Assignment 5: Data Visualization

Jack Alcorn

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 Tuesday, February 23 at 11:59 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 (both the tidy [`NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv`] and the gathered [`NTL-LTER_Lake_Nutrients_PeterPaulGathered_Processed.csv`] versions) and the processed data file for the Niwot Ridge litter dataset.

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

```
#1
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## Warning: package 'purrr' was built under R version 4.0.3
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
## Warning: package 'stringr' was built under R version 4.0.3
```

```
## Warning: package 'forcats' was built under R version 4.0.3
```

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

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.0.3
```

```
getwd()
```

```
## [1] "Z:/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Assignments"
```

```
chem_nutr<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Processed/NTL-LTER_Lake_
nutr <- read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Processed/NTL-LTER_Lake_Nu
NIWOT_RD<-read.csv("/ENV_872_Data_Analy/Environmental_Data_Analytics_2021/Data/Processed/NEON_NIWO_Litt

#2
chem_nutr$sampledate<- as.Date(chem_nutr$sampledate, format = "%Y-%m-%d")
nutr$sampledate<- as.Date(nutr$sampledate, format = "%Y-%m-%d")
NIWOT_RD$collectDate<- as.Date(NIWOT_RD$collectDate, format = "%Y-%m-%d")
chem_nutr$month<-as.factor(chem_nutr$month)
```

Define your theme

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

```
mytheme<-theme_classic(base_size = 14)+
  theme(axis.text = element_text(color = "black"),
        plot.title = element_text(hjust = 0.5),
        legend.position = "right",
        legend.justification ="center",
        legend.background = element_rect(size=0.5, linetype="solid",
                                         colour ="darkblue"))
```

Create graphs

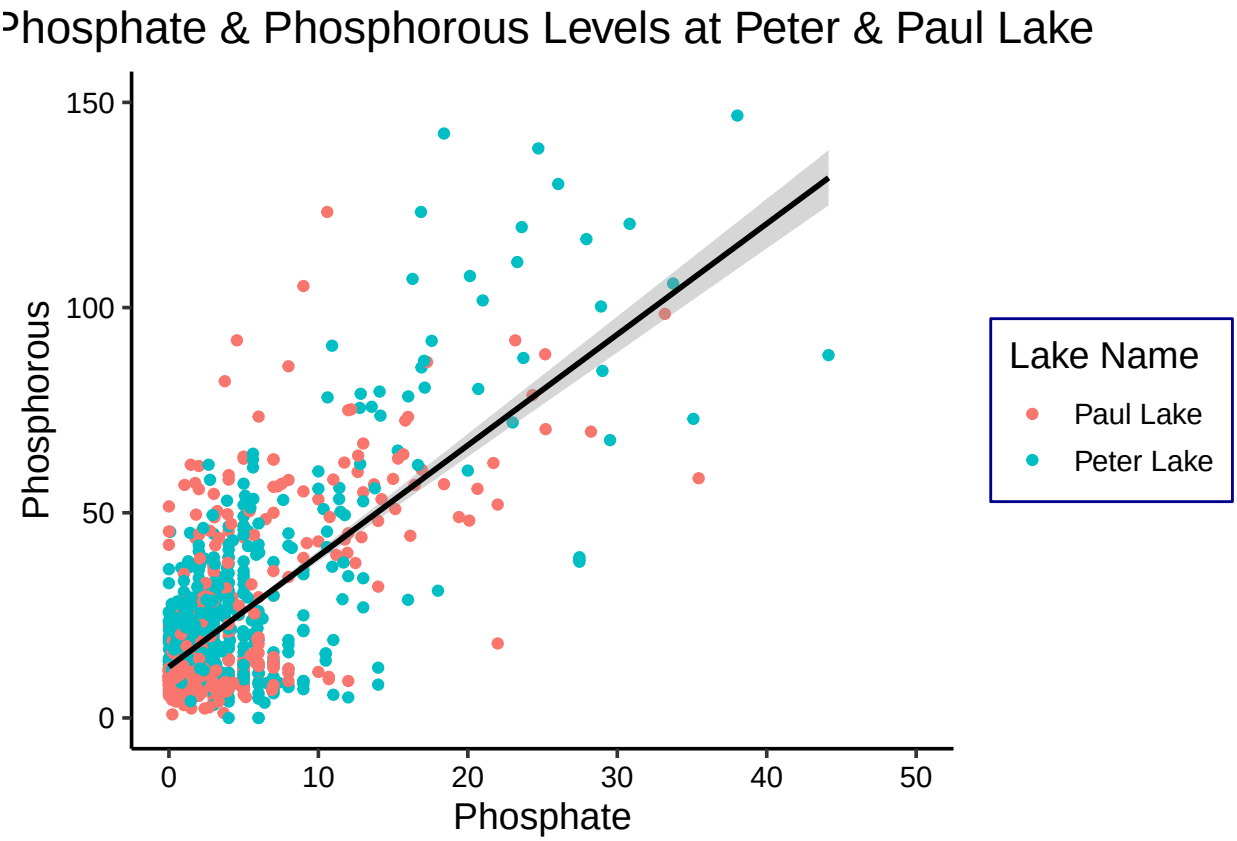
For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

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.

```
ggplot(chem_nutr, aes(x = po4, y = tp_ug, color = lakename))+
         geom_point()+
         xlim(0,50)+
         ylim(0,150)+
         xlab("Phosphate")+
         ylab("Phosphorous")+
         labs(col="Lake Name")+
         ggtitle("Phosphate & Phosphorous Levels at Peter & Paul Lake")+
         geom_smooth(method = lm, color = "black")+
         mytheme
```

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

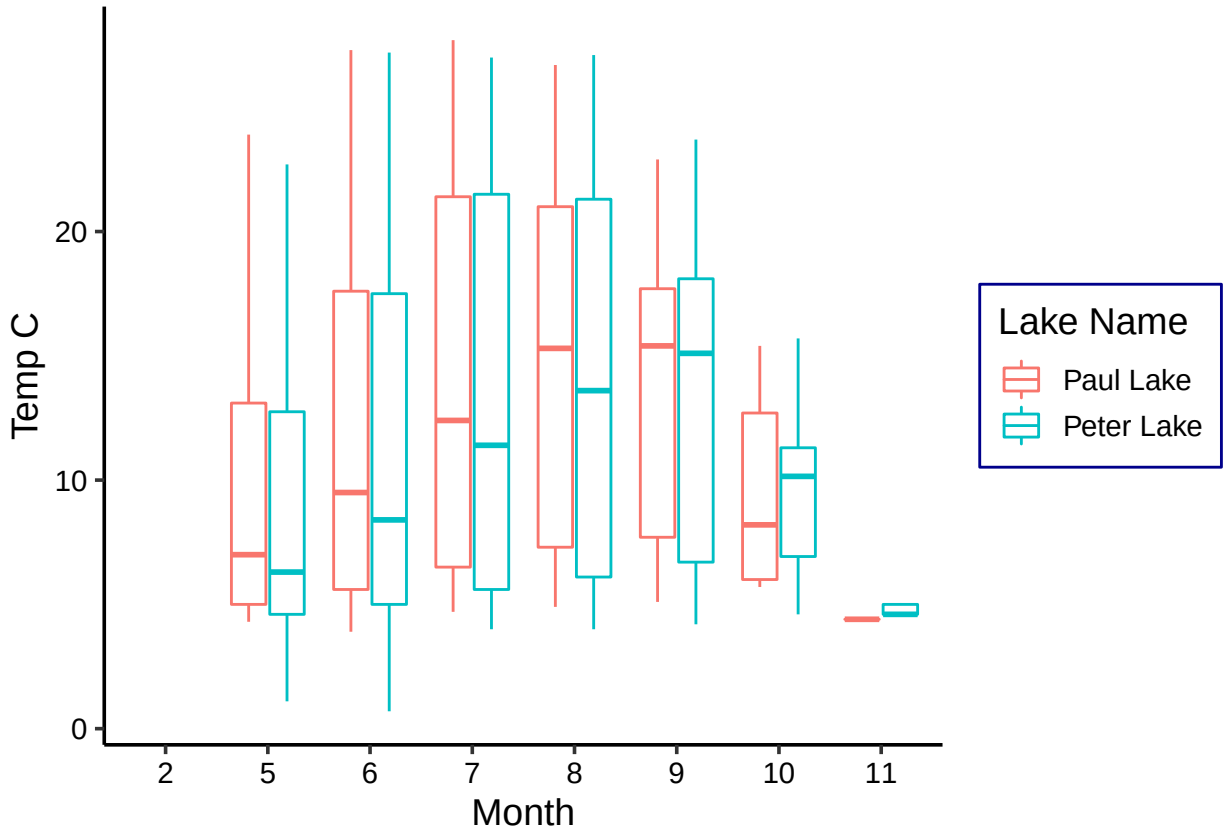
```
## Warning: Removed 21948 rows containing non-finite values (stat_smooth).
## Warning: Removed 21948 rows containing missing values (geom_point).
```



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.

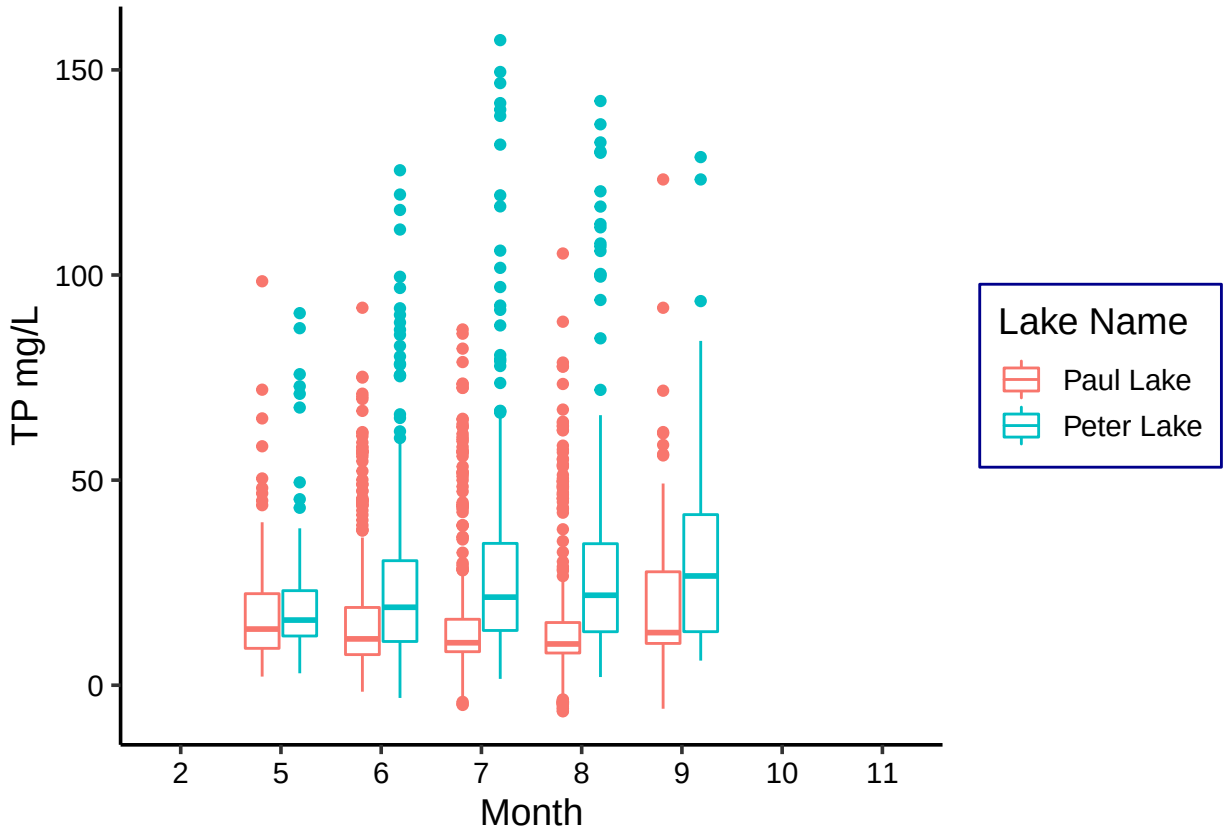
```
theme_set(mytheme)
Q5plot1<-
  ggplot(chem_nutr, aes(x = month, y = temperature_C, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab("Temp C")+
  labs(color="Lake Name")
print(Q5plot1)
```

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



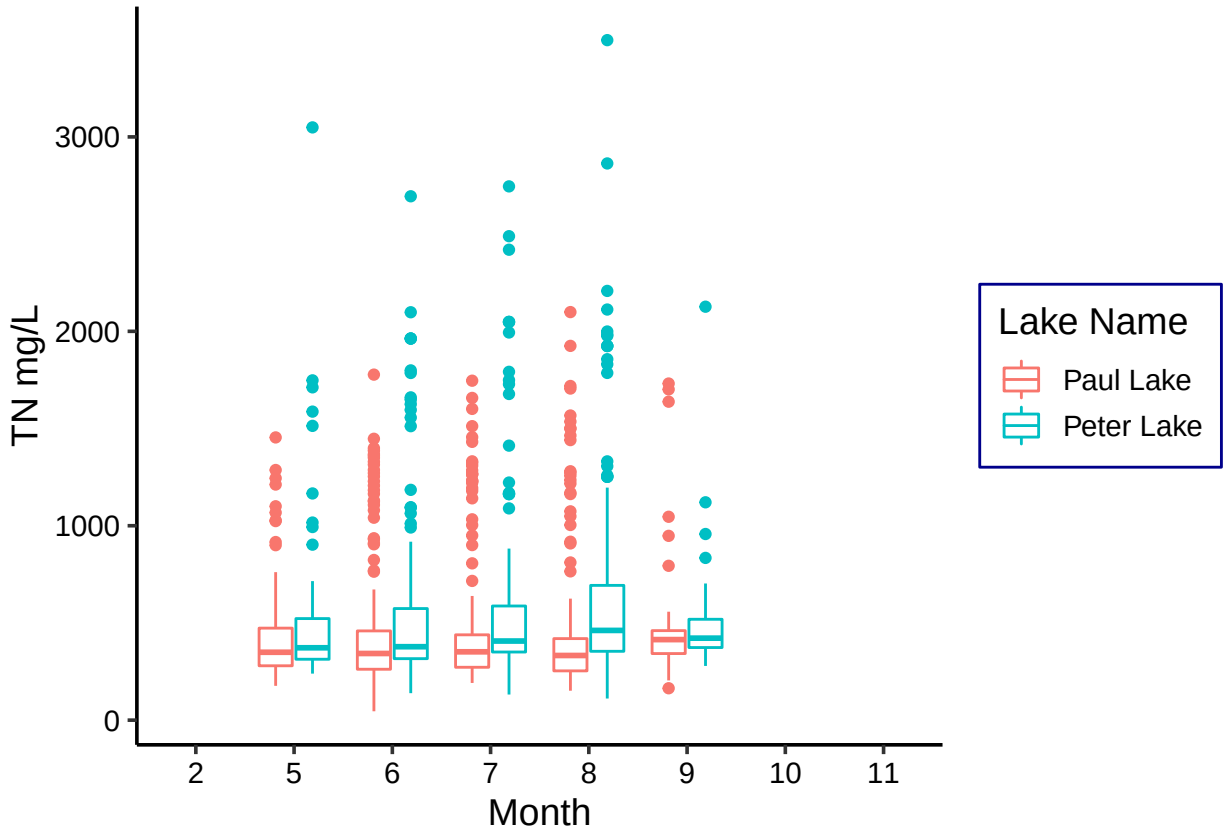

```
Q5plot2<-
  ggplot(chem_nutr, aes(x = month, y = tp_ug, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab("TP mg/L")+
  labs(color="Lake Name")
print(Q5plot2)
```

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




```
Q5plot3<-
  ggplot(chem_nutr, aes(x = month, y = tn_ug, color = lakename))+
  geom_boxplot()+
  xlab("Month")+
  ylab("TN mg/L")+
  labs(color="Lake Name")
print(Q5plot3)
```

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




```
combined_plots <- plot_grid(
  Q5plot1 + theme(legend.position = "none"),
  Q5plot2 + theme(legend.position = "none"),
  Q5plot3 + theme(legend.position = "none"),
  nrow=3, align = 'h')
```

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

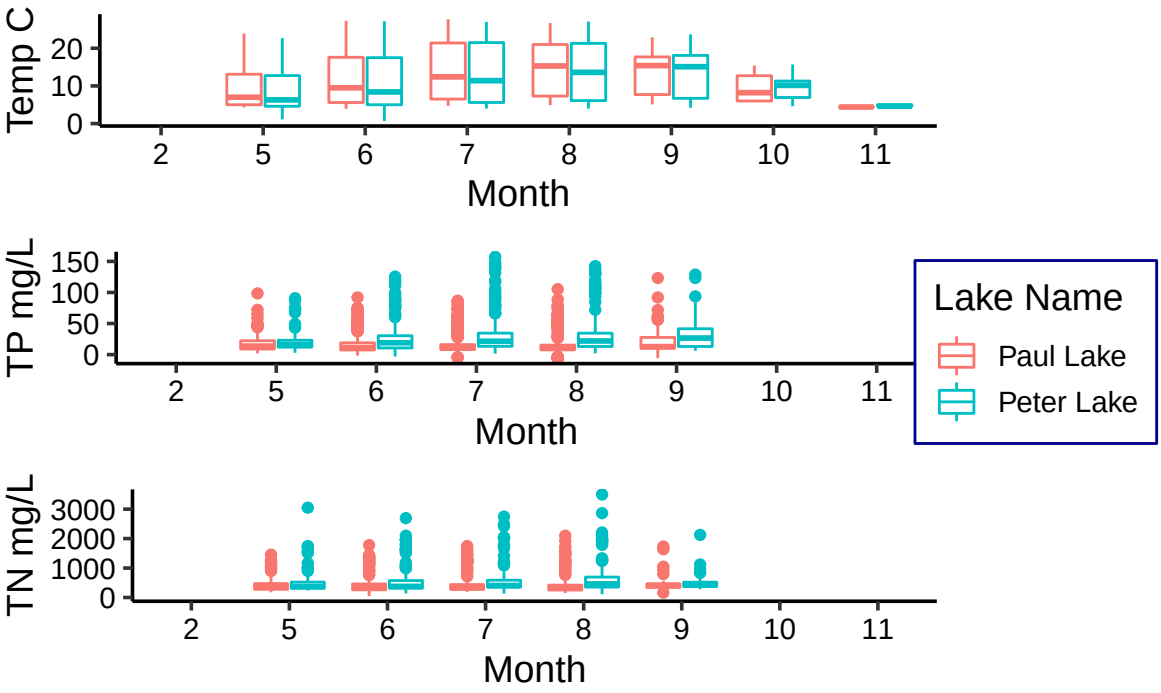
```
## Warning: Removed 20729 rows containing non-finite values (stat_boxplot).
```

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

```
legend <- get_legend(Q5plot1 + theme(legend.box.margin = margin(0, 0, 0, 12))
)
```

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

```
newtheme<-theme(axis.text = element_text(color = "black"),
        plot.title = element_text(hjust = 0.5),
        legend.position = "right",
        legend.justification ="center",
        legend.background = element_rect(size=0.5, linetype="solid",
                                         colour ="darkblue"),
        plot.margin = unit(c(1,1,1,1),"cm"))
final_combined_plot <- plot_grid(combined_plots, legend, rel_widths = c(3, .4))
final_combined_plot+newtheme
```



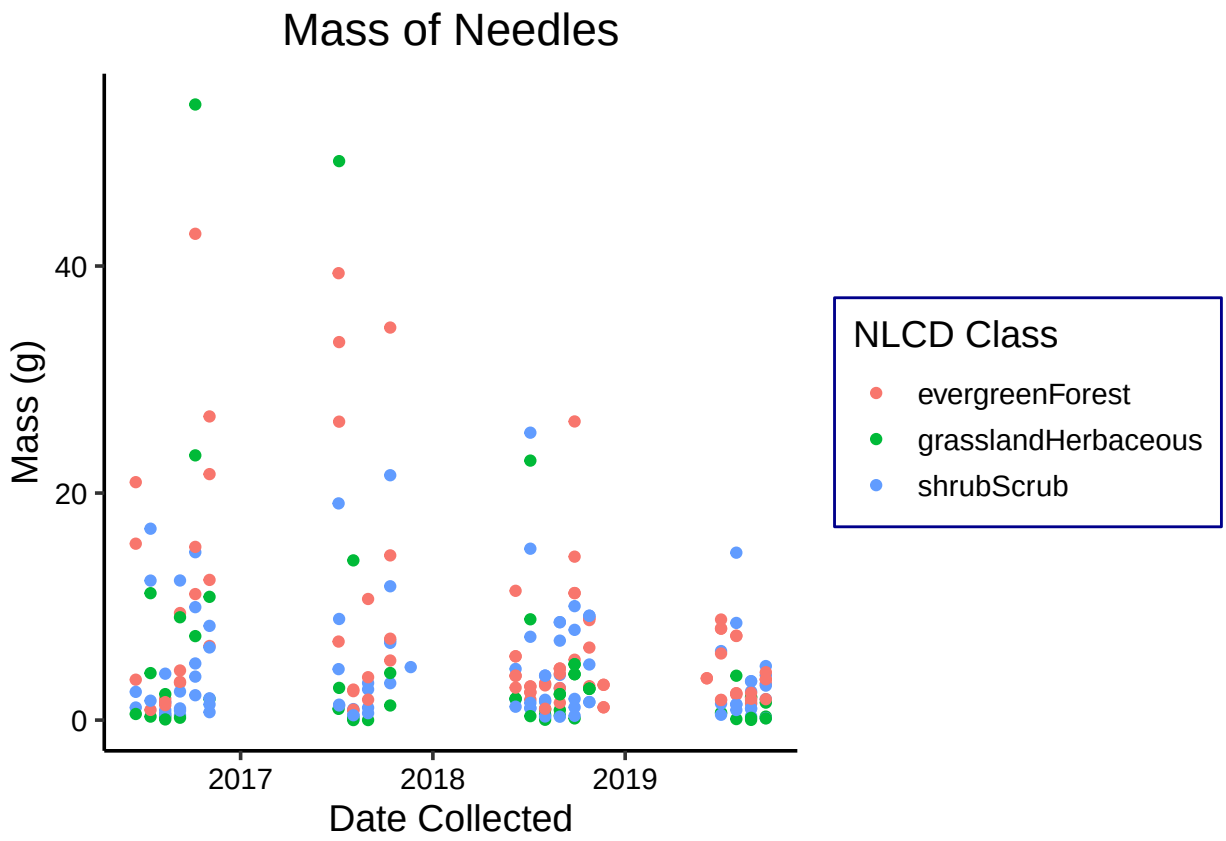
Question: What do you observe about the variables of interest over seasons and between lakes?

Answer: It appears that total nitrogen and total phosphorous values increase with temperature. These are highest during spring and summer months.

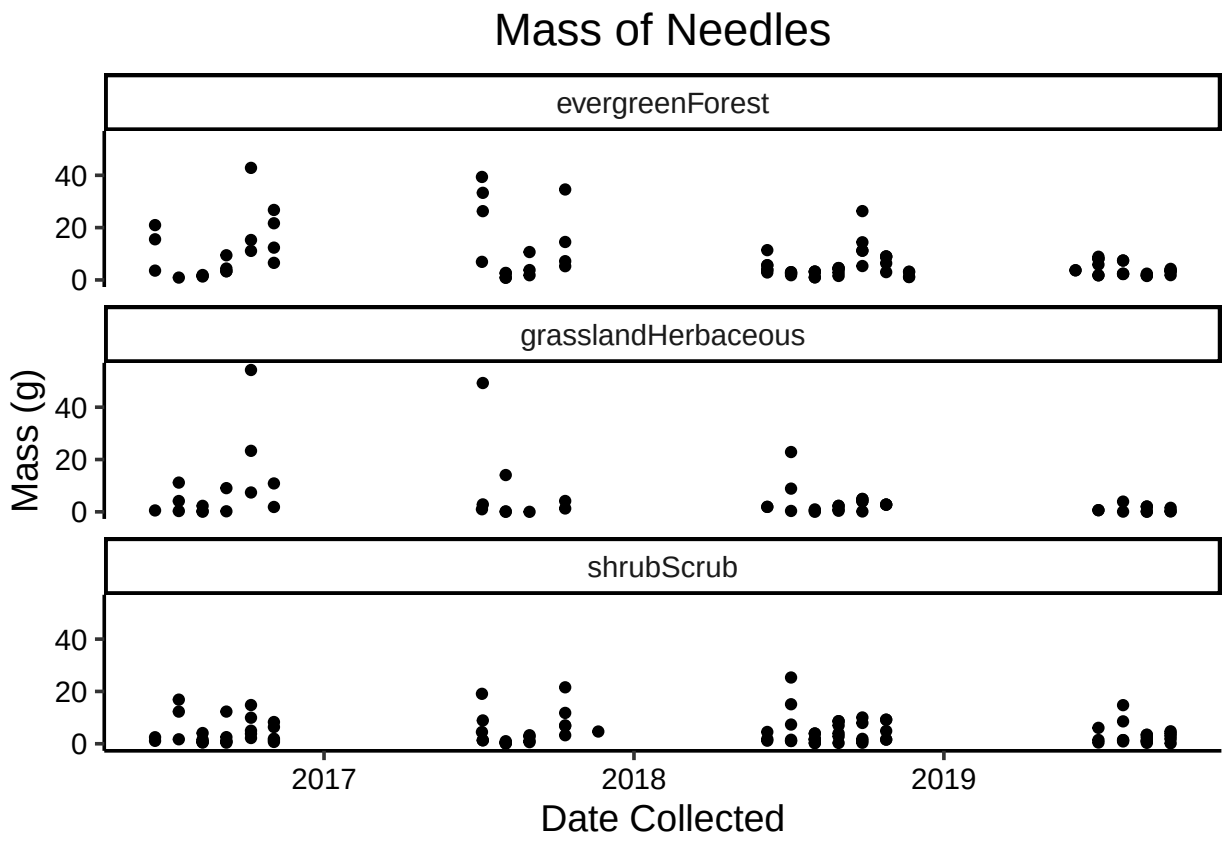
6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)

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

```
#6
theme_set(mytheme)
Needlesplot<-ggplot(subset(NIWOT_RD, functionalGroup == "Needles"), aes(x=collectDate, y=dryMass, color
  geom_point()+
  xlab("Date Collected")+
  ylab("Mass (g)")+
  labs(title = "Mass of Needles", col= "NLCD Class")
print(Needlesplot)
```




```
#7
Needlesplot_facet<-ggplot(subset(NIWOT_RD, functionalGroup == "Needles"), aes(x=collectDate, y=dryMass))
  geom_point()+
  facet_wrap(vars(nlcdClass), nrow = 3)+
  xlab("Date Collected")+
  ylab("Mass (g)")+
  labs(title = "Mass of Needles")
print(Needlesplot_facet)
```



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

Answer: I think that the graph from question 6 does a better job at displaying the results. In my opinion it is simpler and easier to quickly understand. I also personally enjoy the color coding of each NLCD class. With more time and experience I believe I could make a facet graph that is aesthtically more pleasing and easy to understand.