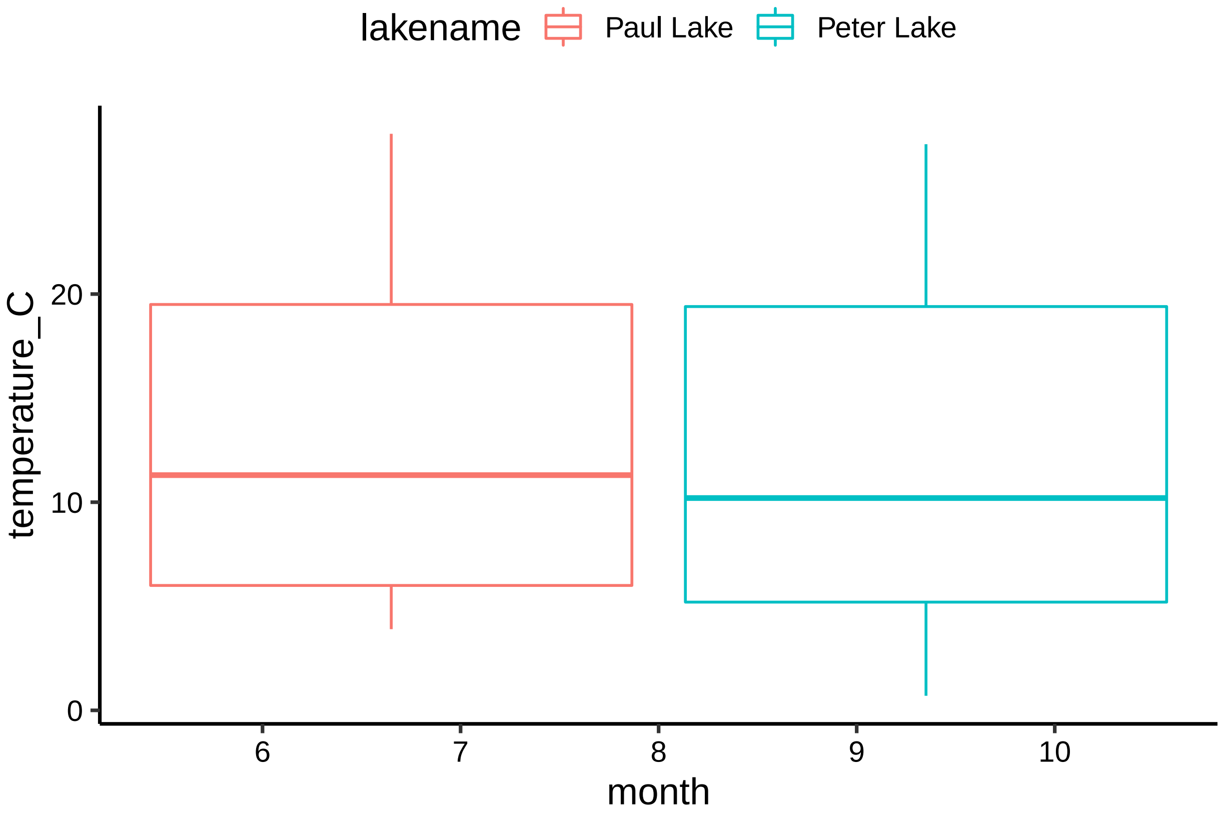
```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 21947 rows containing non-finite values (stat_smooth).

## Warning: Removed 21947 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.

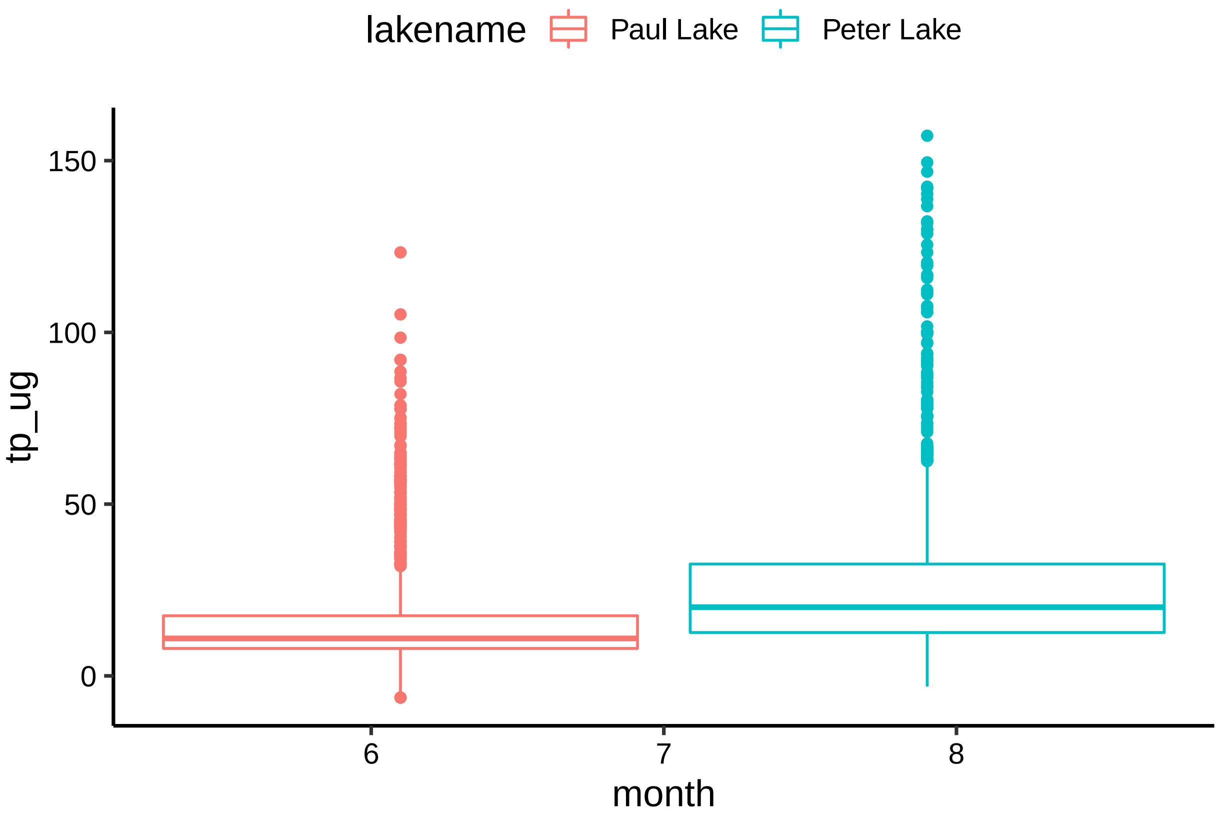
```
#5
#Generate boxplot of temperature
box_temperature<-ggplot(peterpaul, aes(x=month, y = temperature_C, color = lakename)) +
  geom_boxplot()
print(box_temperature)
```

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



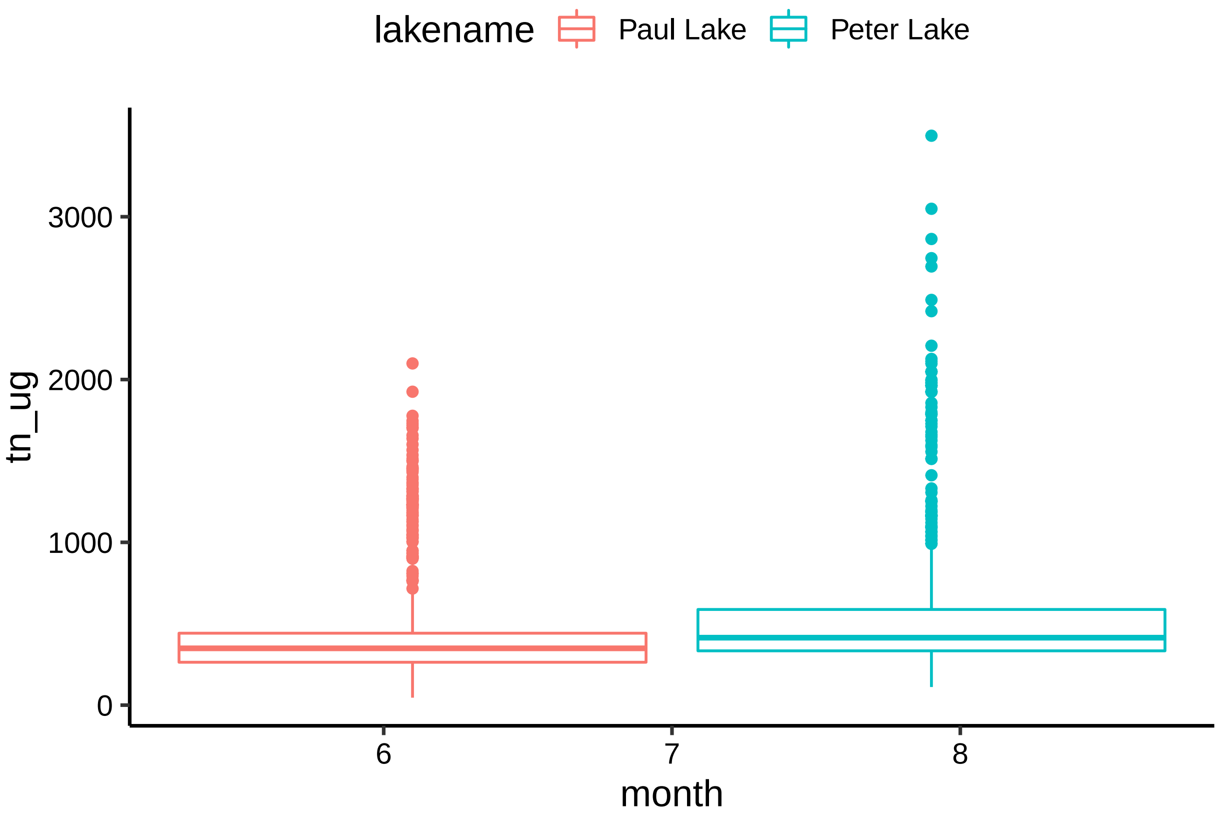
```
#Generate boxplot of TP
box_TP<-ggplot(peterpaul, aes(x=month, y = tp_ug, color = lakename)) +
  geom_boxplot()
print(box_TP)
```

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



```
#Generate boxplot of TN
box_TN<-ggplot(peterpaul, aes(x=month, y = tn_ug, color = lakename)) +
  geom_boxplot()
print(box_TN)
```

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

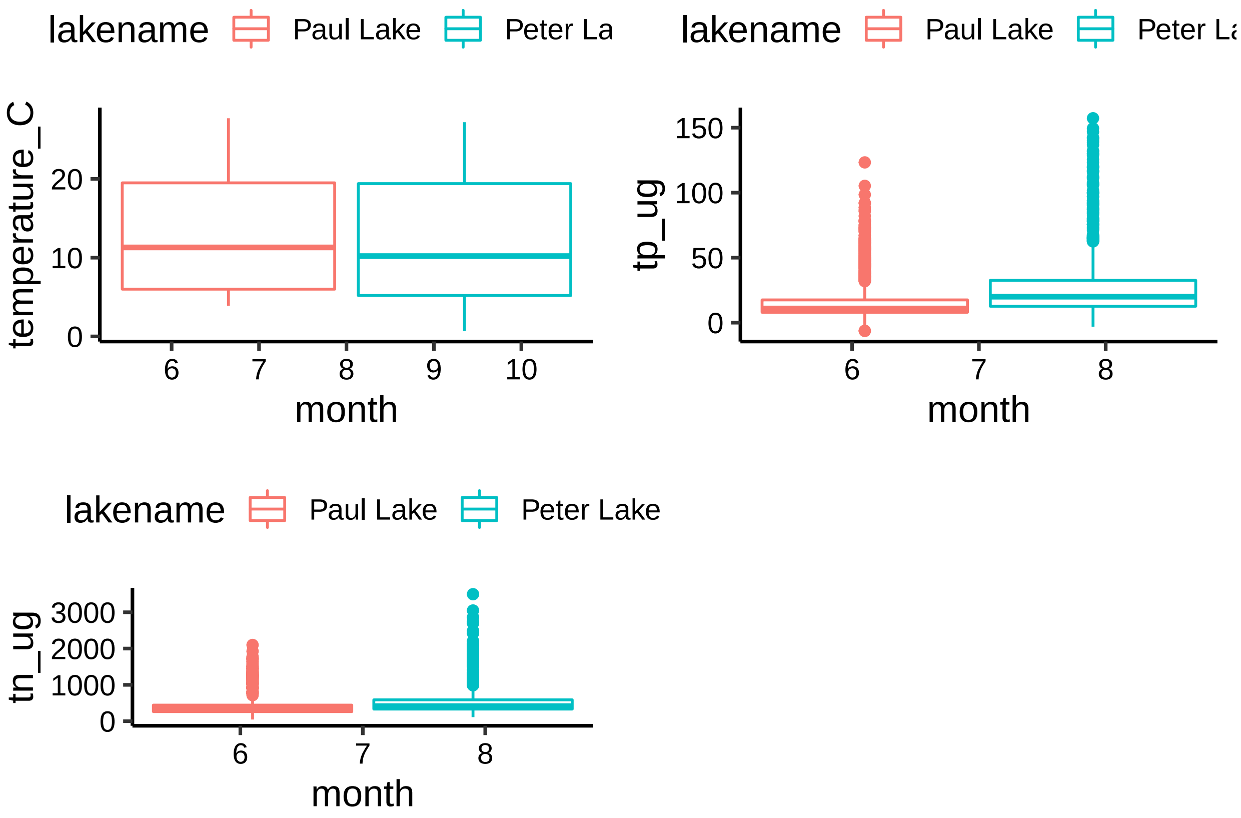


```
#Generate cowplot, combining all three boxplots
plot_grid(box_temperature, box_TP, box_TN, nrow = 2, align = 'h', rel_heights = c(1.25, 1))
```

```
## 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).
```

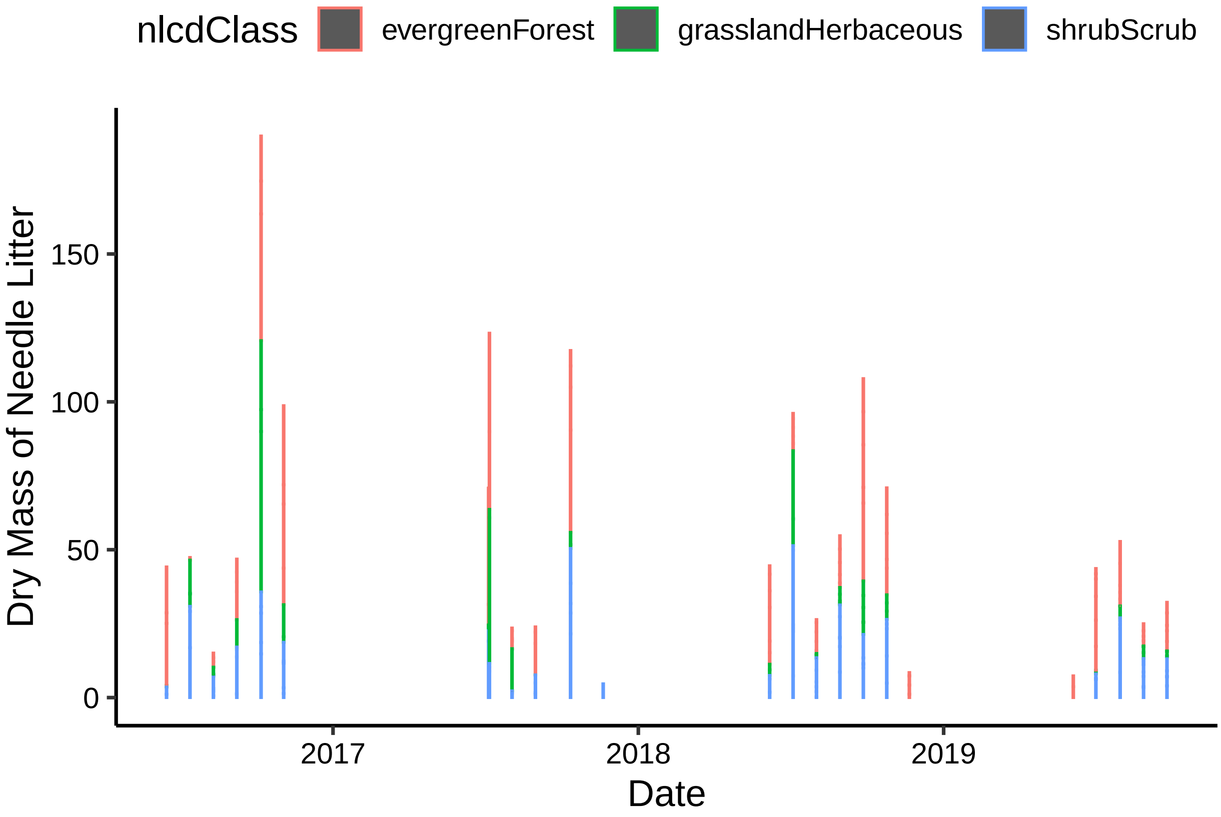


Question: What do you observe about the variables of interest over seasons and between lakes?

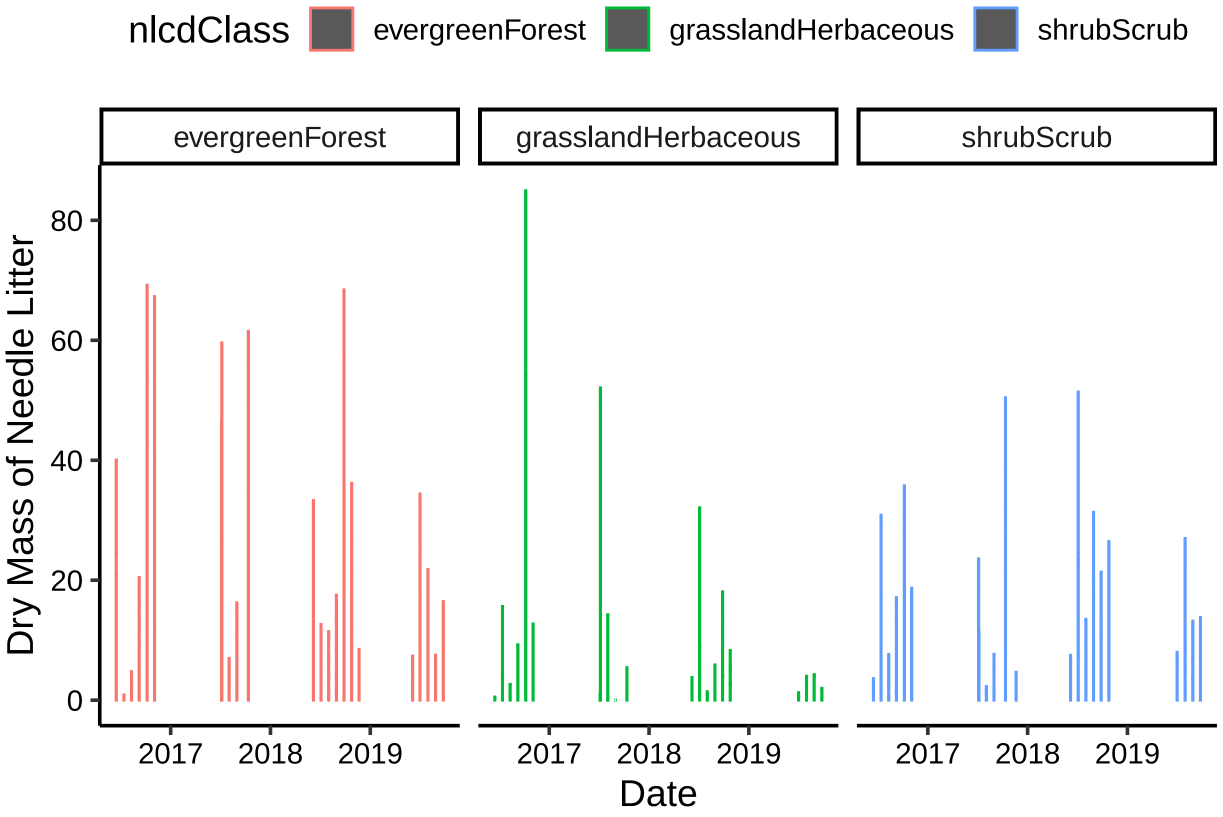
> Answer: There appears to be the largest spread for the temperature variable as indicated by the size of the boxes in the boxplot for this variable. As for the TP and TN variables, they both display data points beyond the upper quartiles of the data. The median temperature appears to be lower in the later months, while median TP and TN are higher in the later months. The same pattern is observed between lakes, the median temperature is higher for Paul Lake, but Peter Lake exhibits higher median TP and TN. However, Peter Lake only occupies the later months, and Paul Lake occupies the earlier 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.

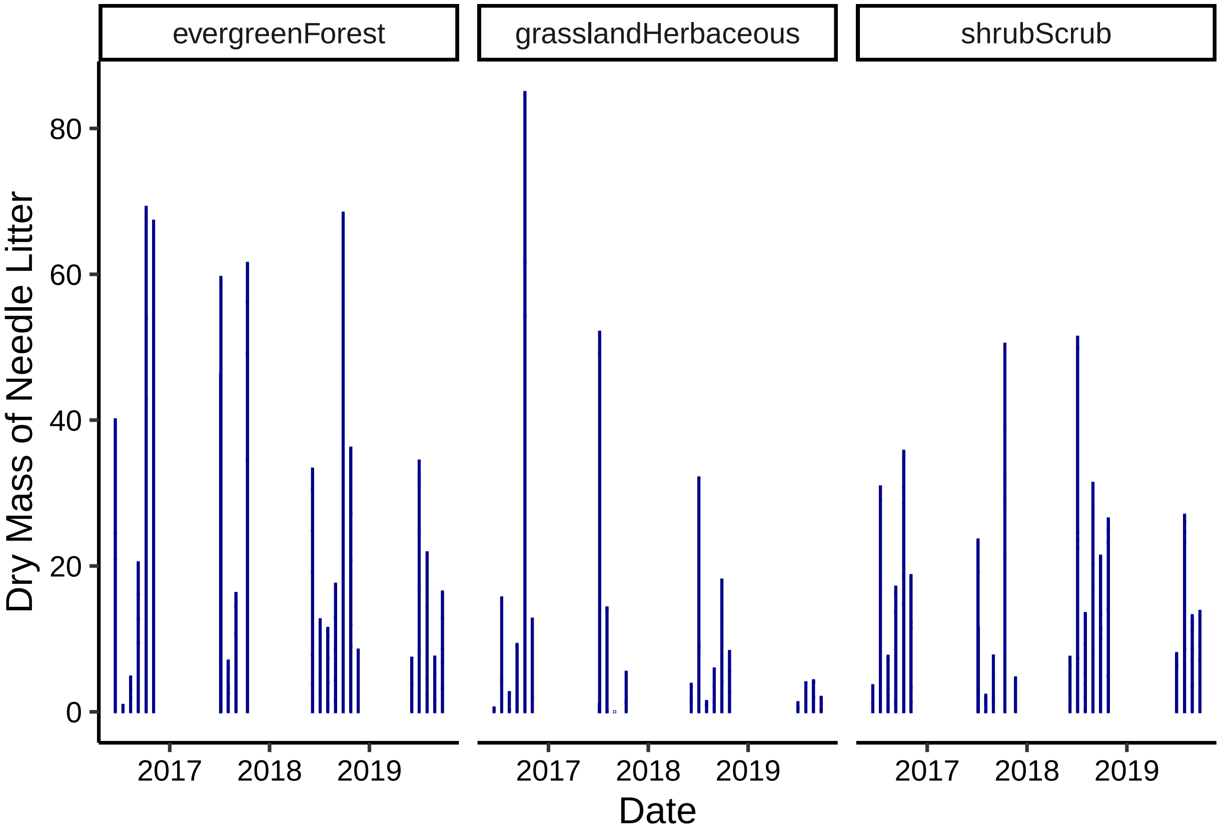
```
#6
#Filter for the Needles functional group
niwotlittersubset<-
  niwotlitter %>%
  filter(functionalGroup=="Needles")

#Generate bar plot
niwotgraph1 <-
  ggplot(niwotlittersubset, aes(x = collectDate,
                                 y = dryMass, color = nlcdClass)) +
  geom_bar(stat = "identity") + ylab("Dry Mass of Needle Litter") + xlab("Date")
print(niwotgraph1)
```



```
#7
#Generate faceted bar plot
niwotgraph2 <-
  ggplot(niwotlittersubset, aes(x = collectDate, y = dryMass, color = nlcdClass)) +
  geom_bar(stat = "identity") + ylab("Dry Mass of Needle Litter") +
  xlab("Date") + facet_wrap(vars(nlcdClass))
print(niwotgraph2)
```



```
#Faceted bar plot without the NLCD classes separated by color
niwotgraph3 <-
  ggplot(niwotlittersubset, aes(x = collectDate, y = dryMass)) +
  geom_bar(stat = "identity", color = "darkblue") + ylab("Dry Mass of Needle Litter") +
  xlab("Date") + facet_wrap(vars(nlcdClass))
print(niwotgraph3)
```



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

> Answer: I believe the first plot, the one that isn't faceted, is more effective. It is easier to compare how much each NLCD class contributes to the dry mass of needle litter. The faceted graph is overwhelming in the sense that it portrays too many bars, making it hard to grasp. Both are effective in showing the change in identity of dry mass over time, but again, the first bar graph is easier to comprehend and compare, as the bars are superimposed on each other, clearly demonstrating which class produces the most needle litter.